SOLVE	RESOLVE	License for SOLVE and RESOLVE	Download SOLVE and RESOLVE	Version: 2.11

SOLVE / RESOLVE

SOLVE:Automated crystallographic structure solution for MIR, SAD, and MAD
RESOLVE: Statistical density modification, local pattern matching, automated model-building, automated ligand-fitting, and prime-and-switch minimum bias phasing

Tom Terwilliger, Los Alamos National Laboratory
Last updated 19-Apr-2006 (version 2.11)

Subscribe to the SOLVE / RESOLVE mailing list!

Download SOLVE/RESOLVE version 2.11 manual as PDF (11 Mb)

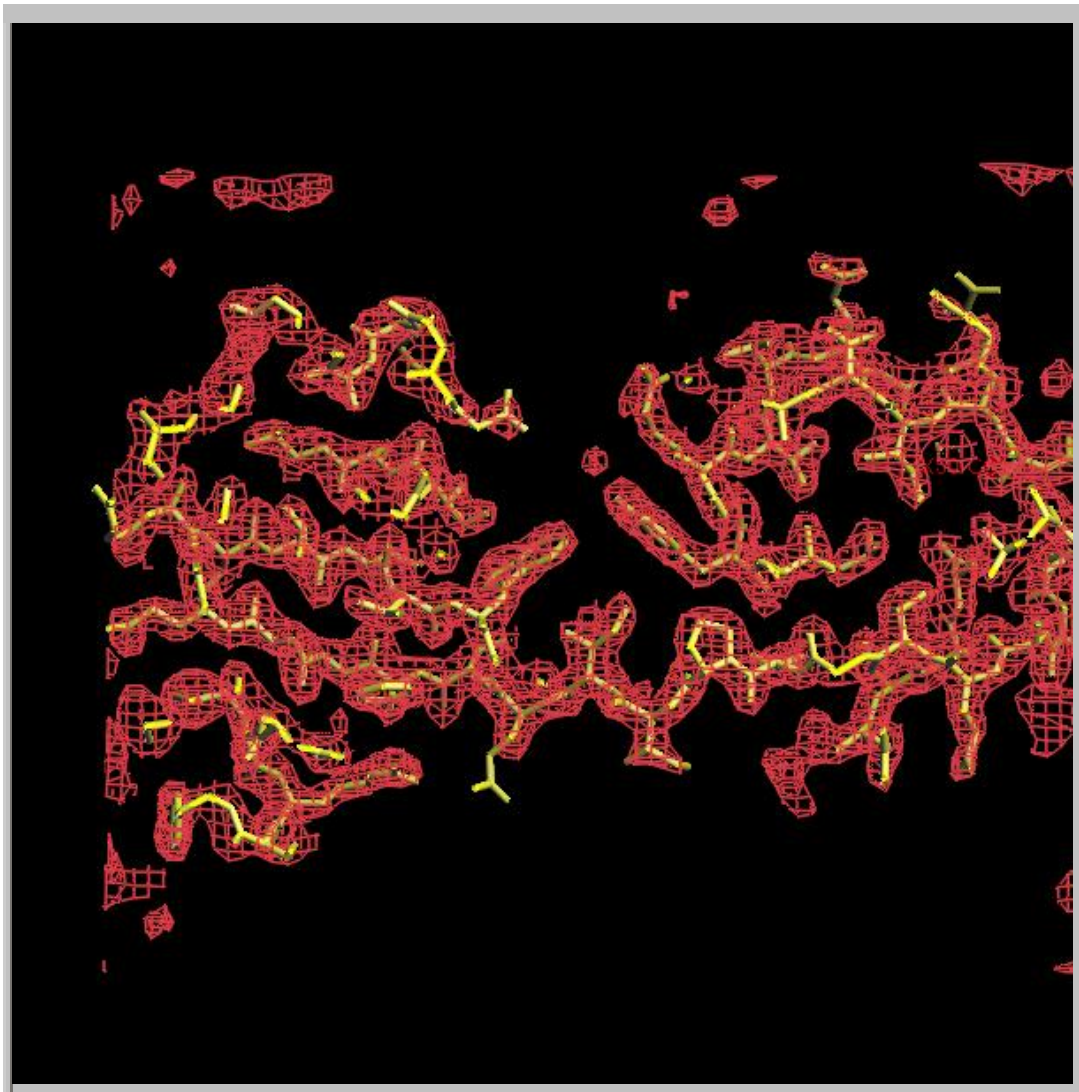
References for SOLVE and RESOLVE

SOLVE: Terwilliger, T.C. and J. Berendzen. (1999) "Automated MAD and MIR structure solution". Acta Crystallographica D55, 849-861.
RESOLVE: Terwilliger, T. C. (2000) "Maximum likelihood density modification," Acta Cryst. D56, 965-972.
RESOLVE model-building: Terwilliger, T.C. (2002) "Automated main-chain model-building by template-matching and iterative fragment extension." Acta Cryst . D59, 34-44.

(for full list see SOLVE references and RESOLVE references)

Features of version 2.11 of SOLVE / RESOLVE

- ***RESOLVE can now use multi-domain NCS in density modification.*** RESOLVE now allows you to specify up to 100 separate regions for applying NCS symmetry. Each one can have a



SOLVE/RESOLVE electron-density map and auto-built atomic model for initiation factor 5a at a resolution of 2.1A. Multiwavelength diffraction data courtesy of Tom Peat; refinement with Garib Murshudov's refmac5; graphics drawn with Alwyn Jones' O

More features of version 2.11 of SOLVE / RESOLVE...

- ***PHENIX Alpha release NOW AVAILABLE:*** The PHENIX project (www.phenix-online.org) is a collaboration aimed at developing a comprehensive and integrated platform for determing macromolecular structures. The PHENIX software has graphical tools to let you choose your structure determination pathway from pre-packaged modules and to combine them in many ways, and it has Wizards that lead you through structure determination. You can run the PHENIX Wizards from a GUI or with scripts. PHENIX has full refinement (phenix.refine), dual-space heavy-atom search (HYSS), automated model-building (RESOLVE, separate set of NCS operators. The regions can be specified with PDB files.
- ***Automated loop fitting*** RESOLVE now has a very powerful loop-fitting algorithm. You provide the residues on the ends of the loop and an mtz file with information for calculating a map, tell it how many residues are in the loop including the ends and how hard to try, and resolve will fit the loop. It takes just a few seconds to run in most cases. If you want to do this in a fully automated way, try using the ResolveBuild and IterativeBuild wizards in PHENIX!
- ***Automated fitting of flexible ligands to electron density maps with the resolve_ligand_fit script:*** RESOLVE now is capable of fitting ligands with many rotatable bonds to maps. RESOLVE starts by finding the location and orientation of the largest fixed part of the ligand, then builds all the other parts sequentially to this core. You can even give resolve a list of ligands and the resolve_ligand_id script will fit each one to the map and score them

TEXTAL), molecular replacement and maximum-likelihood phasing (PHASER), and many other powerful features. Just about anything that SOLVE/RESOLVE scripts can do, PHENIX can do (better)! Give it a try, and be sure to update as new versions become available because it is changing rapidly!

- ***Fragment identification:*** RESOLVE can identify the presence of fragments of structure (helices, strands) in your map and use them to improve your phases. Like local pattern matching, this can make a big difference if your map is of moderate quality. This is also automatically carried out using the new RESOLVE_BUILD script.
- ***Superquick model-building:*** RESOLVE now can build your model at a rate of up to 1 residue every 2-3 seconds if you have a good map ("superquick_build"). Even the more thorough standard model-building in RESOLVE is now 3 times faster than earlier versions.
- ***Swap-space needs for SOLVE/RESOLVE:*** For the standard versions, 1 GB or more of swap space is recommended (700 MB minimum). On linux machines you can now run resolve_huge, and even go as high as "isizeit = 36" if you have 4 GB of swap space. RESOLVE runs best on linux machines if they have 1 GB or more of memory
- You can give SOLVE and RESOLVE all your MAD/SAD data and it will decide at what resolution the signal-to-noise is high enough to use for phasing.
- SOLVE carries out all the steps of macromolecular structure determination from scaling data to calculation of an electron density map, automatically.
- RESOLVE uses statistical density modification (previously called maximum-likelihood density modification) to improve electron density maps
- Prime-and-switch phasing in RESOLVE removes model bias from model-phased maps. See some amazing prime-and-switch examples!
- RESOLVE automatically identifies NCS in heavy-atom sites and applies it for you.
- Version 2.11 contains all of earlier SOLVE and RESOLVE versions. Now you download SOLVE and RESOLVE both at once.
- Your version 2 license is good for all versions 2.xx of SOLVE/RESOLVE. No need for new access codes.

all, identifying which may be correct.

- ***Merging of NCS copies*** is now automatically carried out during iterative model-building and refinement with the RESOLVE_BUILD script.
- ***New standard procedures:*** The best way to use SOLVE/RESOLVE now on SAD/MAD/MIR data is: (1) edit and run one of the standard SOLVE scripts, (2) edit and run the RESOLVE_BUILD script, which does pattern-matching, the new fragment identification procedure, density modification, and iterative autobuilding and is improved from the original script in version 2.06. With fast model-building, the whole process takes only a few hours for a small protein and overnight to a few days for a moderate-sized one.
- ***Evaluate your final model automatically:*** The RESOLVE_BUILD script will automatically calculate a prime-and-switch composite omit map at the very end of model-building and evaluate the model and give you a report on how well your model fits the

	map. You can also use this script to evaluate a model you built yourself. • ***Local pattern matching:*** RESOLVE_PATTERN can identify local patterns in your map and use them to improve your phases. Local pattern matching can make a big difference if your map is of moderate quality. This is automatically carried out using the new RESOLVE_BUILD script. • ***CCP4 version 4 libraries:*** CCP4 version 4 libraries are used in versions of SOLVE/RESOLVE starting with version 2.11. Version 2.11 will work with output from previous versions. You can compile SOLVE/RESOLVE yourself with version 5 libraries if you wish (the differences are minimal for SOLVE/RESOLVE).

How do I get SOLVE and RESOLVE?	***Do you have a SOLVE license?***	***You can download SOLVE and RESOLVE upgrades at any time, once you have your version 2 license, there are no more forms to fill out. (Version 1 users need a new license.)***
	Do you need a SOLVE license?	***You need to fill out the license form; you can get a free 45-day license to try it out.***

Disclaimer/Privacy

SOLVE / RESOLVE home	SOLVE on-line manual	SOLVE Examples

SOLVE

Automated crystallographic structure solution for MIR, SAD, and MAD

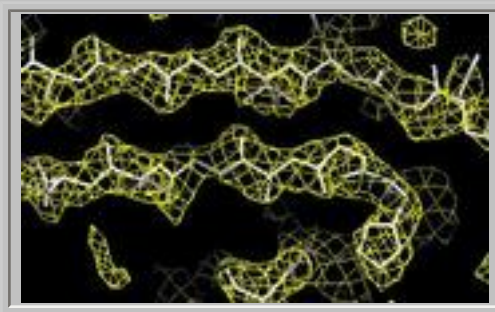
Tom Terwilliger, Los Alamos National Laboratory

Last updated 10-Apr-2006 (version 2.11)

What is SOLVE?	SOLVE is a program that can carry out all the steps of macromolecular structure determination from scaling data to calculation of an electron density map, automatically.
What can SOLVE do?	SOLVE does everything crystallographers do to solve an MIR or MAD structure, but automatically. It scales data, solves Patterson functions, calculates difference Fouriers, looks at a native Fourier to see if there are distinct solvent and protein regions, and can score partial MAD and MIR solutions to build up a complete solution. SOLVE has solved MIR and MAD structures with up to 66 heavy-atom sites. Check out some examples and look at the on-line manual for more info!

Many new features for versions 1.18/1.19/2.01/2.02/2.03/2.04/2.05/2.06/2.07/2.08/2.09/2.10/2.11 of SOLVE:

- You can tell SOLVE to ignore big peaks in the Patterson with the flag uvw_remove.
- SOLVE now will analyze your MAD/SAD data to identify to what resolution there is significant signal in your data; then it will carry out heavy-atom searches and phasing to this resolution. It will write out all the data for RESOLVE to read in and use in phase extension. You can set the working resolution with "res_phase" too.

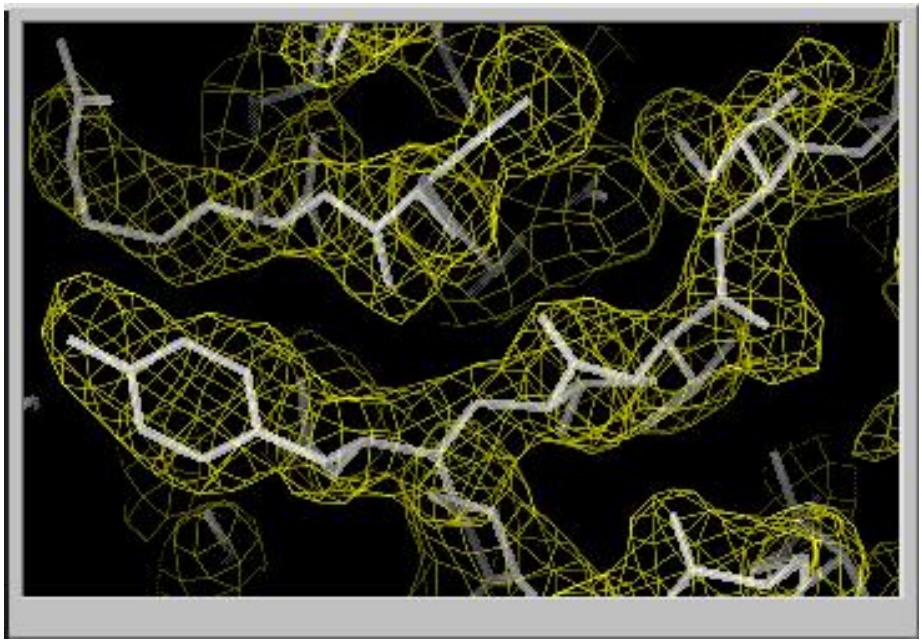


SOLVE / RESOLVE electron density map calculated after automatic solution at 2.4 A of a MAD structure with 56 selenium atoms followed by automatic maximum-likelihood density modification. Data and model courtesy of Ward Smith and Cheryl Janson

- SOLVE now has easy scripts for SAD phasing.
- SOLVE now writes out heavy atom sites in fractional coordinates in solve.xyz and also the inverse in solve_inverse.xyz. You can copy these into your script and use ANALYZE_SOLVE to easily re-run either a modification or the inverse.
- SOLVE has even better MAD phasing! SOLVE now re-refines scattering factors using the final heavy atom parameters, improving the final phases. In combination with RESOLVE, the final maps are greatly improved over earlier versions.
- Must faster search for solutions: SOLVE now follows only the very best solution by default, greatly speeding up the search in most cases.
- You can start from MR or other input phases.
- You can read CCP4 unmerged intensities directly.
- You can specify a SOLVETMPDIR where SOLVE will write scratch files.
- You can tell SOLVE where some sites are and go on from there easily with ANALYZE_SOLVE and with ADDSOLVE
- There is now a SOLVE FAQS page.
- You can load the SOLVE manual with a web browser just by typing "solvehelp" on your terminal!
- Versions 1.19 and 2.01 correct bugs in SAD phasing and allows 6 derivatives again

Disclaimer

SOLVE/RESOLVE home	RESOLVE examples	RESOLVE on-line manual



RESOLVE

Statistical density modification

Prime-and-switch minimal-bias phasing

Local pattern matching

Automated model-building

Automated ligand-fitting

Tom Terwilliger, Los Alamos National Laboratory

What is resolve?

RESOLVE is a program that improves electron density maps.

RESOLVE uses a statistical approach to combine experimental X-ray diffraction information with knowledge about the expected characteristics of an electron density map of a macromolecule.

You can run RESOLVE to improve your maps right after using SOLVE or another program to solve your structure.

Version 2.02 and higher of RESOLVE will build a model of your structure automatically.

	Version 2.03 and higher can be used for iterative model-building and refinement.
How is RESOLVE different than other density-modification programs?	RESOLVE uses a new mathematical formulation to directly maximize the total probability of the phases. Most other approaches rely on phase recombination where the optimal statistical weighting of experimental and modified phases is not known.
What is prime-and-switch phasing and how does it minimize model bias?	Prime-and-switch phasing primes density modification with biased model phases, then switches entirely to an independent source of phase information (the probability of the map) to remove the model bias. Prime-and-switch phasing can give excellent unbiased maps even for crystals with very low solvent content (provided that the biased model did have substantial correct phase information!)
What are other new features of version 2 of resolve?	In addition to prime-and-switch phasing, version 2.0 of RESOLVE can find non-crystallographic symmetry in your heavy-atom sites and apply it automatically. Version 2.02 and higher can build a model for you as well. Versions 2.05 and higher can identify local patterns and use them to improve your phases.Versions 2.06 and higher can carry out iterative pattern id, fragment id, and model-building. Resolve versions 2.08 and higher can carry out automated ligand fitting as well.
How can I read more about resolve?	The mathematics behind RESOLVE and examples of its use are described in the article "Maximum likelihood density modification," appearing in Acta Cryst. D56, 965---972 (2000).

Last updated: 10-Apr-2006 (version 2.11)

SOLVE / RESOLVE Home Page

SOLVE/RESOLVE licensing information

U.S. Universities, non-profits, other non-governmental institutions A one-time $500 license fee is good for all versions from version 2.00 to version 2.99 for all machines at your institution. You can get a free 45-day trial license too.
U.S. Government institutions For U.S. government institutions, the SOLVE/RESOLVE license is free of charge. One license is good for all machines at your institution.
U.S. Commercial institutions Contact the Los Alamos Technology Transfer Division for commercial licenses. You can get a free 45-day trial license if you wish.

Non-U.S. Non-commercial institutions A one-time $600 license and export fee is good for all versions from version 2.00 to version 2.99 for all machines at your institution.
Non-U.S. Commercial institutions Contact the Los Alamos Technology Transfer Division for non-U.S. commercial licenses.

< br > < br >

SOLVE/RESOLVE home page

Downloading and upgrading SOLVE/RESOLVE

The current version of the SOLVE/RESOLVE package is version 2.10.

Please note: One license is good for all machines at your institution for all versions from 2.00 to 2.99 (it goes in "solve2.access" and the upgrade is free for academic/non-profits)

There are three easy parts to installing SOLVE/RESOLVE on your system

- you need to figure out which version of SOLVE/RESOLVE matches your system
- you need to ftp a compressed file to your computer, uncompress it, and run install.csh to put the files in the right places on your system and set up solvehelp (a link to the local version of the manual).
- you need to put the "solve2.access" file you get from us by email in the right place on your system

Once you have SOLVE/RESOLVE installed you can view the manuals just by typing "solvehelp".

1. First you need to choose the file that matches your computer. If none does, then you can get a version compiled on your computer directly by emailing me at "terwilliger@lanl.gov". Each download file contains "solve" and "solve_giant" and "solve_huge". Same for resolve. The giant-size and even bigger huge versions are for huge unit cells and are not usually necessary. Also if you have a really enormous unit cell you may need an even bigger version which you can receive by emailing me directly.

If your computer is a...	*Then ftp this file...*
Alpha running Digital (Tru-64) Unix	solve-2.10-alpha.tar.gz
SGI (R5000 or higher, most Octanes)	solve-2.10-sgi.tar.gz
SGI (Altix)	solve-2.10-altix.tar.gz

Mac OSX (Tiger) Thanks to Luca Jovine! See www.biosci.ki.se/groups/ljo/software/solve_osx.html for installation info.	solve-2.10-osx.tar.bz2
linux (Red Hat 7.3 or Fedora 3 or higher on Pentium III or higher)	solve-2.10-linux.tar.gz
All other machines (and some SGI Octanes which won't run SGI version of SOLVE)	email terwilliger@lanl.gov for source to compile on your machine. You will need to have the CCP4 suite installed on your machine

2. Here is how to get the file containing the SOLVE and RESOLVE programs by ftp, uncompress it, and put the files in the right places on your system. (Note that you can substitute any directory you want for "/usr/local/lib/solve" in the install.csh script and then set the environmental variable SOLVEDIR to point to that directory in everyone's .login or .login_custom file.)

- Make the directory /usr/local/lib/solve on your computer and get into it as "root"
- Click on the ftp links above or this one to download the file you need:

ftp solve.lanl.gov/pub/solve/2.10/

Please note: If this download does not work please try directly running ftp from a terminal with:

```
ftp solve.lanl.gov
anonymous
your-email-address
cd pub/solve
ls
binary
get name-of-the-download-file-you-want
quit
```

- uncompress and extract the files (Note: on OSX use bzip2 to unzip the file):

```
gunzip solve*.gz
tar xof solve*.tar
```

- install solve by running install.csh (after editing it if you wish):

```
cd solve-2.10/
./install.csh
```

- Please note: on some systems the binaries supplied use shared libraries. If your system does not have the shared libraries you will get an error message when you run the programs. The solution is to compile the program on your own machine. Just email terwilliger@lanl.gov and I'll send you instructions.

- You are just about ready to go. Now you can run SOLVE or RESOLVE (usually in /usr/local/bin/solve and /usr/local/bin/resolve) for regular-size runs, and solve_giant and resolve_giant or the bigger solve_huge and resolve_huge for huge unit cells.

- Now this directory also has in it all the symmetry files
- This directory also has the local on-line manual. You can access the on-line manual just by typing "solvehelp" now. (It may only take effect after you log in again).

- Change for versions 2.08 and higher: RESOLVE uses the CCP4 version 5 libraries. In your scripts you will now want to specify both SYMOP (as in earlier versions) and SYMINFO. These are located in the same place. If you use the ones supplied with SOLVE/RESOLVE, they are located in:

  ```
  setenv SYMOP /usr/local/lib/solve/symop.lib
  setenv SYMINFO /usr/local/lib/solve/syminfo.lib
  ```

- Finally, you need to set the CCP4_OPEN environmental variable. Put in your .login_custom or else at the beginning of all files to run solve the following command. (It allows ccp4 routines to overwrite existing files. If you don't do this solve will stop the second time you run it when it tries to open solve.mtz)
  - setenv CCP4_OPEN UNKNOWN # for sh or csh shells
  - export CCP4_OPEN=UNKNOWN # for bash or ksh shells
- You may also wish to set the environmental variable SOLVEDIR which tells SOLVE and RESOLVE where to look for solve2.access and symmetry files. (Change the directory if you don't use the default one.)
  - setenv SOLVEDIR /usr/local/lib/solve/ # for sh or csh shell
  - export SOLVEDIR=/usr/local/lib/solve/ # for bash or ksh shells
- You are now ready to go as soon as your solve2.access file is ready

3. Here is how to set up your solve2.access file. After your license is completed you should receive an email from us with the two lines of information for the solve2.access file. (Note that the old solve.access

file for version 1 won't work any more.) They will look like:

```
License for SOLVE/RESOLVE version 2 expiring 01-jan-03
X89A943951
```

- There should be exactly 10 characters left justified on the second line. This is the access code.
- Please note: There must be a carriage return at the end of the second line of solve2.access for RESOLVE to read it correctly.
- Put the 2 lines into a file called "solve2.access" and put this file in the directory /usr/local/lib/solve or else in the directory named by the environmental variable SOLVEDIR. (See "Intro/Getting Started/The SOLVEDIR environmental variable" for more information on setting that variable.)
- You are ready to go. You should be able to run solve by typing "solve" from any directory if your path is set up to look in /usr/local/bin for programs. You can always type "/usr/local/bin/solve" to run solve if your path isn't set.

If you have problems getting SOLVE/RESOLVE going then:

- try again a couple times following the instructions as closely as possible
- check that the machine you are working on matches the file you have ftp'd
- email me for help at terwilliger@lanl.gov

Mailing list for SOLVE and RESOLVE

Now you can correspond with other SOLVE/RESOLVE users with this email newsgroup.

To subscribe to or unsubscribe from the mailing list, send an email message containing one of the following:

- subscribe solve
- unsubscribe solve

to:

- listmanager@listserv.lanl.gov

To send a message to everyone on the list, just send an email to

- solve@lanl.gov

SOLVE Table of Contents	Alphabetical Index

References for SOLVE

If you use SOLVE for structure determination, please cite the appropriate papers from this list and also please cite the web site "www.solve.lanl.gov". The overall paper describing SOLVE is:

Terwilliger, T.C. and J. Berendzen. (1999) "Automated MAD and MIR structure solution". Acta Crystallographica D55, 849-861.

For automated MAD structure determinations, references 1-3, 5 , 6, and 11 are appropriate; for automated MIR structure determinations, references 1-4 are appropriate.

1. Terwilliger, T. C., Kim, S.-H., and D. Eisenberg. (1987). Generalized method of determining heavy-atom positions using the difference Patterson function. Acta Cryst. A43, 1-5.

2. Terwilliger, T. C. and D. Eisenberg. (1983). Unbiased three-dimensional refinement of heavy-atom parameters by correlation of origin-removed Patterson functions. Acta Cryst. A39, 813-817.

3. Terwilliger, T. C. and D. Eisenberg. (1987). Isomorphous replacement: effects of errors on the phase probability distribution. Acta Cryst. A43, 6-13.

4. Terwilliger, T. C. and J. Berendzen (1996) Correlated phasing of multiple isomorphous replacement data. Acta Cryst. D52, 749-757.

5. Terwilliger, T. C. (1994). MAD phasing: treatment of dispersive differences as isomorphous replacement information. Acta Cryst. D50, 17-23.

6. Terwilliger, T. C. (1994) MAD phasing: Bayesian estimates of Fa. Acta Cryst. D50, 11-16.

7. Terwilliger, T. C. and J. Berendzen (1995). Difference refinement: a method for estimating differences between related structures. Acta Cryst. D51, 609-618.

8. Terwilliger, T. C. and Berendzen, J. (1996). Bayesian difference refinement. Acta Crystallographica section D52, 1004-1011.

9. Terwilliger, T. C. and Berendzen, J. (1996). Bayesian weighting for macromolecular crsytallographic refinement. Acta Cryst. D52, 743-748.

10. Terwilliger, T. C. and Berendzen, J. (1996). Correlated phasing in multiple isomorphous replacement. Acta Cryst D52, 749-757.

11. Terwilliger, T. C. and Berendzen, J. (1997). Bayesian MAD phasing. Acta Cryst. D53, 571-579.

Back to RESOLVE table of contents

RESOLVE references

1. T. C. Terwilliger (1999) "Reciprocal-space solvent flattening," *Acta Crystallographica* **D**55,1863-1871.
2. T. C. Terwilliger (2000) "Maximum likelihood density modification," *Acta Cryst.* D**56**, 965-972
3. T. C. Terwilliger (2001) "Maximum-likelihood density modification with pattern recognition of structural motifs" *Acta Cryst.*, **D**57, 1755-1762
4. T. C. Terwilliger (2001) "Map-likelihood phasing" *Acta Cryst.*, **D**57, 1763-1775
5. Terwilliger, T. C. (2004) Removing model bias in macromolecular crystallography. (in preparation)
6. Terwilliger, T. C. (2002). Statistical density modification with non-crystallographic symmetry. Acta Cryst. D58, 2082-2086.
7. Terwilliger, T. C. (2002). Rapid Automatic NCS identification Using Heavy-Atom Substructures. Acta Cryst. D58, 2213-2215.
8. Terwilliger, T. C. (2002). Automated main-chain model-building by template-matching and iterative fragment extension. Acta Cryst. D59, 34-44.
9. Terwilliger, T. C. (2002). Automated side-chain model-building and sequence assignment by template-matching. Acta Cryst. D59, 45-49.
10. Terwilliger, T. C. (2002). Improving atomic models at moderate resolution by automated iterative model-building and refinement. Acta Cryst. D59, 1174-1182.
11. Terwilliger, T. C. (2003) Statistical density modification using local pattern matching. Acta Cryst. D59, 1688-1701.

Back to RESOLVE table of contents

Introduction to RESOLVE

- ***Why another density modification approach?***
- ***Problems with the phase recombination approach to density modification.***
- ***The statistical approach to density modification***
- ***Using all the available information for density modification***
- ***Carrying out density modification with RESOLVE***
- ***Removing model bias with prime-and-switch phasing***
- ***NCS averaging in RESOLVE***
- ***Local pattern matching in RESOLVE***
- ***Fragment identification in RESOLVE***
- ***Automated model-building and iterative model-building in RESOLVE***
- ***Automated fitting of flexible ligands to an electron-density map***
- ***Merging NCS-related copies of a model in RESOLVE***

Why another density modification approach?

Although density modification (solvent flattening, non-crystallographic symmetry, phase extension, histogram matching, etc.) has been a very powerful tool, its potential is much greater than has been achieved so far. There are two reasons for this:

- The statistical basis of density modification has not been well developed
- The range of potential information included in density modification has not been fully utilized.

Problems with the phase recombination approach to density modification.

RESOLVE uses a statistical approach to density modification, while other methods use an approach in which a map is modified to meet expectations and the new phases are recombined with experimental phases. For the mathematical details, see the references for RESOLVE . You might also wish to see the discussion and extensions in Kevin Cowtan's article "Gaussian Likelihoods in real and reciprocal space" in the CCP4 newsletter.

Principal problems with the phase recombination method

What is the optimal relative weighting of modified and experimental phases?	Incorrect relative weighting means that the final results will not be optimal Incorrect weighting terms mean that the final figures of merit are almost always inflated
When do you stop iterating?	In some approaches the maps initially get better, then get worse unless you stop

---

A statistical approach to density modification

Density modification can be thought of as a way to adjust crystallographic phases (or amplitudes) to make them simultaneously consistent with the experimental data and with our expectations of what an electron density map should look like. The statistical approach is a mathematical way to formulate this statement. By using this formulation, the weighting factors and problems with convergence are taken care of automatically.

In RESOLVE, any set of structure factor amplitudes and phases has an associated probability composed of two simple parts:

Probability of a set of phases (and amplitudes)

The probability of the experimental phases	This is the probability that you would have observed your experimental data if this set of phases (and amplitudes) were correct
The lprobability of the map	This is the probability that the electron density map calculated from this set of phases is drawn from the set of plausible electron density maps for this structure

RESOLVE adjusts your crystallographic phases so as to maximize the total (posterier) probability of those phases. The mathematics is a little complicated but the idea is very simple. To see the mathematics in detail, have a look at T. C. Terwilliger (2000) "Maximum likelihood density modification," *Acta Cryst.* D**56**, 965-972.

Note on terminology: The approach used by resolve is now called "Statistical density modification," a name suggested by Kevin Cowtan. It used to be called "Maximum-likelihood density modification", using the term "likelihood" in a colloquial sense of probability. The old name (as pointed out by Gerard Bricogne and others) is confusing because the maximum-likelihood method is a specific technique that uses a specific definition of "likelihood" that is not used in this approach. Sorry to all for the confusion, and hoping that it will now be more clear. The mathematics remains exactly the same.

Using all the available information for density modification

Density modification is usually thought of as a process that is carried out on an experimental electron density map prior to model building, but iterative model-building methods such as ARP/wARP can also be thought of as density modification techniques. With the statistical approach, partial model information can be seamlessly incorporated into the total expression for the probability of the phases. This allows a hierachical approach to incorporating information about phase probability:

Types of information that can be used in statistical density modification

Experimental phases (if available)

Low-resolution structural information (solvent boundary)

Non-crystallographic symmetry

Partial model information (molecular replacement or model building)

Full atomic model information

The current version of RESOLVE can incorporate all of these types of information.

Carrying out density modification with RESOLVE

RESOLVE carries out density modification on several levels:

Each "mask cycle":

RESOLVE estimates the probability that each point in the map is within the protein or solvent region (a probabilistic "mask")
RESOLVE refines NCS symmetry operators, if present
RESOLVE then carries out one or more minor cycles:

Fitting of the histogram of density in the protein and solvent regions to model histograms (yielding beta = quality of this fit, and sigma= the overall error in the map)
Estimation of target density (a probility function)at each point based on these histograms for solvent and protein regions
Estimation of target density and uncertaingy at each point from NCS or a model map, if present
Calculations of derivatives of map probability with respect to phases
Estimation of phase probability from experimental phase probabilities and the map probability function

RESOLVE carries out mask cycles (up to 5) until no further changes occur in the phases.

If NCS is present, then RESOLVE carries out an initial mask cycle, not including any NCS, to estimate uncertainties in density estimated from NCS copies. Then RESOLVE carries out another initial mask cycle, using NCS but not solvent flattening, to estimate "sigma", the overall error in the map.

If "use_input_solv" is not set and "hklstart" is not specified, then RESOLVE uses the R factor to estimate the solvent content of the crystal. Solvent contents from 0.1 to 0.9 are tested, and the value leading to the minimum R is chosen. This optimal solvent content is written to the file "resolve.solvent." Note: if "use_input_solv" is specified, then RESOLVE assumes that the solvent content is already known and reads it from "solvent_content" if specified, or else from "resolve.solvent" if present, or else the default (0.40) is used.

RESOLVE also uses the R-factor to identify which histogram of solvent densities and protein densities to use in density modification. The file "rho.list" in $SOLVEDIR/segments/ contains several histogram profiles, all based on model electron density maps. These are at resolutions from 1.2 A to 4 A. RESOLVE carries out a test of each histogram initially and chooses the one leading to the lowest R factor. The histogram can be set using "database". The optimal database entry is written to "resolve.database".

Resolve estimates the optimal smoothing radius using a simple formula. For cycles where no density modification has occurred yet (first cycle normally, unless "phases_from_resolve" has been set), R is set with the equation: R=2.41 (dmin)**0.9 (fom)**-0.26. For all other cycles (after density modification has begun), the smoothing radius is 4 A. These can also be set with

"wang_radius", "wang_radius_cycle", "wang_radius_start", or "wang_radius_finish".

If "n_restore" is set by the user to be non-zero (default = 0), then after the phases have converged, the whole process is repeated again, starting with the original phases, but using the current probabilistic solvent mask. This allows an optimized mask to be used in the "first" cycle of density modification.

Removing model bias with prime-and-switch phasing

Electron density maps obtained using phases calculated from atomic models often show peaks at the coordinates of atoms in the models, even when those atoms are incorrectly placed. This effect can be reduced by careful weighting such as can be accomplished by Randy Read's SIGMAA approach, but it cannot be eliminated unless the phases are changed.

Prime-and-switch phasing is a way to remove model bias by using statistical density modification, ***but without including the phase information coming from the model*** once an initial map has been calculated.

The basic procedure is simple:

- Use the best existing amplitudes, phases, and weights to calculate a map
- Identify the probability that each point in the map is in solvent/macromolecule/NC-symmetry regions, etc
- Identify the expected distribution of electron density for points in each class (solvent/macromolecule etc)
- Calculate the log-probability of this map
- Identify how the log-probability of the map would change if the phases were changed
- Adjust the phases to maximize the log-probability of the map

The initial biased phase information from the model is required to get the procedure going. The final phases are essentially unbiased by the model because they are based on the features of the map, not on the prior phase probabilities.

The final phases are generally improved the most when:

- The starting phases are accurate (even if they are biased!)
- There is substantial solvent (25% is enough, the more the better)
- The data are accurate and high resolution (3 A is fine, the higher resolution the better provided there is accurate starting phase information at the high resolution limit)

There are some ways that prime-and-switch phasing can have residual bias:

- If there is low solvent content and not enough cycles are carried out (prime-and-switch phasing converges more slowly for low solvent content)
- If the starting model is highly refined and not enough cycles are carried out (if the model is refined, then the phases have already been adjusted to minimize the density in the solvent region, and prime-and-switch phasing converges more slowly)

There are some cases where prime-and-switch phasing does not yield a nice-looking map

- Usually these are cases where the model-based phases were very inaccurate (though they might have made a nice-looking biased map)
- The estimated corrected figure of merit output by RESOLVE will generally be very low in these cases, so you know that there just was not enough phase information

---

NCS averaging in RESOLVE

Non-crystallographic symmetry is an important source of information about the probabiltiy of an electron density map. RESOLVE can begin with transformation matrices and an estimate of the center-of-mass of molecule 1 that you input. RESOLVE can also figure out the transformations and center-of-mass automatically from the NCS in heavy-atom sites in a PDB file (if the default file "ha.pdb" exists and you don't specify NCS transformations, RESOLVE will try to find the NCS in those sites). RESOLVE can figure out the region over which to apply the NCS relationships automatically. You can help it by restricting the region to search for NCS with the keyword "ncs_domain_pdb xxxx.pdb" and supplying a PDB file that contains dummy atoms in the region where NCS exists (all copies must be supplied in the PDB file).

See also the sample script at resolve_sample_scripts

- RESOLVE uses NCS information in the following way (see Terwilliger, T. C. 2002 "Statistical density modification with non-crystallographic symmetry". Acta Cryst. D58, 2082-2086 and Terwilliger, T. C. (2002). "Rapid Automatic NCS identification Using Heavy-Atom Substructures" Acta Cryst. D58, 2213-2215.)
- If there is more than one type of NCS relationship in the crystal, RESOLVE can carry out NCS averaging separately for each group of molecules related by an NCS relationship. If you use this, you should specify the maximum extent of the region over which to apply each NCS relationship with the ncs_domain_pdb command. In that case RESOLVE can refine the NCS operators using only the part of the molecule that you specify (and not be confused by some other region that is part of another NCS group). If the regions for several NCS groups overlap, the NCS group that will be used for the overlapping points in NCS averaging will be whichever NCS group has the higher NCS correlation near those points.

  - Use the best existing amplitudes, phases, and weights to calculate a map
  - Identify the region near the center-of-mass of molecule 1 for which the NCS transformations (on average) result in significant correlation
  - Define the asymmetric unit of NCS such that all points in it are close together and within the region of significant correlation, and such that no point in it is equivalent to any other point either by crystallographic or non-crystallographic symmetry.
  - Estimate the overall correlation among NCS-related molecules as a function of position in the NCS asymmetric unit (so that the edges might be allowed to vary more than the centers, for example)
  - Map all N copies of the NCS asymmetric unit on to identical grids
  - Generate target density for each molecule using all the other N-1 copies only
  - Map all the target densities back onto the original asymmetric unit of the crystal and use agreement between target density and the map as part of the probability of the map

---

Local pattern matching in RESOLVE

RESOLVE can use the local patterns of density in your electron density map in statistical density modification to improve crystallographic phases. The basic idea is that on a local level (within a sphere of radius 2 A) there are patterns of electron density that are associated with high density at the center of the pattern, and other patterns associated with low density at the center. RESOLVE goes through your electron density map, and at each point it compares the nearby density with a set of 20 templates (it does not use the density at the point of interest or right around it in this analysis). RESOLVE_PATTERN uses this analysis to come up with a new estimate of the density at each point in the map. This new estimate of density (the "image") has the remarkable property that errors in the image are almost uncorrelated with errors in the map used to create it. This means that phase information from the "image" can be combined with phase information from other sources in a simple way. You can see the details of all this in Terwilliger, T. C. (2003) Statistical density modification using local pattern matching. Acta Cryst. D59, 1688-1701.

The resolve_build script below uses image-based phasing. Image-based phasing is the use of an electron density map that

typically comes from either an atomic model or from pattern-matching or from NCS, along with observed values of FP, to estimate phases. The process results in phases and figures of merit similar to those obtained with Randy Read's SIGMAA, but the values come directly from map-probability phasing. The electron density map provided is used as a target for statistical density modification: crystallographic phases are found that, when combined with observed amplitudes, give a map that is as close as possible to the target map. The figures of merit reflect how precisely each phase can be determined using this approach. The phases from image-based phasing are not the same as those from an FC calculation and they are not always unimodal like FC, SIGMAA or Sim-weighted phases.

Fragment identification in RESOLVE

RESOLVE can carry out an FFT-based search for fragments of structure (currently helices, strands), refine the locations of these fragments, and use them in density modification even if a complete model cannot be built. The approach to finding fragments ("Maximum-likelihood density modification with pattern recognition of structural motifs",Terwilliger, T. Acta Cryst D. 57, 1755-1762; 2001) is very similar to Kevin Cowtan's FFT-based search (Cowtan, K., Acta Cryst D54, 750-756, 1998). A template consisting of averaged helical density (or strand density) is rotated over a range of orientations designed to cover most possibilities within about 20 degrees and an FFT convolution is carried out for each orientation to find locations where the template and map match. The best matches are identified and the orientiations and positions are refined. Then a pseudo-map is constructed consisting of the original templates, oriented based on the refined positions found in the search, and weighted by the local correlation coefficient. This pseudo-map is used as a source of phase information through map-probability phasing (Map-likelihood phasing", Terwilliger, T., Acta Cryst., D57, 1763-1775). This approach is similar to the one described in the original publication ("Maximum-likelihood density modification with pattern recognition of structural motifs",Terwilliger, T. Acta Cryst D. 57, 1755-1762; 2001) but works much better than the original method.

Fragment identification is normally carried out right after model-building because the same FFT search can be used for both. The resolve build script includes it.

Automated model-building and iterative model-building in RESOLVE

After the completion of density modification, RESOLVE builds a model of your structure. For versions 2.02 and higher, the model needs sequence information from you. You specify a file with the keyword "seq_file" and RESOLVE expects a sequence of amino acids in 1-letter format. If there are more than one type of chain, RESOLVE expects them separated by a line containing ">>>". . Typically RESOLVE can build 70-90% of the residues for a good map at 2-3 A resolution. You can tell if the model is correct by noting how good the match is to the sequence and by noting the NCS correspondence among chains (if NCS exists). The PDB file that RESOLVE writes out will have the model and also as HETATM records at the end with the heavy atom sites from SOLVE output file ha.pdb.

You can read all the details about RESOLVE automated model-building in Terwilliger, T. C. (2002). Automated main-chain model-building by template-matching and iterative fragment extension. Acta Cryst. D59, 34-44 and Terwilliger, T. C. (2002). Automated side-chain model-building and sequence assignment by template-matching. Acta Cryst. D59, 45-49.

RESOLVE now has superquick model building! The standard RESOLVE model-building for version 2.05 and higher is about 3 times faster than earlier versions. This is made possible by a more selective choice of which fragments to consider extending (no need to work on a fragment that covers a region that is already built). Versions 2.05 and higher also have the option of "superquick_build" which is about 10 times faster than previous versions of RESOLVE model-building. For a very good map (one where RESOLVE can build >80% of the model) superquick_build typically gives almost the same model as the standard build. For a moderate-quality map, the standard build or even the "thorough_build" may give up to 10% more model built.

RESOLVE versions 2.05 and higher include cycles of model-building in which the thresholds for fit of the model to the map are sequentially lowered. This allows much more of the model to be built, while keeping the accuracy of most of the model high. You

can use "aggressive_build" to try and build as much as possible, or "conservative_build" to build only the best parts.

RESOLVE versions 2.06 and higher include the capability of identifying fragments (helices; strands) in a map and including them in density modification

RESOLVE builds a model in the following way.

- Use the best existing amplitudes, phases, and weights to calculate a map
- Identify locations of helices and strands using an FFT-based correlation search with a standard set of helix/strand templates
- Using a library of actual helical/beta templates, find the best match to density near each helix/strand
- Trim the templates down to match the density
- Extend the templates using a template library of short fragments
- Assemble fragments into longer fragments
- Match side-chain density to library of side-chain densities, get probability of each possible sequence alignment, choose those with very high probability
- Map all the fragments to one asymmetric unit so that they are as close together as possible
- Write out PDB file with the fragments as a main-chain model.
- Note: RESOLVE writes out "resolve.mtz" before it starts building the model, so if you don't want to wait, your phases are already ready for you.

RESOLVE (versions 2.06 and higher) can carry out pattern identification, fragment identification, density modification, and iterative model-building and refinement in combination with refmac5 (versions 5.1.24 and higher only!)

- In the first cycle, resolve carries out density modification and builds a model as usual. The model is refined and extended as much as possible. Then the current electron density map is searched for local patterns and for fragments (helices/strands), and maps are created based just on these features.
- On the next cycles, resolve uses the pattern and fragment maps, along with a map created from all models built so far, to generate model density for the asymmetric unit. This density is used along with solvent flattening, NCS, and histogram matching in the next cycle of density modification.
- Additionally, a prime-and-switch composite omit map is created in which all the above information is used except that in each "omit" region of the map, the model-based information is left out; and the omit regions are then spliced together to form a composite omit map. This omit map is used in identifying the "patterns" for the next cycle so as to minimize bias in this step.
- RESOLVE then builds a new model based on the combined phase information. It also uses fragments from the last model as candidates for parts of the new model.
- The process is iterated as long as desired (typically 5-10 cycles are plenty). If the model is not as complete as desired after about 10 cycles (i.e., R-factor > .40) then the model is used in model rebuilding with phase information. This is just like model rebuilding (below) except that experimental phase information is included throughout.
- You can use a standard script " resolve_build.csh " to carry this out.

RESOLVE (versions 2.03 and higher) can also carry out iterative model-rebuilding. This is like model-building except that you start with just a model of some kind and measured amplitudes and resolve does everything from there. This works much more slowly than model-building with experimental phases.

- Rebuilding can be carried out either with or without composite omit maps. Omit maps are recommended for rebuilding of a model with possible model errors.
- Each cycle, RESOLVE starts with input model and create an electron density map (image). Then resolve uses this map as a target in statistical density modification along with the measured FP to estimate phases. Then these phases are used as the starting phases (but not probabilities) in a cycle of density modification including (1) any experimental phase probabilities, (2) solvent flattening, histogram matching, NCS, and (3) model density based on a composite of the last 20 models.
- RESOLVE then builds a model, the model is refined with refmac5, and the model is then extended and rebuilt. On each cycle that is not an omit cycle, RESOLVE uses fragments from the previous model along with fragments identified from

the map itself as possibilities for constructing the new model.
- This process is repeated (typically 100 cycles)
- You can use a standard script " resolve_build.csh " to carry this out (it is the same script as for autobuilding). It can take a long time for rebuilding!

RESOLVE_BUILD (versions 2.06 and higher) can automatically evaluate a model, given a set of amplitudes FP (and phases PHIB and FOM if available). First RESOLVE will rebuild the model (to reduce any bias due to refinement). Then RESOLVE will calculate a prime-and-switch composite omit map (as used in rebuilding) based on the rebuilt model and any phase information you give it. Then RESOLVE will compare the original model to this map and summarize the fit for you.

RESOLVE (versions 2.08 and higher) can carry out fitting of FLEXIBLE LIGANDS to an electron density map. The only inputs needed are an electron density map (or difference map), and either just one (recommended) or else 5-10 copies (ok also) of the ligand in random but stereochemically ideal conformations in a PDB format file. The routine will figure out the allowed bond rotations from the copies of the ligand, and then will fit the ligand into the density starting with the biggest rigid part of the ligand. Parts of the ligand that do not fit are built as reasonably as possible, but may be built out of density or may be left off.

You can use the sample script resolve_ligand_fit.com script which allows you to find one or more than one copy of a ligand in a map.

You can even take a list of PDB files containing different ligands, fit each one to your map, and score them to identify which ligand may be bound, using the sample script resolve_ligand_id.com.

See the additional descriptions in resolve_sample_scripts too.

Also see the list of resolve keywords for additional options.

Thanks to Herb Klei for emphasizing the need for ligand fitting and for suggesting the idea of first finding the biggest fixed part of the ligand and then building the rest from this core!

RESOLVE (versions 2.08 and higher) will automatically merge NCS-related copies of your model during iterative model-building and refinement. The merging is done in the "extend_only" mode of model-building. An mtz file with FP PHIB FOM, a model (with >1 NCS copy) and a coordinate file with positions of atoms or pseudo-atoms (ha_file) used to deduce the NCS relationships are read in. The coordinates of each NCS-related copy are placed at all NCS-related positions, merged (if possible) and then are extended if possible into the density. If you do not want this to be done, use the flag no_merge_ncs_copies. You can merge models yourself with RESOLVE too: use the extend_only flag for model-building and include your model with: pdb_in your-current-model . Note that you need to supply an mtz file with a map to do this. You can specify the keywords trim or no_trim to tell RESOLVE to trim the resulting model back to the density or not.

Back to RESOLVE table of contents

Sample scripts for resolve

The minimal script for running RESOLVE (on experimental data)
Keywords for running RESOLVE with a different input file
Running RESOLVE with different numbers of cycles or resolution
Running RESOLVE with NCS
Prime-and-switch phasing starting from an MR model
Iterative model-building (starting from solve.mtz, solve_image.mtz or an MR model)and refinement with RESOLVE and refmac5
Density modification using local patterns
Adding HL coefficients to a file
Adjusting origin of phase set to match another phase set
Merging phase sets with different origins
Fitting a loop (with fixed ends)
Fitting flexible ligands to difference electron density maps

The minimal script for running resolve: (Some sample data and input files are located the library directory where SOLVE and RESOLVE are installed, usually located at /usr/local/lib/solve/solve-2.03/lib/examples_resolve/)

```
#!/bin/csh
#
# Here is a minimal script to run resolve:
#
# Set CCP4 variables for symmetry information and
# for file handling:
#
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
setenv CCP4_OPEN UNKNOWN
#
# Now run resolve:
#
resolve<<EOD
solvent_content 0.4             ! your solvent content goes here. Next line is your
sequence file
seq_file protein.seq
EOD
```

- RESOLVE assumes "solve.mtz" as input with FP PHIB FOM HLA HLB HLC HLD
- Output is "resolve.mtz" with FP PHIM FOMM HLAM HLBM HLCM HLDM (M is for modified)
- You need to tell RESOLVE the solvent content (approximately)

- If you want to save time and do not want a model, specify the keyword: no_build

- The sequence file for your protein should look like:

```
MIVLTVHYSSEGILV   [put the sequence of chain type 1 here, up to 80 characters per line]
>>>      [this defines the end of chain 1]
MKLVERWISSTV      [put the sequence of chain type 2 here, up to 80 characters per line]
```

NOTE: just input 1 copy of each unique chain

Keywords for running RESOLVE with a different input file:

```
hklin other.mtz
LABIN FP=F PHIB=PHI FOM=W HLA=HLA HLB=HLB HLC=HLC HLD=HLD
hklout resolve.mtz
solvent_content 0.4             ! your solvent content goes here. Next line is your
sequence file
seq_file protein.seq
```

- The input file is named with HKLIN
- The column names can be specified with the LABIN keyword
- The output file can be specified with the HKLOUT keyword.

Running RESOLVE with a different number of cycles or changing resolution:

```
mask_cycles 4                   ! number of cycles of solvent mask generation
(default = 5)
minor_cycles 3                  ! cycles of iteration for each mask cycle (default =
10)
solvent_content 0.4             ! your solvent content goes here. Next line is your
sequence file
seq_file protein.seq
mask_cycles 4                   ! number of cycles of solvent mask generation
resolution 20 3.5               ! You can limit the resolution if you want
```

- You might want fewer mask cycles to save time if the phase change in the last cycles is very small.
- Alternatively you might want more minor cycles for each mask cycle if the phase change in the last minor cycles of each mask cycle is not very small.
- You can limit the resolution if you want. Currently, RESOLVE does not fill in reflections that are not in the input file. RESOLVE will fill in phase information for any reflections that it reads in that have a non-zero structure factor amplitude

Running RESOLVE with NCS

You can have RESOLVE identify NCS from atoms in a PDB file:

```
ha_file ha.pdb
```

You can also generate an appropriate file for molecular replacement cases by selecting 3 equivalent atoms in each molecule and creating a PDB file with just these 3*N atoms.

In some cases the NCS correlation in the starting map is very low and NCS will be rejected. You can sometimes override this with the keyword

```
force_ncs
```

which will use NCS even if the correlation is very low. Also if you know the number of NCS operators is 6 you might want to try

```
ncs_restrict 6
```

Alternatively, you can enter NCS operators directly to resolve. In this case you also need to enter an estimate of the center-of-mass of molecule 1.

You need to input N sets of rotation matrices/translations/centers-of-mass, 1 for each copy in the asymmetric unit. Start with the identity for molecule 1. You can conveniently get all the numbers you need from the CCP4 program lsqkab if you have a model that you are starting with. The rotation/translation matrices and the center-of-mass are all input in a form appropriate for operating on orthogonal Angstrom coordinates (not fractional coordinates).

```
        rota_matrix  .92  .01  .33         !rotation matrix for molecule j ->molecule 1
        rota_matrix -.01 -.99  .03
        rota_matrix  .05  .10 -.95
        tran_orth    .50  .00 .00          ! translation for molecule j -> molecule 1
        center_orth   25.  39. 44.         !  center of mass, molecule j

        fraction_ncs 0.15                  ! fraction of the asymmetric unit in 1 copy of
NCS
                        ! (only needed if it is not equal to (fraction protein)/(number of
NCS copies)
```

- If you have more than one type of domain/molecule to use in NCS, then you can specify them separately. Before each new group of NCS relationships, just enter the keyword

  ```
  new_ncs_group
  ```

  If you are using multi-domain averaging then it is best if you can specify the maximum extent of the region over which each set of NCS operators is to be applied. You can do this for each ncs group by specifying

  ```
  ncs_domain_pdb  domain_1.pdb
  ```

  where domain_1.pdb has dummy atoms that cover the region of this domain. Only the region within domain_1.pdb that actually has some NCS correlation will be used. You can specify the radius of spheres to be drawn around each of these atoms in constructing the region with the keyword

```
rad_mask 2.5
```

- RESOLVE automatically identifies a mask for NCS, but requires an estimate of the center of molecule 1, in orthogonal A units ("center_orth")
- RESOLVE expects rotation matrices ("rota_matrix") and translation vectors ("trans_orth") in orthogonal Angstrom units that map each of the N molecules onto molecule 1 in the asymmetric unit. This is the form produced by the CCP4 program lsqkab with xyzin1 (reference)= molecule 1 and xyzin2 (working)= molecule j. If you have transformations that go from molecule 1 to molecule N, you can specify "invert" to invert the transformations.
- Input the matrices first (three lines of "rota_matrix" for each matrix); then the translation vector
- Input the identity as the first rotation matrix, with a zero translation vector

For at least the first molecule, you need to enter an estimate of the center of the molecule ("center_orth"). If you enter centers for others, resolve will verify that the centers match the positions expected from the symmetry you input

Prime-and-switch phasing starting from a SIGMAA-weighted map (see also script above starting right from an MR model)

To carry out prime-and-switch phasing starting with a model and SIGMAA phases: *(Sample data and input files are located the library directory where SOLVE and RESOLVE are installed, usually located at /usr/local/lib/solve/solve-2.06/lib/examples_resolve/)*

1. Run Randy Read's SIGMAA program to get as unbiased a starting map as possible into "sigmaa.mtz"
2. Go to the directory where you ran sigmaa and type (or put in a command file):

```
#!/bin/csh
# Here is a very minimal script to run prime-and-switch phasing
#
# Set CCP4 variables for symmetry information and
# for file handling:
#
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
setenv CCP4_OPEN UNKNOWN
#
# Now run prime-and-switch phasing:
#
resolve<<EOD
hklin sigmaa.mtz
labin FP=FP FC=FC PHIC=PHIC FOM=WCMB FWT=FWT
hklout ps.mtz
solvent_content 0.4             ! your solvent content goes here. Next line is your
sequence file
seq_file protein.seq
prime_and_switch
EOD
#
# Now "ps.mtz" has the output amplitudes, phases,
# figure of merit and HL-coeffs in columns labelled: FP PHIM FOMM HLAM HLBM HLCM HLDM
#
```

Note: The least biased map from SIGMAA is FWT exp(i PHIC), so RESOLVE works best starting with FWT from SIGMAA. However, you can leave out FWT if you wish. You need FP FC PHIC at a minimum.

- Output is "resolve.mtz" with FP PHIM FOMM HLAM HLBM HLCM HLDM (M is for modified)

Iterative model-building with RESOLVE and refmac5

You can carry out model-building and refinement iteratively with RESOLVE and refmac5 . A script that you can edit and use is resolve_build.csh. This is convenient for starting with solve.mtz (right from SOLVE)

You can also carry out iterative model-rebuilding (starting from an MR model, for example) with resolve_build.csh.

Adding HL coeffs

You can convert from PHI FOM to PHI FOM HL ABCD in an approximate way with this script:

```
hklin phases.mtz
labin FP=FP PHIB=PHIB FOM=FOM
scale_refl 0.  ! just use prior phase information
mask_cycles 1
minor_cycles 1
solvent_content 0.5  !any number, not used but needed
no_build
hklout phases_hl.mtz
```

Adjusting origin to match another phase set

You can apply an offset of the origin for a dataset to match the origin in another phase set. It uses an FFT convolution to find the best allowed offset between the maps calculated with the two phase sets. Unlike merging two phase sets (below) HL coefficients are optional. Note: the FP and SIGFP in the output file will come from the hklin file NOT the hklmerge file.

```
hklin random_1_hl.mtz   ! File with target map (fixed)
labin FP=FP PHIB=PHIM FOM=FOMM
hklmerge random_2_hl.mtz  ! File to be offset
labmerge FP=FP PHIB=PHIM FOM=FOMM HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM
hkl_offset_file resolve_off.mtz  ! The hklmerge file, offset all by itself
```

Merging phase sets with different origins

You can merge phase sets with different origins with this script. It uses an FFT convolution to find the best allowed offset between the maps calculated with the two phase sets. Merging the files REQUIRES HL coefficients (see above for generating them if you do not have them). Note: the FP and SIGFP in the output file will come from the hklin file NOT the hklmerge file:

```
hklin random_1_hl.mtz   ! File with target map (fixed)
labin FP=FP PHIB=PHIM FOM=FOMM HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM
hklmerge random_2_hl.mtz  ! File to be offset
labmerge FP=FP PHIB=PHIM FOM=FOMM HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM
hkl_offset_file resolve_off.mtz  ! The hklmerge file, offset all by itself
```

```
hkl_merge_file resolve_merge.mtz  ! The hklmerge file offset, then merged with hklin
```

Fitting a loop (with fixed ends)

You can now fit a loop if you know the ends. Put the two amino acids before the loop and the two after the loop in a pdb file (loop_ends.pdb) and put all the rest of your model in another PDB file (all_but_loops.pdb) that will be used as a mask to prevent building where your model exists. Then run the script below, specifying the number of residues in the loop, including the 2 on each end that you are supplying.

```
hklin resolve.mtz
labin FP=FP PHIB=PHIM FOM=FOMM
pdb_in loop_ends.pdb     ! Just 2 residues at beginning of loop, 2 at end.
seq_file seq.dat
extend_only
loop_only                ! just fit the loop
model all_but_loop.pdb
build_outside_model      ! don't build in region defined by model
no_sub_segments          ! fit the whole loop to sequence, not parts
n_random_loop 20
loop_length 8            ! exact length of loop, including ends in pdb_in
                         ! you can leave it out if unknown
rms_random_loop 0.3      ! how much to vary each try of building the loop
EOD
```

Fitting flexible ligands to difference electron density maps

You can fit a flexible ligand into a difference electron density map.

You have two choices of input for the ligand. You can either input a single copy (preferred method) and the program will guess acceptable configurations, or you can make about 10 copies of the ligand in different allowed conformations and put them all into a one PDB file.

In either case, RESOLVE will identify the rigid parts of the molecule and all allowed relationships between them from this group of structures. It will then build the ligand into the electron density map. See the list of resolve keywords for additional options (i.e., delta_phi_ligand...)

```
hklin difference_density.mtz   ! File with map
labin FP=FP PHIB=PHIM FOM=FOMM
ligand_file ligand_in_1_or_10_conformations.pdb
```

That's it. RESOLVE will fit the ligand into the density and write it out to "ligand_fit.pdb" along with an analysis of the fit.

If you want to input an experimental map and a model for most of the structure you can add the keyword model:

```
model protein_model.pdb
```

and RESOLVE will subtract off the density from the model and just fit the remaining (ligand) density.

For more information on ligand fitting, see the introduction in ligand fitting.

A script you can edit is resolve_ligand_fit.com.

To score many ligands against your map, use resolve_ligand_id.com.

If you want to calculate a map with difference density for the ligand, then use resolve_difference script. This script starts with an mtz file containing native amplitudes (FP) and a PDB file containing a model for the structure without the ligand and calculates a simple difference map. You can also use any other suitable program to calculate this map. The output file resolve_diff.mtz (with FP PHIM FOMM) can then be used for ligand fitting with resolve_ligand_fit.com (for fitting one type of ligand), resolve_ligand_id.com (for testing a list of ligands), or a simple script like:

```
hklin resolve_map.mtz   ! File with map
labin FP=FP PHIB=PHIM FOM=FOMM
ligand_file ligand_in_1_or_10_conformations.pdb
model protein_model.pdb
```

If the ligand-fitting routine does not fit your ligand well, there are some options to get it to try harder:

- You can tell resolve to screen more potential locations of fragments in the FFT convolution search (default=300) with:

  ```
  n_ligand_pos 400
  ```

- You can tell resolve to refine more potential locations of fragments in the FFT convolution search (default=100) with:

  ```
  n_ligand_pos_ref 200
  ```

- You can tell resolve to search more finely for potential locations of fragments in the FFT convolution search (default=40 degrees) with:

  ```
   delta_phi_ligand 20
  ```

- You can tell resolve to keep track of more placements (default=100) of partially-built ligands with:

  ```
  n_keep_plac 200
  ```

- Resolve will always take all the refined placements of fragments as a group and try to build from them, keeping the top n_keep_plac all the time. Resolve also tries one at a time if n_indiv_tries_min > 0. You can tell resolve to try to build starting with more individual placements (defaults min:0, max:5) of each sub-fragment with:

  ```
  n_indiv_tries_min 20
  ```

  ```
  n_indiv_tries_max 20
  ```

- You can tell resolve to try more sub-fragments as possible starting points for ligand building with:

  ```
  n_group_search 10
  ```

- You can center the search at a desired position (A) with:

  ```
  search_center  3 25 6
  ```

```
#!/bin/csh
#
#
#    Resolve ligand-fitting script
#     T. Terwilliger 05-Jan-2005
#
#    This script requires resolve version 2.08 and ccp4 version 5.0.2
#
#    Before running this script, please get map coefficients with
#    resolve_completion.com script(output is resolve_map.mtz optimized for ligand )
#    or else input map coefficients of your own in hklin and labin
#
#
setenv SOLVEDIR /usr/local/lib/solve/
setenv SOLVETMPDIR .
setenv SYMOP $SOLVEDIR/symop.lib
setenv SYMINFO $SOLVEDIR/syminfo.lib
setenv CCP4_OPEN UNKNOWN
unlimit
#
echo "Resolve ligand-fitting script version 2.08.8 of 05-Jan-2005"
echo ""
#   get N ligands in N passes through map,
#    excluding previous as we go
#
set hklin = resolve_diff.mtz
set labin = "FP=FP PHIB=PHIM FOM=FOMM"
set number_of_ligands = 1    # number of copies of ligand to find
set ligand_template = ligand_template.pdb    # file with 1-10 copies of ligand
set model = PARTIAL_MODEL.pdb    #  model without ligand or ""
set resolve = /u1/terwill/resolve/work/resolve.linux  # where is resolve
set dmin = NONE   # NONE to use all data
set dmax = NONE   # NONE to use all data
#
#  some parameters you can set if you want to...
#
set n_indiv_tries_min = 10    # usually 0-10, but set up to 100 to try harder to find soln
set n_indiv_tries_max = 10    # usually 5-10, but set up to 100 to try harder to find soln
set n_group_search = 3    # usually 3, but set up to 10 to try harder to find soln
set search_dist = 10    # usually 10  A; always at least 5; smaller speeds up search
set no_local_search = ""    # usually "";  "no_local_search" to force complete search
set delta_phi_ligand = 40    # usually 40 degree increments; set lower to search more
```

```
#
if ( $dmin == NONE || $dmax == NONE ) then
 set resolution_line = ""
else
 set resolution_line = "resolution $dmin $dmax"
 echo "Resolution limits: $dmin $dmax A"
endif
#
echo "Input mtz file with map: $hklin"
if  (! -f $hklin ) then
  echo "Sorry, the file $hklin does not seem to exist?"
  exit
endif
echo "Labin line for $hklin is $labin"
echo "Number of copies of ligand to find: $number_of_ligands"
echo "Template PDB file containing 1 to 10 copies of ligand: $ligand_template"
if  ( ! -f $ligand_template) then
  echo "Sorry, the file $ligand_template does not seem to exist?"
  exit
endif
if ($model != "") then
 echo "The model $model will be used to mask out part of the map"
 if  ( ! -f $model) then
  echo "Sorry, the file $model does not seem to exist?"
  exit
 endif
set model_use = "model ALL.PDB"
else
set model_use = ""
endif
#
echo "Location of resolve: $resolve"
if  ( ! -f $resolve) then
  echo "Sorry, the program $resolve does not seem to exist?"
  exit
endif

echo "Number of groups (fragments) to search for with FFT: $n_group_search"
echo "FFT search will be in increments of $delta_phi_ligand degrees"
if ($no_local_search != "") then
 echo "Entire map will be searched"
endif
echo "Fitting will finish when number of tries is $n_indiv_tries_max or when"
```

```
echo "all atoms are found and the number of tries is at least $n_indiv_tries_min"
#
echo ""
#-------------------------------------------------------------------------------
#-------------------------------------------------------------
#  figure out if this machine uses grep -a or just grep for text files:
echo "A" > test_a.dat
 set test_grep = `grep -a "a" test_a.dat >& tmp.dat`
if ( $status ) then
#  there was an error...do not use grep -a
 set grep_type = "grep"
else
 set grep_type = "grep -a"
endif
set test_grep = `$grep_type "A" test_a.dat`
if ( $#test_grep != 1 ) then
 echo "Sorry, unable to set the grep command on this system...giving up"
 exit
endif
rm test_a.dat
#-------------------------------------------------------------

#-------------------------------------------------------------------------------
#
if ($model != "")then
cp $model ALL.PDB
cat ALL.PDB|$grep_type 'CRYST1' > resolve_ligand.pdb
cat ALL.PDB|$grep_type 'ORIGX[1-3]' >> resolve_ligand.pdb
cat ALL.PDB|$grep_type 'SCALE[1-3]' >> resolve_ligand.pdb
#  make sure the headers exist...
@ header_lines = `cat resolve_ligand.pdb |wc -l`
if ( $header_lines >= 4 ) goto ok1
 echo "The input PDB file $model needs to have at least 4 lines of headers with "
 echo "CRYST1 SCALE1 SCALE2 SCALE3"
 echo "Yours seems to have instead $header_lines of headers"
 cat resolve_ligand.pdb
 exit
ok1:
else
 if ( -f ALL.PDB ) rm -f ALL.PDB
 if ( -f resolve_ligand.pdb ) rm -f resolve_ligand.pdb
endif
#
```

```
set cycle = 1
while ($cycle <= $number_of_ligands)
#
echo "Searching for copy $cycle of ligand..."
$resolve<<EOD
$resolution_line
hklin $hklin
labin $labin
ha_file NONE
no_build
delta_phi_ligand $delta_phi_ligand
$no_local_search
search_dist  $search_dist
$model_use   ! exclude region defined by model
ligand_file $ligand_template
ligand_resno $cycle   # label ligand with $cycle
n_indiv_tries_min  $n_indiv_tries_min
n_indiv_tries_max  $n_indiv_tries_max
n_group_search $n_group_search  ! how many groups to search for
! some more parameters that can be set:
! search_center  3 25 6 ! center the search here
! group_search  3     ! use this group in FFT search
! fit_phi_range -180 180  ! range of torsion angles to check
! fit_phi_inc   20    ! increment for torsion angle check
EOD
#
if ($cycle == 1 ) then
  cp ligand_map.mtz ligand_map_all.mtz
  cp ligand_map.map ligand_map_all.map
endif
#
cp ligand_fit.pdb ligand_$cycle.pdb
#
#  add new ligand to ALL.PDB and to resolve_ligand.pdb
#
if ( -f ALL.PDB )then
 cat  ligand_fit.pdb |$grep_type ATOM >> ALL.PDB
else
 cp  ligand_fit.pdb ALL.PDB
endif
if ( -f resolve_ligand.pdb ) then
 cat  ligand_fit.pdb |$grep_type ATOM >> resolve_ligand.pdb
else
```

```
 cp ligand_fit.pdb resolve_ligand.pdb
endif
#
@ cycle ++
end
#
cp ligand_map_all.mtz ligand_map.mtz
cp ligand_map_all.map ligand_map.map
#
echo "Evaluating fit of ligand to map...."
$resolve<<EOD>TEMP.DAT
hklin $hklin
labin $labin
$resolution_line
model resolve_ligand.pdb
evaluate_model
EOD
$grep_type 'region of model' TEMP.DAT|$grep_type Map
rm -f TEMP.DAT
echo ""
#
echo "All done...your fitted $number_of_ligands copies of ligand are in resolve_ligand.pdb"
echo "and coefficients for a map showing just the ligand regions are in ligand_map.mtz"
```

```
#!/bin/csh
#
#
#    Resolve ligand identification script
#     T. Terwilliger 05-Jan-2005
#
#    This script requires resolve version 2.08 and ccp4 version 5.0.2
#
#    Before running this script, please get map coefficients with
#    resolve_completion.com script(output is resolve_map.mtz optimized for ligand )
#    or else input map coefficients of your own in hklin and labin
#
#
setenv SOLVEDIR /usr/local/lib/solve/
setenv SOLVETMPDIR .
setenv SYMOP $SOLVEDIR/symop.lib
setenv SYMINFO $SOLVEDIR/syminfo.lib
setenv CCP4_OPEN UNKNOWN
unlimit
#
echo "Resolve ligand identification script version 2.08.5 of 05-Jan-2005"
echo ""
#
#   Test many ligands for fit to map and identify the best
#
set hklin = resolve_diff.mtz
set labin = "FP=FP PHIB=PHIM FOM=FOMM"
set ligand_list_file = ../ligand_list_1.dat    # file containing list of ligand PDB files
set model = PARTIAL_MODEL.pdb
set resolve = /u1/terwill/resolve/work/resolve.linux  # where is resolve
set dmin = 2.6
set dmax = 200
#
#  some parameters you can set if you want to...
#
set n_indiv_tries_min = 10    # usually 0 to 10, but set up to 20 to try harder to find soln
set n_indiv_tries_max = 10    # usually 5-10, but set up to 20 to try harder to find soln
set n_group_search = 3    # usually 3, but set up to 10 to try harder to find soln
set search_dist = 10    # usually 10  A; always at least 5; smaller speeds up search
set no_local_search = ""    # usually "";  "no_local_search" to force complete search
set delta_phi_ligand = 40    # usually 40 degree increments; set lower to search more
#
```

```
echo "Resolution limits: $dmin $dmax A"
echo "Input mtz file with map: $hklin"
if  (! -f $hklin ) then
  echo "Sorry, the file $hklin does not seem to exist?"
  exit
endif
echo "Labin line for $hklin is $labin"
echo "File containing list of template PDB files is $ligand_list_file"
if  ( ! -f $ligand_list_file) then
  echo "Sorry, the file $ligand_list_file does not seem to exist?"
  exit
endif
if ($model != "") then
 echo "The model $model will be used to mask out part of the map"
 if  ( ! -f $model) then
  echo "Sorry, the file $model does not seem to exist?"
  exit
 endif
set model_use = "model $model"
else
set model_use = ""
endif
#
echo "Location of resolve: $resolve"
if  ( ! -f $resolve) then
  echo "Sorry, the program $resolve does not seem to exist?"
  exit
endif

echo "Number of groups (fragments) to search for with FFT: $n_group_search"
echo "FFT search will be in increments of $delta_phi_ligand degrees"
if ($no_local_search != "") then
 echo "Entire map will be searched"
endif
@ n1 = $n_indiv_tries_max + 1
@ nn = $n_indiv_tries_min + 1
echo "Fitting will finish when number of tries is $n1 or when"
echo "all atoms are found and the number of tries is at least $nn"
#
echo ""
#-------------------------------------------------------------------------------
#-------------------------------------------------------------
#  figure out if this machine uses grep -a or just grep for text files:
```

```
echo "A" > test_a.dat
 set test_grep = `grep -a "a" test_a.dat >& tmp.dat`
if ( $status ) then
#  there was an error...do not use grep -a
 set grep_type = "grep"
else
 set grep_type = "grep -a"
endif
set test_grep = `$grep_type "A" test_a.dat`
if ( $#test_grep != 1 ) then
 echo "Sorry, unable to set the grep command on this system...giving up"
 exit
endif
rm test_a.dat
#-------------------------------------------------------------

#-------------------------------------------------------------------------------
#
if ($model != "")then
#  make sure the headers exist...
@ header_lines = `cat $model |wc -l`
if ( $header_lines >= 4 ) goto ok1
 echo "The input PDB file $model needs to have at least 4 lines of headers with "
 echo "CRYST1 SCALE1 SCALE2 SCALE3"
 echo "Yours seems to have instead $header_lines of headers"
 cat resolve_ligand.pdb
 exit
ok1:
endif
#
#  make sure all the ligand files in $ligand_list_file exist
set count = 0
set keep_going = 1
set lines = `cat $ligand_list_file |wc -l`
echo "Number of files in $ligand_list_file is $lines"
set count = $lines
while ($count > 0)
 set file = `tail -${count} $ligand_list_file |head -1 `
 #echo "File: $file"
 if ( ! -f $file ) then
  echo "Sorry the file $file from $ligand_list_file does not exist?"
  exit
 endif
```

```
@ count--
end
#
echo "Evaluating all ligands in $ligand_list_file now...and placing fitted"
echo "ligand xx in resolve_ligand_xx.pdb"
echo ""
echo " LIGAND    ATOMS PLACED      CC          SCORE        TEMPLATE FILE"
set count = $lines
set count_up = 0
while ($count > 0)
@ count_up++
 set file = `tail -${count} $ligand_list_file |head -1 `
 #echo "Evaluating ligand $file"
#
$resolve<<EOD   > resolve_fit_id_$count_up.log
resolution $dmax $dmin
hklin $hklin
labin $labin
ha_file NONE
no_build
delta_phi_ligand $delta_phi_ligand
$no_local_search
search_dist  $search_dist
$model_use   ! exclude region defined by model
ligand_file $file
ligand_resno 1 # label ligand with
n_indiv_tries_min  $n_indiv_tries_min
n_indiv_tries_max  $n_indiv_tries_max
n_group_search $n_group_search  ! how many groups to search for
! some more parameters that can be set:
! search_center  3 25 6 ! center the search here
! group_search  3     ! use this group in FFT search
! fit_phi_range -180 180  ! range of torsion angles to check
! fit_phi_inc   20    ! increment for torsion angle check
EOD
#
if ( -f ligand_fit.pdb )then
 if($count_up == "1" ) then
  cat ligand_fit.pdb  > all_ligands.pdb
 else
  cat ligand_fit.pdb|$grep_type ATOM  >> all_ligands.pdb
 endif
$resolve<<EOD>TEMP.DAT
```

```
hklin $hklin
labin $labin
resolution $dmin $dmax
model ligand_fit.pdb
evaluate_model
EOD
else
echo "" > TEMP.DAT
echo "" > resolve.ligand_scores
endif
#
set correl_line = `$grep_type 'region of model' TEMP.DAT|$grep_type Map`
#if (-f TEMP.DAT ) rm -f TEMP.DAT
if ( $#correl_line > 0 ) then
 set correl = $correl_line[$#correl_line-$#correl_line]
else
 set correl = 0
endif
#
set score_line = `cat resolve.ligand_scores |$grep_type MODIFIED`
set placed_line = `cat resolve.ligand_scores |$grep_type "ATOMS PLACED"`
set poss_line = `cat resolve.ligand_scores |$grep_type TOTAL`
if ( $#score_line > 0 ) then
 set score = $score_line[1-1]
else
 set score = 0
endif
if ( $#placed_line > 0 ) then
 set placed = $placed_line[1-1]:r
else
 set placed = 0
endif
if ( $#poss_line > 0 ) then
 set poss = $poss_line[1-1]:r
else
 set poss = 0
endif
cp ligand_fit.pdb resolve_ligand_${count_up}.pdb
echo "    $count_up        ${placed}/${poss}          $correl         $score     $file"
@ count --
end
#-----------------------------------------------------------------------
#-----------------------------------------------------------------------
```

```
echo ""
echo "Re-evaluating all ligands in $ligand_list_file now over common region"
echo ""
echo " LIGAND   CC     TEMPLATE FILE"
set count = $lines
set count_up = 0
head -10000 all_ligands.pdb > model_2.pdb   # limit to 10000 atoms
while ($count > 0)
@ count_up++
 set file = `tail -${count} $ligand_list_file |head -1 `
 #echo "Evaluating ligand $file"
#
$resolve<<EOD>TEMP.DAT
hklin $hklin
labin $labin
resolution $dmin $dmax
model resolve_ligand_${count_up}.pdb
model_2 model_2.pdb    ! evaluate in region of model_2
evaluate_model
EOD
#
set correl_line = `$grep_type 'region of model' TEMP.DAT|$grep_type Map`
#if (-f TEMP.DAT ) rm -f TEMP.DAT
if ( $#correl_line > 0 ) then
 set correl = $correl_line[$#correl_line-$#correl_line]
else
 set correl = 0
endif
#
echo "    $count_up      $correl  $file"
@ count --
end
#
#
#
```

```
#!/bin/csh
set main_version = 2
set minor_version = 09
set edit_version = 12
#
#                NOTE:  This script requires resolve version 2.08 or higher
echo " "
echo "RESOLVE build script version $main_version.$minor_version.$edit_version "
echo " "
echo "Date: `date`"
echo "Working directory: `pwd`"
echo " "
#
#        RESOLVE iterative model-building script
#                  26-Jun-2005
#
#                 Tom Terwilliger
#           Los Alamos National Laboratory
#
#
#   Input:  FP and SIGFP, experimental phases.
#         Input model  (optional)
#
#   Autobuilding:  Iteratively improve phases using statistical density
#         modification including solvent flattening, histogram-matching,
#         NCS, local pattern recognition, local fragment identification,
#         and model-building and refinement. Optionally include cycles of
#         rebuilding (see below); accept if R decreases.
#
#   Rebuilding: Start with input model and calculate phases using
#         density modification including the model information. Optionally
#         calculate and use prime-and-switch composite omit map.
#         Build new model into the map...iterate
#
#   Model evaluation: as rebuilding, but just evaluate the model using the
#         prime-and-switch composite omit map
#
#   Output: Optimized phases in resolve_best.mtz
#         Partially-refined model in resolve_best.pdb
#         Log file for best cycle in resolve_best.log
#         CCP4-style map file in resolve_best.map
#
#   Major changes:  Version 2.08 allows you to build
#         outside of a region defined by a model.
#         Version 9 allows definition of a mask with a PDB file
#
#=====================================================================
#    EDIT THE NEXT FEW SETS OF LINES TO MATCH YOUR DATA AND SYSTEM
```

```
#
#     PLEASE NOTE: each of the " = " below must have a SPACE on either
#       side (hklin = solve.mtz  NOT hklin=solve.mtz )
#
#=====================================================================
#
#  Anything special for your location (SOLVEDIR etc);
#
setenv SOLVEDIR /usr/local/lib/solve/
setenv TMPDIR .                         # large scratch files go here
#
# location of resolve and resolve libraries;  place for large scratch files
#
set resolve =  resolve
set resolve_pattern =  resolve_pattern
#
#   location of refmac5
#
set refmac5 = refmac5 # NONE if you do not have refmac5
#
#   Input mtz file with at least FP and SIGFP and PHIB
#
set hklin = ../fobs.mtz
set labin = 'FP=FP SIGFP=SIGFP '
set labin_cont = ''
#
#   Optional input model to be rebuilt
#
set pdb_in =  ../2gn5.pdb #  NONE if no model to rebuild
#
#   resolution, solvent_content, sequence, heavy-atom file
#
set dmin =  2.6                   #   high-res limit
set dmax = 200                    #   low-res limit
set solvent_content = 0.40        #   solvent content (required)
set seq_file = ../seq.dat         #   amino acid sequence file (up to 80 char/line)
set ha_file =  NONE               #     e.g., ha.pdb or NONE; used to get NCS
#
set refine_model =  YES       #  say NO if resolution > 3 A usually
set compare_file =  ../coords.pdb #   NONE if no comparison file
#
#    Just build outside the region defined by input model? (Only use if input model has
#    side-chains. Not for a backbone model)
#
set mask_pdb_file = NONE   # PDB file specifying mask for density modification
#
#   criteria for ending..
#
set n_cycle_build_max = 20   # try 20
```

```
set n_cycle_build_min = 3    # try 3
set unacceptable_r = 0.4 # try 0.4 keep going if R > unacceptable_r
set n_cycle_rebuild_max = 50  # try 30 for build, 50 for rebuild
set n_cycle_rebuild_min = 20  # try 20 (Rebuild skipped if R < unacceptable_r)
set n_cycle_rebuild_omit = 10 #  every n'th cycle do omit  if
                              # omit_on_rebuild is YES (starting with nth cycle)
#
#   misc optional parameters...
#
set use_resolve_fragments = AUTO  #  AUTO or YES or NO  (AUTO usually)
set use_resolve_pattern   =  AUTO #  AUTO or YES or NO  (AUTO usually)
#
set superquick_build =     NO #  usually NO. Use YES to get a quick answer
#
set clean_up = YES          # YES usually
set extra_verbose = NO      # only for debugging
set omit_on_rebuild = NO    # NO usually
set quick_as_possible = NO  # NO usually
set n_cycle_image_min = 3   # 3 usually use image info for at least these cycles
set cycle_start = 1         # 1 usually
set evaluate_only =  NO   # NO usually; YES ok; NEVER skip
set use_prob_mask = NO    # NO usually; yes for version 2.0 probability mask
#
#==============================================================================
#               Normally no need to edit below here...
#==============================================================================
#
set skip =  NO
#
set main_version_minimum = 2
set minor_version_minimum = 08
set build_outside_model = NO  # Does not always work. refmac5 fails if resolve writes a residue
                              # with same residue number as a residue in pdb_in
#
setenv SYMOP $SOLVEDIR/symop.lib
setenv SYMINFO $SOLVEDIR/syminfo.lib
setenv CCP4_OPEN UNKNOWN
unlimit
limit coredumpsize 0
#
if ( $extra_verbose == YES ) then
 set ip = 1
else
 set ip = 0
endif
#-----------------------------------------------------------------
#  figure out if this machine uses grep -a or just grep for text files:
echo "A" > test_a.dat
 set test_grep = `grep -a "a" test_a.dat >& tmp.dat`
```

```
if ( $status ) then
#  there was an error...do not use grep -a
 set grep_type = "grep"
else
 set grep_type = "grep -a"
endif
set test_grep = `$grep_type "A" test_a.dat`
if ( $#test_grep != 1 ) then
 echo "Sorry, unable to set the grep command on this system...giving up"
 exit
endif
rm -f test_a.dat
#-------------------------------------------------------------
#  check for all the library files we will need etc
#
if ( $resolve == NONE || $resolve_pattern == NONE ) then
 echo "Sorry, you need to define resolve and resolve_pattern..."
 exit
endif
foreach program ($resolve $resolve_pattern $refmac5)
if ( $ip ) echo "Checking for the program $program ..."
if ( -f $program ) then
 if ( $ip ) echo "OK"
 goto ok
endif
if (  $program == NONE ) then
 if ( $ip ) echo "This program is not used"
 goto ok
endif
which $program >& tmp.dat
set test = `head -1 tmp.dat`
if ( $#test != 1 ) goto bad
if ( -f $test ) then
 if ( $ip ) echo "OK"
 goto ok
endif
bad:
 echo "Sorry, the program $program does not exist?"
 echo "Please check its definition in this script..."
 exit
endif
ok:
end
#
#
foreach lib_file ( symop.lib segments/rho.list \
 segments/fragment_1_0.pdb segments/fragment_2_0.pdb  \
 segments/segment_001_1.pdb segments/segment_001_2.pdb \
 segments/side_chains_best.dat segments/side_chains_rho.dat \
```

```
 segments/gap_library.dat \
 segments/template_30_mean_1.ezd segments/template_30_mask_1.ezd \
 segments/template_30_corr_1.ezd segments/template_30_mean_2.ezd \
 segments/template_30_mask_2.ezd segments/template_30_corr_2.ezd \
 hist.sol_91 hist.sol_92 \
 patterns/hist_cc_0.8.dat patterns/index_95.dat \
 patterns/patterns.info patterns/box001.dat )
#
if ( $ip ) echo "Checking for library file $SOLVEDIR/$lib_file..."
if ( ! -f $SOLVEDIR/$lib_file) then
 echo "Sorry, the library file $lib_file does not exist?"
 echo "Please check the definition of SOLVEDIR in your script: $SOLVEDIR"
 exit
endif
end
#-------------------------------------------------------------
#
if ( $quick_as_possible == YES ) then
 echo "Trying to do everything as quickly as possible..."
 echo "Setting superquick_build, no refinement, no rebuild, no pattern ID,"
 echo "no fragment ID, minimum cycles"
 set refine_model = NO
 set superquick_build = YES
 set use_resolve_pattern = NO
 set use_resolve_fragments = NO
endif
if ( $evaluate_only == YES ) then
 echo "Just evaluating the model $pdb_in"
 set n_cycle_rebuild_max = 0
 set n_cycle_rebuild_min = 0
endif
if ( ( $pdb_in != NONE ) && ( $evaluate_only == YES ) && ( ! -f $pdb_in ) ) then
 echo "Sorry cannot find the model $pdb_in ... required for evaluate_only"
 exit
endif
if ( $build_outside_model == NO || $pdb_in == NONE )then
  # do nothing
else
 echo "Building only around model $pdb_in (excluding region in existing model)"
endif
if ( -f HEADER.pdb ) rm -f HEADER.pdb
if ($pdb_in != NONE )then
$grep_type CRYST1 $pdb_in >>HEADER.pdb
$grep_type ORIGX1 $pdb_in >>HEADER.pdb
$grep_type ORIGX2 $pdb_in >>HEADER.pdb
$grep_type ORIGX3 $pdb_in >>HEADER.pdb
$grep_type SCALE1 $pdb_in >>HEADER.pdb
$grep_type SCALE2 $pdb_in >>HEADER.pdb
$grep_type SCALE3 $pdb_in >>HEADER.pdb
```

```
#  make sure the headers exist...
@ header_lines = `cat HEADER.pdb |wc -l`
if ( $header_lines >= 4 ) goto ok1
 echo "The input PDB file $pdb_in needs to have at least 4 lines of headers with "
 echo "CRYST1 SCALE1 SCALE2 SCALE3"
 echo "Yours seems to have instead $header_lines of headers"
 cat HEADER.pdb
 exit
ok1:
echo "REMARK RESOLVE MODEL" >> HEADER.pdb
endif
#
if ( $n_cycle_build_min < 1 ) then
   echo "Sorry, please set n_cycle_build_min to at least 1"
   exit
endif
if ( $n_cycle_build_max < 1 ) then
   echo "Sorry, please set n_cycle_build_max to at least 1"
   exit
endif
if ( $n_cycle_rebuild_min < 0 ) then
   set n_cycle_rebuild_min = 0
endif
if ( $n_cycle_rebuild_max < 0 ) then
    set n_cycle_rebuild_max = 0
endif
#
#
set past_cycle = $n_cycle_build_min
set n_cycle = $n_cycle_build_max
set n_cycle_rebuild = $n_cycle_rebuild_max
#   (set n_cycle to n_cycle_rebuild_max later)
#
set minimum_coverage = 0
#
#
#
# check for resolve and resolve_pattern
if ( -f resolve.ok ) rm -f resolve.ok
if ( -f resolve.version ) rm -f resolve.version
$resolve <<EOD  >& resolve.log_0
 quit
EOD
if ( ! -f resolve.ok ) then
 echo "Sorry please check the location of resolve..."
 echo "    is it really $resolve?"
 exit
endif
#  make sure this sort of worked...
```

```
set test_ok = `cat resolve.ok`
set test_ok_mem = $test_ok[$#test_ok-$#test_ok]
if ( $test_ok_mem != "ALLOCATED" ) then
 echo "Sorry, resolve was not able to run properly at all"
 echo "The end of the resolve log file says..."
 echo "-------------------------------------------------------"
 tail -12 resolve.log_0
 echo "-------------------------------------------------------"
 exit
endif
if ( ! -f resolve.version ) then
 echo "Sorry, this script requires version $main_version_minimum.$minor_version_minimum or higher of resolve"
 exit
endif
@ version = `cat resolve.version|head -1`
set version_minimum = "$main_version_minimum$minor_version_minimum"
if ( $version < $version_minimum )then
 echo "Sorry, this script requires version $main_version_minimum.$minor_version_minimum or higher of resolve"
 exit
else
 if ( $ip ) echo "Running version $version of resolve"
endif
if ( -f resolve_pattern.ok ) rm -f resolve_pattern.ok
$resolve_pattern <<EOD >& /dev/null
end
EOD
if ( ! -f resolve_pattern.ok ) then
 echo "Sorry please check the location of resolve_pattern... "
 echo "   is it really $resolve_pattern?"
 exit
endif
#
# make sure we have these files...
#
if ( -f $hklin ) then
   echo "Data are in the mtz file $hklin"
else
   echo "Sorry, cannot find the mtz file $hklin"
   exit
endif
echo "LABIN information: $labin $labin_cont"
echo " "
#
if ( $seq_file != NONE &&  -f $seq_file  ) then
  echo "Sequence information from ${seq_file}"
else if ( $seq_file != NONE ) then
  echo "Sorry, cannot find the seq_file file $seq_file"
  exit
endif
```

```
#
if ( ( $ha_file != NONE ) && ( -f $ha_file  ) ) then
  echo "Atom information for NCS from $ha_file"
else if ( $ha_file != NONE ) then
  echo "Sorry, cannot find the ha_file file $ha_file"
  exit
endif
#
if ( ( $pdb_in != NONE ) && ( ! -f $pdb_in ) ) then
 echo "Sorry, cannot find your pdb_in file $pdb_in"
 exit
else if ( ( $cycle_start == 1 ) && ( $pdb_in != NONE ) ) then
 set use_resolve_fragments = NO
 set use_resolve_pattern   =  NO
endif
#
if ( $superquick_build == YES ) then
 set superquick_build = "superquick_build"
else
 set superquick_build = " "
endif
#
@ grep_phib = `echo $labin $labin_cont | $grep_type 'PHIB='|wc -m`
@ grep_fp = `echo $labin $labin_cont | $grep_type 'FP='|wc -m`
#
if ( $grep_fp == 0 ) then
 echo "Sorry, FP is always required in the labin line..."
 exit
endif
if ( $grep_phib == 0 && $pdb_in == NONE ) then
 echo "Sorry if you have no pdb_in file you need phases (PHIB=...)"
 exit
else if ( $pdb_in != NONE ) then
 echo "Using  phase probabilities from $hklin in rebuilding"
endif
if ( $pdb_in == NONE ) then
 set scale = 1.0
else if ( $grep_phib != 0 ) then
 set scale = 1.0
else
 set scale = 0.2
endif
#
echo " "
echo "solvent content ${solvent_content}"
echo "resolution ${dmin} ${dmax}"
if ( $refine_model == NO ) then
 echo "Model will not be refined"
else
```

```
 echo "Model will be refined"
endif
echo " "
if ( $compare_file != NONE ) then
 echo "Coordinates for comparison from $compare_file"
endif
if ( $compare_file != NONE && ! ( -f $compare_file ) ) then
  echo "Sorry, cannot find the comparison PDB file $compare_file"
  exit
endif
#
set blank = " "
#
if ( $pdb_in != NONE ) then
  set combine_resolve_fragments = $blank
  echo "Not using fragment ID"
else if ( $use_resolve_fragments == YES ) then
  set combine_resolve_fragments = "combine_map fragments.map"
  echo "Using fragment ID"
else if ( $use_resolve_fragments == AUTO ) then
  set combine_resolve_fragments = "combine_map fragments.map"
  echo "Testing using fragment ID"
else
  set combine_resolve_fragments = $blank
  echo "Not using fragment ID"
endif
if ( $pdb_in != NONE ) then
  set combine_resolve_pattern = $blank
  echo "Not using resolve_pattern"
else if ( $use_resolve_pattern == YES) then
  set combine_resolve_pattern = "combine_map pattern.map"
  echo "Using resolve_pattern"
else if ( $use_resolve_pattern == AUTO) then
  set combine_resolve_pattern = "combine_map pattern.map"
  echo "Testing using resolve_pattern"
else
  set combine_resolve_pattern = $blank
  echo "Not using resolve_pattern"
endif
#  make n_unacceptable_r  have 5 digits exactly
set n_unacceptable_r = $unacceptable_r:e
if ( $n_unacceptable_r < 1 ) then
 set n_unacceptable_r = 10000
endif
while ( $n_unacceptable_r < 10000 )
 @ n_unacceptable_r = $n_unacceptable_r * 10
end
if ( ( $pdb_in != NONE ) && ( $evaluate_only != YES ) ) then
 echo "If R-factor < 0.$n_unacceptable_r then no rebuilding will be done"
```

```
 echo "If R-factor >= 0.$n_unacceptable_r then maximum cycles will be done"
endif
if ($mask_pdb_file != NONE  && ! -f $mask_pdb_file)then
  echo "Sorry, could not find the mask pdb file $mask_pdb_file"
  exit
else if ($mask_pdb_file != NONE )then
  set mask_line = "model $mask_pdb_file"
  set use_mask_line = "use_model_mask"
  echo "Using $mask_pdb_file to define mask for density modification"
else if ($use_prob_mask == YES )then
  set mask_line = ""
  set use_mask_line = "use_prob"
  echo "Using probability-based mask (version 2.0)"
else
  set mask_line = ""
  set use_mask_line = ""
endif
#
#
#          SETUP:
#
# ==========================================================================
# ==========================================================================
# hklin_image will be our combined phase information
#
set hklin_image = start_image.mtz
set labin_image = 'FP=FP PHIB=PHIM FOM=FOMM HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM'
set labin_image_cont = 'FreeR_flag=FreeR_flag'
#
#  hklin_exp is the file we will use to evaluate phase quality.
set hklin_exp = exp.mtz
set labin_exp = 'FP=FP PHIB=PHIM FOM=FOMM HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM'
set labin_exp_cont = 'FreeR_flag=FreeR_flag'
#
set build_score_best_overall = -1000
set build_score_best_overall_cycle = -1000
set r_best_overall = 99999
#
# ==========================================================================
set cycle = $cycle_start
set cycle_prev = $cycle
@ cycle_prev --
# ==========================================================================
if ( $cycle_start == 1 ) then
 goto cycle_one
endif
echo "Starting at cycle $cycle_start"
#
cycle_one:
```

```
# ==========================================================================
set jump = NO
if ( $jump == YES ) then
set best_cc_value = 0000
set read_mask = " "
echo "JUMPING TO MIDDLE OF FILE AT CYCLE $cycle_start"
endif
if ( $jump == YES ) goto jump_point
# ==========================================================================
#
while ( $cycle <= $n_cycle )
#
set cycle_prev = $cycle
@   cycle_prev --
#
 if ( $ip ) echo " "
if ( $ip ) echo "Cycle $cycle -------------------------------------------"
#
#
if ( -f resolve.log_$cycle ) then
 if ( $ip ) echo "removing previously-existing log file resolve.log_$cycle"
 rm -f resolve.log_$cycle
endif
set build_score_best = -1000
set placed_best = 0
set built_best = 0
if ( -f work_model.pdb ) then
  rm -f work_model.pdb
endif
#
if ( $cycle > 1 ) then
 goto cycle_point
endif
#
# ==========================================================================
#
#  First cycle:
#
#   remove all the files from previous runs..
if ( -f pattern.map ) then
 if ( $ip ) echo "removing existing pattern.map"
 rm -f pattern.map
endif
if ( -f fragments.map ) then
 if ( $ip ) echo "removing existing fragments.map"
 rm -f fragments.map
endif
if ( -f combine.map ) then
 if ( $ip ) echo "removing existing combine.map"
```

```
 rm -f combine.map
endif
if ( -f current_model.pdb ) then
  if ( $ip ) echo "Removing current_model.pdb"
  rm -f current_model.pdb
endif
if ( -f resolve_peaks.dat ) then
  if ( $ip ) echo "removing resolve_peaks.dat"
  rm -f resolve_peaks.dat
endif
if ( -f resolve_peaks.use) then
  if ( $ip ) echo "removing resolve_peaks.use"
  rm -f resolve_peaks.use
endif
if ( -f fragments.dat ) then
  if ( $ip ) echo "removing fragments.dat"
  rm -f fragments.dat
endif
foreach best_file (resolve_best.pdb resolve_best.log resolve_best.mtz \
    refmac_best.log )
if ( -f $best_file ) rm -f $best_file
end
#
#==============================================================================
echo "Copying phase information from $hklin to $hklin_exp"
${resolve}<<EOD >>resolve.log_0
! Copying over $hklin to $hklin_exp
hklin $hklin
labin $labin
labin $labin_cont
hklout $hklin_exp
solvent_content $solvent_content
resolution $dmin $dmax
res_start $dmin
mask_cycles 1
minor_cycles 0
ha_file NONE
no_build
EOD
#
#  make sure this sort of worked...
if ( `cat resolve.ok` != 'OK' ) then
 echo "Sorry, resolve was not able to finish even the first cycle"
 echo "The end of the resolve log file says..."
 echo "-------------------------------------------------------"
 tail -10 resolve.log_0
 echo "-------------------------------------------------------"
 exit
endif
```

```
#
#  for model start we go right on to rebuilding...cycle 2
#
if (  $pdb_in != NONE )  then
 echo " "
 echo "Going right on to rebuilding/model evaluation now..."
 break
endif
#
#  First cycle just do std density modification...2x with
#   2 sets of parameters
#
set best_dm_type = "quick"
#
if ( $quick_as_possible == YES ) then
 set types = "quick"
else
 set types = "quick"
endif
foreach build_type ( $types )
if ( $build_type == "quick" ) then
 set build_1 = "mask_cycles 3"
 set build_2 = "minor_cycles 1"
else
 set build_1 = $blank
 set build_2 = $blank
endif
echo "Testing $build_type density modification..."
#
if ( -f resolve.built ) rm -f resolve.built
$resolve <<EOD >>resolve.log_$cycle
!  Resolve_build script
!  Cycle $cycle...testing $build_type density modification
resolution $dmin $dmax
hklin $hklin
labin $labin
labin $labin_cont
solvent_content $solvent_content
ha_file $ha_file
seq_file $seq_file
$superquick_build
compare_file $compare_file
hklout dummy.mtz
$build_1
$build_2
$mask_line
$use_mask_line
EOD
#
```

```
#  make sure this sort of worked...
if ( `cat resolve.ok` != 'OK' ) then
 echo "Sorry, resolve was not able to finish even the first cycle"
 echo "The end of the resolve log file says..."
 echo "-------------------------------------------------------"
 tail -10 resolve.log_0
 echo "-------------------------------------------------------"
 exit
endif
#
#   score this build:
#
if ( -f resolve.built ) then
    set built = `cat resolve.built|head -1`
    set placed = `cat resolve.built | head -2|tail -1`
    set build_score = `cat resolve.built | head -4|tail -1`
 echo "Score for $build_type : $build_score   built: $built  placed: $placed"
else
 set build_score = 0
 set built = 0
 set placed = 0
endif
#
if ( $build_score > $build_score_best ) then
 set best_dm_type = $build_type
 set build_score_best = $build_score
 set built_best = $built
 set placed_best = $placed
 cp dummy.mtz resolve.mtz_$cycle
 cp resolve.mtz_$cycle omit.mtz
 cp resolve_peaks.dat resolve_peaks.use
 cp resolve.pdb  work_model.pdb
endif
#
end
echo "Best density modification obtained with $best_dm_type approach"
echo $best_dm_type > resolve.type
#
echo " "
echo "Starting building at cycle $cycle"
echo " "
echo "-------------------------------------------------------------"
echo "         RESIDUES  SIDE-CHAINS"
echo " CYCLE     BUILT      PLACED        SCORE      R       Free R   BEST"
#
goto finish
#
#==============================================================================
cycle_point:
```

```
#
# Figure out if we need to try alternate ncs:
if ( -f resolve.ncs_used ) then
 set ncs_used = `cat resolve.ncs_used|head -1`
else
 set ncs_used = NO
endif
if ( ( $ncs_used == NO ) &&  ( -f ncs_test.pdb ) ) then
 set ha_file = ncs_test.pdb
 echo "Trying NCS from $ha_file based on last model built..."
endif
#
#  for first few cycles always use image info if available...
if ( ( $cycle <= $n_cycle_image_min) && ( -f combine.map ) ) then
 goto use_image
endif
#
# density-modify with model information only:
#
if ( -f resolve.built ) rm -f resolve.built
${resolve}<<EOD>>resolve.log_$cycle
! map with model information only cycle $cycle
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
hklout resolve.mtz_$cycle
solvent_content $solvent_content
ha_file $ha_file
seq_file $seq_file
$superquick_build
compare_file $compare_file
composite_all
composite_pdb refine.pdb_
composite_pdb_first 1
composite_pdb_last $cycle_prev
pdb_in refine.pdb_$cycle_prev
image
mask_cycles 1
nohl
$mask_line
$use_mask_line
EOD
#
if ( -f resolve.built ) then
    set built = `cat resolve.built|head -1`
    set placed = `cat resolve.built | head -2|tail -1`
    set build_score = `cat resolve.built | head -4|tail -1`
 if ( $ip ) echo "Build score with model but no image info: $build_score   built: $built  placed: $placed"
else
```

```
 set build_score = 0
 set built = 0
 set placed = 0
 echo "REMARK No model built" > resolve.pdb
 echo "REMARK No model built"
endif
#
set best_method = "no_image"
set build_score_best = $build_score
set built_best = $built
set placed_best = $placed
cp resolve_peaks.dat resolve_peaks.use
cp resolve.pdb  work_model.pdb
#
#  Include model information with image info and build a new model:
#
if ( ! -f combine.map ) then
 goto skip_image
endif
#
use_image:
#
#
 if ( $ip ) echo "Adding in information from combine.map to $hklin_image"
#
${resolve}<<EOD >>resolve.log_$cycle
! applying combine info cycle $cycle
resolution $dmin $dmax
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
solvent_content $solvent_content
pattern_phase
hklout $hklin_image
cc_map_file combine.map
ha_file NONE
no_build
EOD
#
if ( -f resolve.built ) rm -f resolve.built
${resolve}<<EOD>>resolve.log_$cycle
! Get map with model and image information cycle $cycle
hklin $hklin_image
labin $labin_image
labin $labin_image_cont
hklout dummy.mtz
solvent_content $solvent_content
ha_file $ha_file
seq_file $seq_file
```

```
$superquick_build
compare_file $compare_file
composite_all
composite_pdb refine.pdb_
composite_pdb_first 1
composite_pdb_last $cycle_prev
pdb_in refine.pdb_$cycle_prev
image
mask_cycles 1
nohl
$mask_line
$use_mask_line
EOD
#
#   score this build:
#
if ( -f resolve.built ) then
    set built = `cat resolve.built|head -1`
    set placed = `cat resolve.built | head -2|tail -1`
    set build_score = `cat resolve.built | head -4|tail -1`
 if ( $ip ) echo "Score for build using image info  : $build_score   built: $built  placed: $placed"
else
 set build_score = 0
 set built = 0
 set placed = 0
endif
if ( $build_score > $build_score_best ) then
 set best_method = "image"
 set build_score_best = $build_score
 set built_best = $built
 set placed_best = $placed
 cp resolve_peaks.dat resolve_peaks.use
 cp resolve.pdb  work_model.pdb
 cp dummy.mtz resolve.mtz_$cycle
 if ( $ip ) echo "Including image information"
else if ( $use_resolve_fragments == AUTO ) then
 echo "No longer including image information"
  set combine_resolve_pattern = $blank
  set combine_resolve_fragments = $blank
  rm -f combine.map
  goto skip_image
else
 if ( $ip ) echo "Not using image information on this cycle"
endif
#
# Also get the omit map if we are using images...
#
if ( $ip ) echo "Getting composite ps omit map..."
if ( -f omit.mtz ) then
```

```
 rm -f omit.mtz
endif
$resolve <<EOD >>resolve.log_$cycle
!  Cycle $cycle...composite prime-and-switch omit density modification
hklin $hklin_image
labin $labin_image
labin $labin_image_cont
hklout omit.mtz
solvent_content $solvent_content
no_build
ha_file $ha_file
composite_all
composite_pdb refine.pdb_
composite_pdb_first 1
composite_pdb_last $cycle_prev
add_mask
image
omit
mask_cycles 1
nohl
$mask_line
$use_mask_line
EOD
#
skip_image:
#
finish:
#
#  Start of refine-extend cycles
#
foreach extend_cycle ( 0 1 2)
#
if ( $ip ) echo "Extend cycle $extend_cycle"
if( $extend_cycle > 0 ) then
 if ( $ip ) echo "Extending model"
#
if ( -f resolve.built ) rm -f resolve.built
${resolve}<<EOD >> resolve.log_$cycle
hklin resolve.mtz_$cycle
labin FP=FP PHIB=PHIM FOM=FOMM
hklout dummy.mtz
solvent_content $solvent_content
resolution ${dmin} ${dmax}
extend_only
ha_file $ha_file
seq_file $seq_file
compare_file $compare_file
pdb_in resolve.pdb
no_trim
```

```
nohl
EOD
endif
if ( ( $extend_cycle > 0 ) && ( -f resolve.built ) ) then
    set built = `cat resolve.built|head -1`
    set placed = `cat resolve.built | head -2|tail -1`
    set build_score = `cat resolve.built | head -4|tail -1`
 if ( $ip ) echo "Score for extension: $build_score   built: $built  placed: $placed"
else if ( $extend_cycle > 0 ) then
 set build_score = 0
 set built = 0
 set placed = 0
endif
if ( ( $extend_cycle > 0 ) && ( $build_score > $build_score_best ) ) then
 set build_score_best = $build_score
 set built_best = $built
 set placed_best = $placed
 cp resolve.pdb work_model.pdb
else if ( $extend_cycle > 0 ) then
 if ( $ip ) echo "Extension did not improve score...going on"
#  break to end of extend cycles..
 break
endif
#
#
#  decide if we refine model this cycle..
#
if ( -f resolve.coverage ) then
 set coverage1 = `cat resolve.coverage|head -3|tail -1`
 set coverage = 0
 if ( $#coverage1> 0 ) set coverage= $coverage1[$#coverage1-$#coverage1]:e
else
 set coverage = 0
endif
if ( $refine_model == NO ) then
 set refine_it = NO
else if ( $build_score_best < 2 ) then
 set refine_it = NO
else if (  $coverage < $minimum_coverage )then
 set refine_it = NO
else if ( ! -f work_model.pdb ) then
 set refine_it = NO
 echo "REMARK No model built" > work_model.pdb
 echo "REMARK No model built extend cycle $extend_cycle"
else
 set refine_it = YES
endif
if ( $refine_it == YES ) then
 set ncyc = 20
```

```
else
 set ncyc = 0
endif
if ( ( $refmac5 != NONE ) && ( $build_score_best > 1 ) ) then
if ( $ip ) echo "Refining model $ncyc cycles and placing it in refine.pdb_$cycle"
#
cat work_model.pdb | $grep_type -v HETATM > resolve.pdb
#
# --------refine with refine.--------------
#
#  NOTE: refine against "best" map = resolve.mtz_$cycle
#
$refmac5 xyzin resolve.pdb hklin resolve.mtz_$cycle xyzout refine.pdb  \
PROTOUT $CCP4_SCR/protout.dat \
PROTCOUNTS $CCP4_SCR/counts.dat \
PROTSCR $CCP4_SCR/counts.scr \
<<EOD > refmac.log_${cycle}
MAKE HYDRogens No
MAKE CHECK 0
LABI FP=FP SIGFP=SIGFP HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM FREE=FreeR_flag
REFI TYPE RESTrained RESOlution ${dmax} ${dmin}
#   Maximum likelihood refinement
REFI RESI MLKF
#   Overall B value refinement
REFI BREF OVER      ! Refine overall B-values
#   Tight restraints  Lowered further 051403
WEIG MATR 0.05
#   Scale down shifts at every cycle by factor 0.5
DAMP 0.5 0.5
#   No Blur (scale down reliability of) phases
REFI PHASed  BBLUrring 00.0  SCBLurring 1.
#  Babinet's bulk solvent scale parameters. But fix B value
SCALe TYPE BULK
SCALe LSSCale FIXBulk BBULk 200.0
#   Anisotropic scaling. It is default
SCALe LSSCale ANISotropic
#   Fix Babinet's bulkd solvent for sigmaA
SCAL MLSC FIXBulk BBULk 200.0 SCBUlk -0.05
NCYC  $ncyc
MONI MEDI
BINS 10
EOD
#-------------------------------------------------------
#
cp refine.pdb refine.pdb_$cycle
else if (  $build_score_best > 1  ) then
#   no refmac5...just copy over
 cat work_model.pdb | $grep_type -v HETATM > resolve.pdb
 cp resolve.pdb refine.pdb
```

```
 cp refine.pdb refine.pdb_$cycle
else
if ( $ip ) echo "No model built this try"
 echo "REMARK  No model built cycle $cycle " > refine.pdb_$cycle
 echo "REMARK  No model built cycle $cycle "
endif
#
end_of_extend:
end
#
#
#  Get R and freeR value on refine.pdb_$cycle:
#
set freer_line = `$grep_type 'FREE R VALUE                    :' refine.pdb_$cycle|tail -1`
set r_line = `$grep_type 'R VALUE            (WORKING SET) :' refine.pdb_$cycle|tail -1`
if ( $#freer_line > 0 ) then
 set freer = $freer_line[$#freer_line-$#freer_line]:e
else
 set freer = 99999
endif
if ( $#r_line > 0 ) then
 set r = $r_line[$#r_line-$#r_line]:e
else
 set r = 99999
endif
if ( $ncyc == 0 )then
 set freer = $r
endif
#
if ( $ip ) echo " R/freeR for this model: 0.$r/0.$freer"
#
if ( $cycle < 10 ) then
 set b1 = " "
else
 set b1 = ""
endif
if ( $built_best < 100 ) then
 set b2 = " "
else
 set b2 = ""
endif
if ( $placed_best >= 100 ) then
 set b3 = ""
else if ( $placed_best >= 10 )then
 set b3 = " "
else
 set b3 = "  "
endif
if ( $build_score_best < 100 )then
```

```
 set b4 = " "
else
 set b4 = ""
endif
# Decide how to score. If R < unacceptable_r or r_best_overall < unacceptable_r
#   then use R. Otherwise use score
#
if ( ( $r < $n_unacceptable_r ) && ( $r < $r_best_overall ) ) then
 set this_is_best = YES
else if ( ( $r_best_overall >= $n_unacceptable_r ) && \
   ( $build_score_best > $build_score_best_overall ) ) then
 set this_is_best = YES
else
 set this_is_best = NO
endif
if ( $this_is_best == YES ) then
 if ( $ip ) echo "This is the best model overall so far. Previous best had score of $build_score_best_overall"
 if ( $ip ) echo "Copied refine.pdb_$cycle to resolve_best.pdb"
 if ( $ip ) echo "Copied resolve.log_$cycle resolve_best.log"
 if ( $ip ) echo "Copied resolve.mtz_$cycle to resolve_best.mtz"
 @ build_score_best_overall = $build_score_best
 @ build_score_best_overall_cycle = $cycle
 @ r_best_overall = $r
 cp refine.pdb_$cycle resolve_best.pdb
 cp resolve.log_$cycle resolve_best.log
 cp resolve.mtz_$cycle resolve_best.mtz
 if( -f refmac.log_$cycle ) cp refmac.log_$cycle refmac_best.log
 echo "  ${b1}$cycle      ${b2}$built_best        ${b3}$placed_best         ${b4}$build_score_best     0.$r    0.$freer    *"
else
 echo "  ${b1}$cycle      ${b2}$built_best        ${b3}$placed_best         ${b4}$build_score_best     0.$r    0.$freer    "
endif
#
#
#=====================================================================
#=====================================================================
#
#
set have_map = NO
if ( ( "$combine_resolve_fragments" != $blank ) && ( -f resolve_peaks.dat )  )  then
if ( $ip ) echo "Carrying out fragment ID based on resolve.mtz_$cycle"
cp resolve_peaks.use resolve_peaks.dat
#
$resolve <<EOD >> resolve.log_$cycle
hklin resolve.mtz_$cycle
labin FP=FP PHIB=PHIM FOM=FOMM
solvent_content $solvent_content
nohl
build_image
noget_peaks      ! Don't need peaks because we just got them!
```

```
mask_cycles 0
minor_cycles 0
superquick_build
ha_file NONE
$mask_line
$use_mask_line
EOD
cp dump.map fragments.map
set have_map = YES
endif
#
if ( "$combine_resolve_pattern" != $blank ) then
if ( $ip ) echo "Carrying out pattern ID based on omit.mtz"
$resolve_pattern<<EOD >> resolve.log_$cycle
hklin omit.mtz
labin FP=FP PHIB=PHIM FOM=FOMM
recover_image
cc_map_file dump.map
EOD
cp dump.map pattern.map
set have_map = YES
endif
#
#  Decide which to combine..
#
if ( $have_map == YES ) then
if ( $ip ) echo "Combining all image information into combine.map"
#
$resolve<<EOD >> resolve.log_$cycle
hklin omit.mtz
labin FP=FP PHIB=PHIM FOM=FOMM
$combine_resolve_pattern
$combine_resolve_fragments
ha_file NONE
EOD
#
else
 if ( $ip ) echo "No image information generated this cycle"
 if ( -f combine.map ) rm -f combine.map
endif
#
#
#  Decide if we should keep going or quit...
#   quit if
#   r_best_overall < n_unacceptable_r and
#   it has been at least past_cycle cycles since things got better
#
@ cutoff_cycle = $cycle - $past_cycle
if ( ( $build_score_best_overall_cycle <= $cutoff_cycle ) && ( $r_best_overall < $n_unacceptable_r ) ) then
```

```
 echo "Best model was built on cycle $build_score_best_overall_cycle which was"
 echo "   more than $past_cycle cycles ago..."
 @ cycle ++
 break
endif
#
end_of_cycle:
#
#  clean up...
#
if ( $clean_up == YES  && -f resolve.log_$cycle ) then
 if ( -f resolve.log_$cycle ) rm -f resolve.log_$cycle
 if ( -f refmac.log$cycle ) rm -f refmac.log_$cycle
 if ( -f resolve.mtz_$cycle ) rm -f resolve.mtz_$cycle
endif
#==============================================================================
#
if ( $ip ) echo "END OF CYCLE $cycle"
@ cycle ++
end
#
if ( $pdb_in != NONE ) then
 echo "Copying $pdb_in to resolve_group.pdb and resolve_best.pdb for rebuilding/evaluation..."
 cp $pdb_in resolve_best.pdb
 cp $pdb_in resolve_group.pdb
 goto start_rebuild
endif
#
echo "" > resolve_group.pdb
@ cyc = $cycle - 5
if ( $cyc < 1 ) set cyc = 1
while ( $cyc < $cycle )
 echo "Adding refine.pdb_$cyc to resolve_group.pdb"
 cat refine.pdb_$cyc >>resolve_group.pdb
 @ cyc ++
end
echo "Starting PDB group model for rebuild is in resolve_group.pdb"
#
if ( -f resolve_best.pdb ) then
 echo "Best model was built on cycle $build_score_best_overall_cycle"
 echo "This model is now in resolve_best.pdb"
else
 echo "No model was built...quitting"
 exit
endif
#
#=================================================================
if (  $quick_as_possible == YES ) then
 goto wrap_up
```

```
endif
if (  $r_best_overall < $n_unacceptable_r ) then
 goto wrap_up
endif
if ( $n_cycle_rebuild_max == 0 ) then
 goto wrap_up
endif
 echo " "
 echo "Our best R-factor (0.$r_best_overall) is > cutoff of $unacceptable_r"
 echo "so we are going to try and rebuild the model..."
 echo " "
#=================================================================
#
start_rebuild:
#
set past_cycle = $n_cycle_rebuild_min
@ cycle_start = $cycle + 1
@ n_cycle = $cycle_start + $n_cycle_rebuild - 1
#
#
#
set pdb_start = resolve_group.pdb          #   starting PDB file
#
if ( $pdb_in == NONE ) then
 set scale = 1.0
 set extend = NO
 set reuse_chain = NO
else if ( $grep_phib != 0 ) then
#   we have phases, so use scale = 1.0...but we are rebuilding model, so
#    use extend and reuse_chain to get maximum building
 set scale = 1.0
 set extend = YES
 set reuse_chain = YES
else
#  slightly different rules for rebuilding from scratch from a model
#   without any phase information:
 set scale = 0.2
 set extend = YES
 set reuse_chain = YES
endif
#
set r_max = 0.$r_best_overall # maximum free R to write anything to resolve_best.pdb
#
if ( $ip ) echo "Output density-modified phases will be in image_only_dm_image.mtz_[cycle]"
#
#
if ( $evaluate_only != YES && $extend == NO )then
 echo "Model will not be extended each cycle"
 set extend_flag = ' 0 '
```

```
else if ( $evaluate_only != YES ) then
 echo "Model will be extended each cycle"
 set extend_flag = ' 0 1 2 '
endif
#
if ( $evaluate_only != YES && $omit_on_rebuild == YES )then
 echo "Omit prime-and-switch phasing will be used every  $n_cycle_rebuild_omit cycle"
 set omit = 'omit'
else
 echo "Standard image-based phasing will be used"
 set omit = ' '
endif
if ( $ip ) echo "Scale on map weighting is $scale"
set n_r_max = $r_max:e
if ( $cycle_start > 2 ) then
 echo "Maximum R to save model: 0.$n_r_max "
endif
#
echo " "
#
@ cycle = $cycle_start - 1
@ cycle_pdb_start = $cycle
#
echo "Copying ${pdb_start} to refine.pdb_$cycle"
cp ${pdb_start} refine.pdb_$cycle
#
#=========================================================================
@ cycle ++
if ( $evaluate_only != YES ) then
echo "Starting rebuilding at cycle $cycle of total of $n_cycle cycles"
#==============================================================
#==============================================================
#
echo " "
echo "-------------------------------------------------------------"
echo "         RESIDUES  SIDE-CHAINS"
echo " CYCLE     BUILT      PLACED        SCORE      R       Free R   BEST"
endif
#
set n_omit_cycle = 0
while ( $cycle <= $n_cycle )
@ n_omit_cycle ++
if ( $n_cycle_rebuild_omit == 0 ) then
#  is_omit_cycle is NO for this one; defaults for omit_use and pdb_in
#
 set is_omit_cycle = NO
 set omit_use = $omit
 set pdb_in_current_model = 'pdb_in current_model.pdb'
else if ( $n_omit_cycle == 1 || $n_omit_cycle == $n_cycle_rebuild_omit ) then
```

```
 set n_omit_cycle = 1
 set is_omit_cycle = YES
 if ( $ip ) echo "Using $omit model phasing on this cycle and building from scratch"

 set omit_use = $omit
 set pdb_in_current_model = ' '
else
 set omit_use = ' '
 set pdb_in_current_model = 'pdb_in current_model.pdb'
 set is_omit_cycle = NO
endif
#
set cycle_prev = $cycle
@ cycle_prev --
#
if ( -f resolve.log_$cycle ) then
 if ( $ip ) echo "removing previously-existing log file resolve.log_$cycle"
 rm -f resolve.log_$cycle
endif
set build_score_best = -1000
set built_best = 0
set placed_best = 0
#
if ( -f work_model.pdb ) then
  if ( $ip ) echo "removing work_model.pdb"
  rm -f work_model.pdb
endif
#
if ( $build_outside_model == YES ) then
# put all current models in all.pdb
if ( -f all.pdb ) rm -f all.pdb
set c_cycle = $cycle_pdb_start
set c_end = $cycle_prev
while ($c_cycle <= $c_end)
if ( -f refine.pdb_${c_cycle} ) cat refine.pdb_${c_cycle} >> all.pdb
@ c_cycle ++
end
set add_mask = 'add_mask'
set model_all = 'model all.pdb'
set build_outside = 'build_outside_model'
else
set add_mask = ''
set model_all = ''
set build_outside = ''
endif
#
#  get image of pdb_start and all models up to now into image.mtz.
#  Reflns from hklin_exp
#
```

```
if ( $ip ) echo "Getting map of ${pdb_start} and all models built into binary file dump.map"
#
${resolve}<<EOD >> resolve.log_$cycle
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
resolution ${dmin} ${dmax}
composite_pdb refine.pdb_
composite_pdb_start $cycle_pdb_start
composite_pdb_end $cycle_prev
composite_all
dump_composite
EOD
#
if ( $ip ) echo "Getting image of ${pdb_start} and built models using dump.map into image_only.mtz"
#
${resolve}<<EOD >> resolve.log_$cycle
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
resolution ${dmin} ${dmax}
no_build
pattern_phase
prior_weight 1.0
hklout image_only.mtz
read_cc_map
cc_map_file dump.map
nohl
EOD
#  make sure this sort of worked...
if ( `cat resolve.ok` != 'OK' ) then
 echo "Sorry, resolve was not able to finish on the first cycle"
 echo "The end of the resolve log file says..."
 echo "-------------------------------------------------------"
 tail -10 resolve.log_$cycle
 echo "-------------------------------------------------------"
 exit
endif
#
if ( $ip ) echo "Density-modifying image of $pdb_start and prev models to get image_only_dm.mtz"
#
${resolve}<<EOD >> resolve.log_$cycle
hklin image_only.mtz
labin FP=FP PHIB=PHIM FOM=FOMM  HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM
hklout image_only_dm.mtz
solvent_content $solvent_content
prior_weight 1.0
database 5
resolution ${dmin} ${dmax}
```

```
$add_mask      ! optionally add mask based on all.pdb (model_all)
$model_all
mask_cycles 1
no_build
nohl
EOD
#
if ( $ip ) echo "Carrying out $omit_use phasing using image_only_dm.mtz "
if ( $ip ) echo " and ${pdb_start} and building model"
#
#
#   either build using this map, or build around starting model only!
#
if ( -f resolve.built ) rm -f resolve.built
if ( -f built_model.pdb ) rm -f built_model.pdb
if ( -f resolve.pdb ) rm -f resolve.pdb
#
if ( $build_outside_model == NO || $pdb_in == NONE )then
 if ($ip) echo "Standard build"
#
${resolve}<<EOD >> resolve.log_${cycle}
!standard build
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
hklstart image_only_dm.mtz
labstart FP=FP PHIB=PHIM FOM=FOMM
hklout image_only_dm_image.mtz_${cycle}
solvent_content $solvent_content
database 5
resolution ${dmin} ${dmax}
composite_pdb refine.pdb_
composite_pdb_start $cycle_pdb_start
composite_pdb_end $cycle_prev
composite_all
prior_weight 1.0
scale_refl $scale
image
$omit_use
mask_cycles 0
$superquick_build
ha_file $ha_file
seq_file $seq_file
compare_file $compare_file
$pdb_in_current_model
nohl
$mask_line
$use_mask_line
EOD
```

```
if ( -f resolve.pdb ) cp resolve.pdb work_model.pdb
if ( -f resolve.pdb ) cp resolve.pdb built_model.pdb
cp HEADER.pdb work_model_start.pdb
#
else
#
 if ($ip) echo "Building around model $pdb_in in 2 steps"
cp HEADER.pdb work_model.pdb
cp HEADER.pdb built_model.pdb
cat $pdb_in |$grep_type ATOM >>work_model.pdb
cat $pdb_in |$grep_type HETATM >>work_model.pdb
cp work_model.pdb work_model_start.pdb
#
${resolve}<<EOD >> resolve.log_${cycle}
! building around model $pdb_in in 2 steps
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
hklstart image_only_dm.mtz
labstart FP=FP PHIB=PHIM FOM=FOMM
hklout image_only_dm_image.mtz_${cycle}
solvent_content $solvent_content
database 5
resolution ${dmin} ${dmax}
composite_pdb refine.pdb_
composite_pdb_start $cycle_pdb_start
composite_pdb_end $cycle_prev
composite_all
prior_weight 1.0
scale_refl $scale
image
$omit_use
mask_cycles 0
no_build
ha_file $ha_file
seq_file $seq_file
compare_file $compare_file
$pdb_in_current_model
$add_mask   ! add mask from model to rmsd mask
$model_all
$mask_line
$use_mask_line
nohl
EOD
#
${resolve}<<EOD >> resolve.log_${cycle}
! Building outside input model $pdb_in only
hklin  image_only_dm_image.mtz_${cycle}
labin FP=FP PHIB=PHIM FOM=FOMM
```

```
ha_file NONE
add_mask
model work_model_start.pdb
database 5
solvent_content $solvent_content
resolution ${dmin} ${dmax}
nohl
build_only
build_outside_model
ha_file $ha_file
seq_file $seq_file
compare_file $compare_file
$superquick_build
EOD
 if ( -f resolve.built)    $grep_type ATOM resolve.pdb  >>work_model.pdb
 if ( -f resolve.built)    $grep_type ATOM resolve.pdb  >>built_model.pdb
endif
#
if ( $omit_use == omit )then
 cp resolve_composite_map.mtz  build.mtz
 if ( $ip ) echo "Copied resolve_composite_map.mtz to build.mtz"
else
 cp image_only_dm_image.mtz_${cycle} build.mtz
 if ( $ip ) echo "Copied image_only_dm_image.mtz to build.mtz"
endif
 cp image_only_dm_image.mtz_${cycle} refine.mtz
 if ( $ip ) echo "Copied image_only_dm_image.mtz to refine.mtz"
#
#
if (  -f resolve.built ) then
    set built = `cat resolve.built|head -1`
    set placed = `cat resolve.built | head -2|tail -1`
    set build_score = `cat resolve.built | head -4|tail -1`
    cp resolve.pdb built_model.pdb
else
 set build_score = 0
 set built = 0
 set placed = 0
 echo "REMARK No model built" > work_model.pdb
 if ( -f built_model.pdb ) rm -f built_model.pdb
 echo "REMARK No model built"
endif
set build_score_best = $build_score
set built_best = $built
set placed_best = $placed
 if ( $ip ) echo "Score : $build_score   built: $built  placed: $placed"
#
#==================================================================
#
```

```
#  Start of refine-extend cycles
#
if  ( $build_outside_model == YES && $pdb_in != NONE )then
set add_mask_use = 'add_mask'
set model_in = 'model work_model_start.pdb'
set build_outside_model_use = 'build_outside_model'
else
set add_mask_use = ''
set model_in = ''
set build_outside_model_use = ''
endif
foreach extend_cycle ( $extend_flag )
#
if ( $ip ) echo "Extend cycle $extend_cycle"
if( -f built_model.pdb && $extend_cycle > 0 ) then
 if ( $ip ) echo "Extending model"
#
if ( -f resolve.built ) rm -f resolve.built
${resolve}<<EOD >> resolve.log_$cycle
hklin build.mtz
labin FP=FP PHIB=PHIM FOM=FOMM
hklout dummy.mtz
solvent_content $solvent_content
resolution ${dmin} ${dmax}
extend_only
$add_mask_use
$model_in     ! building outside work_model_start only
$build_outside_model_use    ! and map to the region of this model too
ha_file $ha_file
seq_file $seq_file
compare_file $compare_file
pdb_in built_model.pdb    ! extend what we just built (only)
no_trim
nohl
$mask_line
$use_mask_line
EOD
endif
if ( ( $extend_cycle > 0 ) && ( -f resolve.built ) ) then
    set built = `cat resolve.built|head -1`
    set placed = `cat resolve.built | head -2|tail -1`
    set build_score = `cat resolve.built | head -4|tail -1`
 if ( $ip ) echo "Score for extension: $build_score   built: $built  placed: $placed"
else if ( $extend_cycle > 0 ) then
 set build_score = 0
 set built = 0
 set placed = 0
endif
if ( ( $extend_cycle > 0 ) && ( $build_score > $build_score_best ) ) then
```

```
 set build_score_best = $build_score
 set built_best = $built
 set placed_best = $placed
 cp resolve.pdb built_model.pdb
 cp work_model_start.pdb work_model.pdb
 $grep_type ATOM built_model.pdb >>work_model.pdb
 if ( $ip ) echo "Adding extended model to starting work model"
else if ( $extend_cycle > 0 ) then
 if ( $ip ) echo "Extension did not improve score...going on"
#  break to end of the extend cycles...
 break
endif
#
#
#  decide if we refine model this cycle..
#
if ( -f resolve.coverage ) then
 set coverage1 = `cat resolve.coverage|head -3|tail -1`
 set coverage = 0
 if ( $#coverage1> 0 ) set coverage= $coverage1[$#coverage1-$#coverage1]:e
else
 set coverage = 0
endif
if ( $refine_model == NO ) then
 set refine_it = NO
else if ( $build_score_best < 2 ) then
 set refine_it = NO
else if ( ( $coverage < $minimum_coverage ) && ( $build_outside_model == NO) )then
 set refine_it = NO
else if ( ! -f work_model.pdb ) then
 echo "REMARK No model built" > work_model.pdb
 echo "REMARK No model built extend cycle $extend_cycle"
 set refine_it = NO
else
 set refine_it = YES
endif
if ( $refine_it == YES ) then
 set ncyc = 20
else
 set ncyc = 0
endif
if ( ( $refmac5 != NONE ) && ( $build_score_best > 1 ) ) then
if ( $ip ) echo "Refining model $ncyc cycles and placing it in refine.pdb_$cycle"
#
cat work_model.pdb | $grep_type -v HETATM > resolve.pdb
#
# --------refine with refine.--------------
#
#  NOTE: refine against "best" map = refine.mtz
```

```
#
cp resolve.pdb refine.pdb   # in case it doesn't work...
#
$refmac5 xyzin resolve.pdb hklin refine.mtz xyzout refine.pdb  \
PROTOUT $CCP4_SCR/protout.dat \
PROTCOUNTS $CCP4_SCR/counts.dat \
PROTSCR $CCP4_SCR/counts.scr \
<<EOD > refmac.log_${cycle}
MAKE HYDRogens No
MAKE CHECK 0
LABI FP=FP SIGFP=SIGFP HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM FREE=FreeR_flag
REFI TYPE RESTrained RESOlution ${dmax} ${dmin}
#   Maximum likelihood refinement
REFI RESI MLKF
#   Overall B value refinement
REFI BREF OVER      ! Refine overall B-values
#   Tight restraints  Lowered further 051403
WEIG MATR 0.05
#   Scale down shifts at every cycle by factor 0.5
DAMP 0.5 0.5
#   No Blur (scale down reliability of) phases
REFI PHASed  BBLUrring 00.0  SCBLurring 1.
#  Babinet's bulk solvent scale parameters. But fix B value
SCALe TYPE BULK
SCALe LSSCale FIXBulk BBULk 200.0
#   Anisotropic scaling. It is default
SCALe LSSCale ANISotropic
#   Fix Babinet's bulkd solvent for sigmaA
SCAL MLSC FIXBulk BBULk 200.0 SCBUlk -0.05
NCYC $ncyc
MONI MEDI
BINS 10
EOD
#-------------------------------------------------------
#
cp refine.pdb refine.pdb_$cycle
else if (  $build_score_best > 1  ) then
#   no refmac5...just copy over
 cat work_model.pdb | $grep_type -v HETATM > resolve.pdb
 cp resolve.pdb refine.pdb
 cp refine.pdb refine.pdb_$cycle
else
 if ( $ip ) echo "No model built this try"
 echo "REMARK  No model built cycle $cycle " > refine.pdb_$cycle
 echo "REMARK  No model built cycle $cycle "
endif
#
end_of_extend:
end
```

```
#
#  Get R and freeR value on refine.pdb_$cycle:
set freer_line = `$grep_type 'FREE R VALUE                    :' refine.pdb_$cycle|tail -1`
set r_line = `$grep_type 'R VALUE           (WORKING SET) :' refine.pdb_$cycle|tail -1`
#
if ( $#freer_line > 0 ) then
 set freer = $freer_line[$#freer_line-$#freer_line]:e
else
 set freer = 99999
endif
if ( $#r_line > 0 ) then
 set r = $r_line[$#r_line-$#r_line]:e
else
 set r = 99999
endif
if ( $ncyc == 0 )then
 set freer = $r
endif
#
if ( $ip ) echo " R/freeR for this model: 0.$r/0.$freer"
#
if ( $cycle < 10 ) then
 set b1 = " "
else
 set b1 = ""
endif
if ( $built_best < 100 ) then
 set b2 = " "
else
 set b2 = ""
endif
if ( $placed_best >= 100 ) then
 set b3 = ""
else if ( $placed_best >= 10 )then
 set b3 = " "
else
 set b3 = "  "
endif
if ( $build_score_best < 100 )then
 set b4 = " "
else
 set b4 = ""
endif
# Decide how to score. If R < unacceptable_r or r_best_overall < unacceptable_r
#   then use R. Otherwise use score
#
if ( ( $r < $n_unacceptable_r ) && ( $r < $r_best_overall ) ) then
 set this_is_best = YES
else if ( ( $r_best_overall >= $n_unacceptable_r ) && \
```

```
   ( $r <= $n_r_max ) && \
   ( $build_score_best > $build_score_best_overall ) ) then
 set this_is_best = YES
else
 set this_is_best = NO
endif
if ( $this_is_best == YES ) then
 if ( $ip ) echo "This is the best model overall so far. Previous best had score of $build_score_best_overall"
 if ( $ip ) echo "Copied refine.pdb_$cycle to resolve_best.pdb"
 if ( $ip ) echo "Copied resolve.log_$cycle resolve_best.log"
 if ( $ip ) echo "Copied refine.mtz to resolve_best.mtz"
 @ build_score_best_overall = $build_score_best
 @ build_score_best_overall_cycle = $cycle
 @ r_best_overall = $r
 cp refine.pdb_$cycle resolve_best.pdb
 cp resolve.log_$cycle resolve_best.log
 cp refine.mtz resolve_best.mtz
 if( -f refmac.log_$cycle ) cp refmac.log_$cycle refmac_best.log
 echo "  ${b1}$cycle      ${b2}$built_best        ${b3}$placed_best         ${b4}$build_score_best     0.$r    0.$freer    *"
else
 echo "  ${b1}$cycle      ${b2}$built_best        ${b3}$placed_best         ${b4}$build_score_best     0.$r    0.$freer    "
endif
#
#=================================================================
#
if ( $reuse_chain == YES )then
 cat  refine.pdb > current_model.pdb
 if ( $ip ) echo " Using refine.pdb fragments as part of next build cycle"
else if ( -f current_model.pdb ) then
  rm -f current_model.pdb
endif
#
#
if ( $ip ) echo " "
if ( $ip ) echo "End of cycle $cycle with score of $build_score_best"
if ( $ip ) echo " "
#
#  Decide if we should keep going or quit..
@ cutoff_cycle = $cycle - $past_cycle
if ( ( $build_score_best_overall_cycle <= $cutoff_cycle ) && ( $r_best_overall < $n_unacceptable_r ) ) then
 echo "Our best model was built on cycle $build_score_best_overall_cycle which was"
 echo "   more than $past_cycle cycles ago...calling it quits with this model,"
 echo "   now in resolve_best.pdb"
 break
endif
#
#  clean up...
#
if ( $clean_up == YES  ) then
```

```
 if ( -f resolve.log_$cycle ) rm -f resolve.log_$cycle
 if ( -f refmac.log_$cycle ) rm -f refmac.log_$cycle
 if ( -f resolve.mtz_$cycle ) rm -f resolve.mtz_$cycle
 if ( -f refine.mtz ) rm -f refine.mtz
 if ( -f image_only_dm_image.mtz_$cycle ) rm -f image_only_dm_image.mtz_$cycle
endif
#
if ( is_omit_cycle == YES ) then
#  set this cycle as first one to consider when creating composite from PDB files
 set cycle_pdb_start = $cycle
endif
#
@ cycle ++
#
#   end of build cycle
#
end
#
wrap_up:
if ($evaluate_only == NEVER)then
  exit
endif
#
#==========================================================================
#==========================================================================
echo " "
echo "Analyzing model based on prime-and-switch composite omit map"
#
echo "First rebuilding current model to remove refinement bias..."
if ( ! -f resolve_best.pdb ) then
 echo "Sorry cannot find resolve_best.pdb?"
 exit
else
 cp resolve_best.pdb test.pdb_0
endif
if ( $skip == YES ) then
 goto evaluate_it
endif
#
${resolve}<<EOD > resolve_eval.log_1
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
resolution ${dmin} ${dmax}
composite_pdb test.pdb_
composite_pdb_start 0
composite_pdb_end 0
composite_all
dump_composite
```

```
EOD
#
${resolve}<<EOD >> resolve_eval.log_1
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
resolution ${dmin} ${dmax}
no_build
pattern_phase
prior_weight 1.0
hklout image_only.mtz
read_cc_map
cc_map_file dump.map
nohl
EOD
#
if ( $ip ) echo "Density-modifying image of resolve_best.pdb to get image_only_dm.mtz"
#
${resolve}<<EOD >> resolve_eval.log_1
hklin image_only.mtz
labin FP=FP PHIB=PHIM FOM=FOMM  HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM
hklout image_only_dm.mtz
solvent_content $solvent_content
prior_weight 1.0
database 5
resolution ${dmin} ${dmax}
no_build
nohl
EOD
#
if ( $ip ) echo "Carrying out phasing using image_only_dm.mtz "
if ( $ip ) echo " and resolve_best.pdb and rebuilding model"
#
if ( -f resolve.built ) rm -f resolve.built
${resolve}<<EOD >> resolve_eval.log_1
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
hklstart image_only_dm.mtz
labstart FP=FP PHIB=PHIM FOM=FOMM
hklout image_only_dm_image.mtz
solvent_content $solvent_content
database 5
resolution ${dmin} ${dmax}
composite_pdb test.pdb_
composite_pdb_start 0
composite_pdb_end 0
composite_all
prior_weight 1.0
```

```
scale_refl $scale
image
mask_cycles 0
$superquick_build
seq_file $seq_file
compare_file $compare_file
ha_file $ha_file
hklout test.mtz_1
nohl
$mask_line
$use_mask_line
EOD
if ( -f resolve.pdb ) then
 cp resolve.pdb test.pdb_1
else
 echo "Sorry, no model obtained on rebuilding...quitting"
 exit
endif
if (  -f resolve.built  ) then
    set built = `cat resolve.built|head -1`
    set placed = `cat resolve.built | head -2|tail -1`
    set build_score = `cat resolve.built | head -4|tail -1`
else
 set build_score = 0
 set built = 0
 set placed = 0
 echo "Sorry, no model obtained on rebuilding...quitting"
 exit
endif
#
#  report what happened in our rebuilding...
echo "Rebuilt model...$built residues built, $placed with side-chains in test.pdb_1"
#
echo "Generating composite prime-and-switch composite omit map based on rebuilt model"
#
${resolve}<<EOD >> resolve_eval.log_1
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
resolution ${dmin} ${dmax}
composite_pdb test.pdb_
composite_pdb_start 1
composite_pdb_end 1
composite_all
dump_composite
EOD
#
${resolve}<<EOD >> resolve_eval.log_1
hklin $hklin_exp
```

```
labin $labin_exp
labin $labin_exp_cont
resolution ${dmin} ${dmax}
no_build
pattern_phase
prior_weight 1.0
hklout image_only.mtz
read_cc_map
cc_map_file dump.map
nohl
EOD
#
if ( $ip ) echo "Density-modifying image of test.pdb_1 to get image_only_dm.mtz"
#
${resolve}<<EOD >> resolve_eval.log_1
hklin image_only.mtz
labin FP=FP PHIB=PHIM FOM=FOMM  HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM
hklout image_only_dm.mtz
solvent_content $solvent_content
prior_weight 1.0
database 5
resolution ${dmin} ${dmax}
no_build
nohl
$mask_line
$use_mask_line
EOD
#
if ( $ip ) echo "Getting composite ps omit map using image_only_dm.mtz "
if ( $ip ) echo " and test.pdb_1"
#
${resolve}<<EOD >> resolve_eval.log_1
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
hklstart image_only_dm.mtz
labstart FP=FP PHIB=PHIM FOM=FOMM
hklout image_only_dm_image.mtz
solvent_content $solvent_content
database 5
resolution ${dmin} ${dmax}
composite_pdb test.pdb_
composite_pdb_start 1
composite_pdb_end 1
composite_all
prior_weight 1.0
scale_refl $scale
image
omit
```

```
mask_cycles 0
no_build
nohl
$mask_line
$use_mask_line
EOD
cp resolve_composite_map.mtz build.mtz
if ( $ip ) echo "Composite omit is in build.mtz"
#
evaluate_it:
#
#  Get R-factor for the model resolve_best.pdb...relative to hklin_exp
#
#
$refmac5 xyzin resolve_best.pdb hklin $hklin_exp xyzout refine.pdb  \
PROTOUT $CCP4_SCR/protout.dat \
PROTCOUNTS $CCP4_SCR/counts.dat \
PROTSCR $CCP4_SCR/counts.scr \
<<EOD > refmac_eval.log_${cycle}
MAKE HYDRogens No
MAKE CHECK 0
LABI FP=FP SIGFP=SIGFP FREE=FreeR_flag
REFI TYPE RESTrained RESOlution ${dmax} ${dmin}
#   Maximum likelihood refinement
REFI RESI MLKF
#   Overall B value refinement
REFI BREF OVER      ! Refine overall B-values
#   Tight restraints  Lowered further 051403
WEIG MATR 0.05
#   Scale down shifts at every cycle by factor 0.5
DAMP 0.5 0.5
#   No Blur (scale down reliability of) phases
REFI PHASed  BBLUrring 00.0  SCBLurring 1.
#  Babinet's bulk solvent scale parameters. But fix B value
SCALe TYPE BULK
SCALe LSSCale FIXBulk BBULk 200.0
#   Anisotropic scaling. It is default
SCALe LSSCale ANISotropic
#   Fix Babinet's bulkd solvent for sigmaA
SCAL MLSC FIXBulk BBULk 200.0 SCBUlk -0.05
NCYC 0
MONI MEDI
BINS 10
EOD
#-------------------------------------------------------
#
#  Get R and freeR value on refine.pdb:
set freer_line = `$grep_type 'FREE R VALUE                    :' refine.pdb|tail -1`
set r_line = `$grep_type 'R VALUE            (WORKING SET) :' refine.pdb|tail -1`
```

```
#
if ( $#freer_line > 0 ) then
 set freer = $freer_line[$#freer_line-$#freer_line]:e
else
 set freer = 99999
endif
if ( $#r_line > 0 ) then
 set r = $r_line[$#r_line-$#r_line]:e
else
 set r = 99999
endif
#
echo " "
echo "=========================================================================="
if ( $evaluate_only == YES ) then
 echo "Work R-factor for input model $pdb_in : 0.$r   free R: 0.$freer"
 echo "Evaluation of $pdb_in using prime-and-switch composite omit map:"
else
 echo "Work R-factor for resolve_best.pdb: 0.$r   free R: 0.$freer"
 echo "Evaluation using prime-and-switch composite omit map:"
endif
#
#
#   Evaluate the model relative to the composite omit ps map:
#
${resolve} <<EOD > evaluate.log
hklin build.mtz
labin FP=FP PHIB=PHIM FOM=FOMM
model resolve_best.pdb
evaluate_model
EOD
#
#==========================================================================
echo "See evaluate.log for more details on the analysis of this model"
echo " "
echo "All done...please carefully evaluate your model."
echo "Your best indicators are the free R factor and the match"
echo "to the prime-and-switch composite omit map in evaluate.log."
echo "Note that no waters are built in this model."
echo " "
echo "Output files are:"
if ( -f resolve_best.pdb ) then
 echo "resolve_best.pdb  ... partially-refined model"
endif
if ( -f resolve_best.log ) then
 echo "resolve_best.log  ... log file in which this model was built"
endif
if ( -f resolve_best.mtz ) then
$resolve<<EOD>final.log
```

```
hklin resolve_best.mtz
labin FP=FP PHIB=PHIM FOM=FOMM
mask_cycles 1
minor_cycles 0
no_build
ccp4_map_file resolve_best.map
EOD
 echo "resolve_best.mtz  ... FP, SIGFP, best overall phases, freeR_flag...."
 echo "resolve_best.map  ... ccp4-style map from resolve_best.mtz"
endif
if ( -f refmac_best.log ) then
 echo "refmac_best.log   ... log file for refinement of resolve_best.pdb"
endif
if ( -f build.mtz ) then
$resolve <<EOD>> final.log
hklin build.mtz
labin FP=FP PHIB=PHIM FOM=FOMM
mask_cycles 1
minor_cycles 0
no_build
ccp4_map_file resolve_composite_omit.map
EOD
#
 echo "resolve_composite_omit.map ... composite omit map "
endif
#==================================================================
```

SOLVE/RESOLVE home page	Back to RESOLVE table of contents

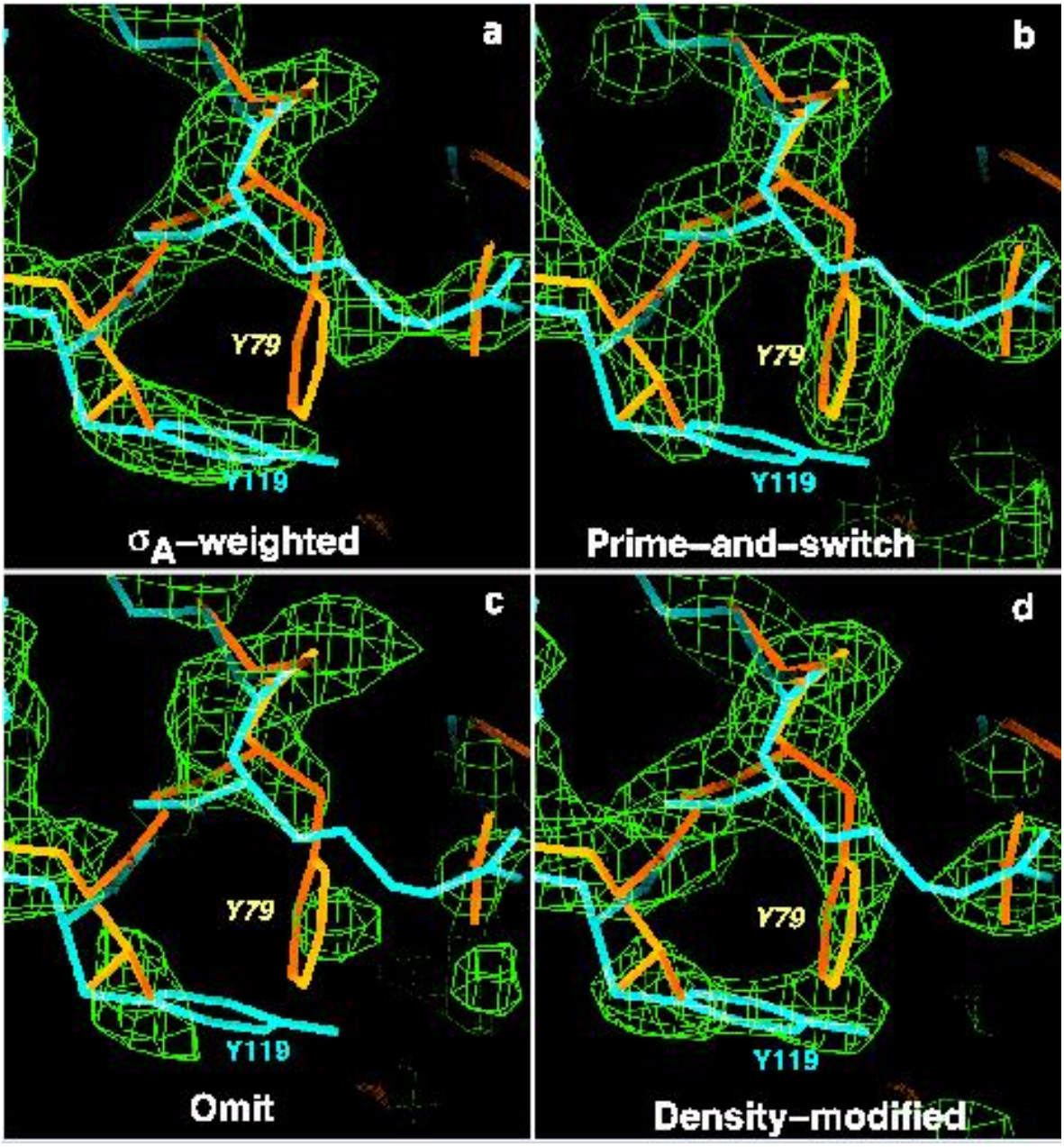
RESOLVE examples

Removal of model bias with prime-and-switch phasing	IF5A molecular replacement phasing Dehalogenase molecular replacement phasing
Improvement of experimentally- phased maps	IF5A MAD map

Removal of model bias with prime-and-switch phasing

IF5A molecular replacement phasing

- **The system: initiation factor 5A (IF5A)**
- **X-ray data on *P. aerophilum* IF5A to 2.2 A** *(Peat T.S., Newman J, Waldo G.S, Berendzen J. & Terwilliger T.C. Structure Of Translation Initiation-Factor 5a From Pyrobaculum aerophilum At 1.75 Angstrom Resolution. Structure 6, 1207-1214 (1998))*
- **60% solvent**
- **atomic model from *M. jannaschii* IF5A (***Kim, K. K, Hung, L. W., Yokota, H., Kim, R. & Kim, S.-H. Crystal structures of eukaryotic translation initiation factor 5A from Methanococcus jannaschii at 1.8 angstrom resolution. Proc. Natl. Acad Sci. USA 95, 10419-10424 (1998))*
- **RMSD between atomic model from *M. jannaschii* and refined *P. aerophilum* structure is 1.7 A.**
- **The test: phase the *P. aerophilum* data with the *M. jannaschii* structure**
- **Methods tested: (a) sigma-A, (b) prime-and-switch, (c) omit map, and (d) density modification (with RESOLVE)**
- **Results: (MR model in blue, correct model in yellow). Sigma-A map looks a lot like the model in blue used to calculate it; prime-and-switch map looks just like the correct model in yellow. Omit map is noisy, and density-modified map is halfway between sigmaA and prime-and-switch.**

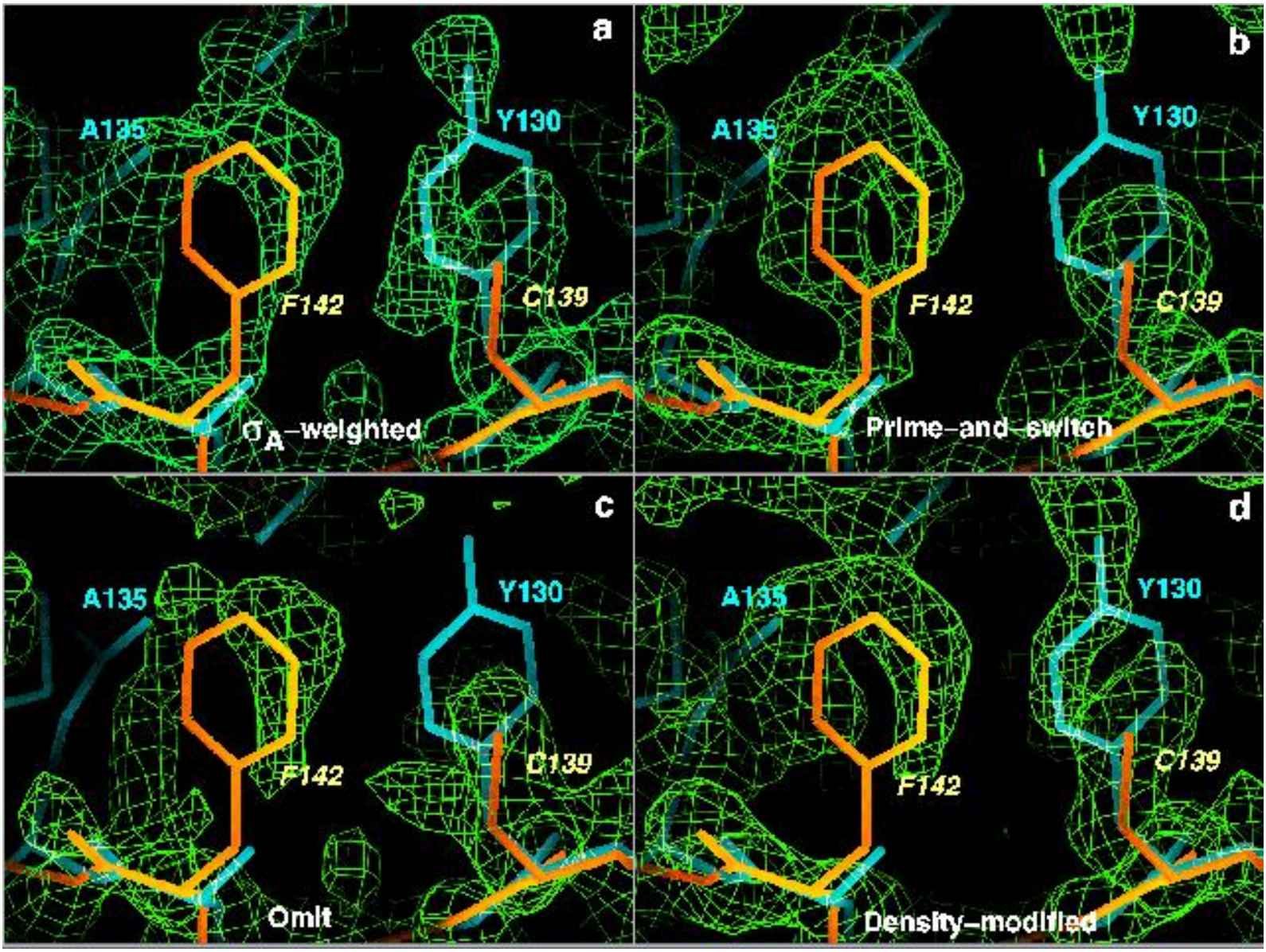


Removal of model bias with prime-and-switch phasing

Dehalogenase molecular replacement phasing

- **The system: *Rhodococcus* dehalogenase (*dhlA*)**
- **X-ray data on *Rhodococcus dhlA* to 1.5 A** *(Newman, J., Peat, T. S., Richard, R., Kan, L., Swanson, P. E., Affholter, J. A., Holmes, I. H., Schindler, J. F., Unkefer, C. J., & Terwilliger, T. C. Haloalkane dehalogenases: Structure of a Rhodococcus enzyme. Biochemistry 38, 16105-16114 (1999).)*
- **30% solvent**
- **atomic model from dehalogenase *linB* from *S. paucimobilis***(*Marek, J., Vevodova, J., Smatanova, I. K., Nagata, Y., Svensson, L. A., Newman, J., Takagi, M. & Damborsky, J. Crystal structure of the haloalkane dehalogenase from Sphingomonas paucimobilis UT26. Biochemistry 39, 14082-14086 (2000).)*

- **RMSD between atomic model from *S. paucimobilis lin B* and refined *RhodococcusdhlA* structure is 1.4 A.**
- **The test: phase the *Rhodococcus dhlA* data with the *S. paucimobilis linB* structure**
- **Methods tested: (a) sigma-A, (b) prime-and-switch, (c) omit map, and (d) density modification (with RESOLVE)**
- **Results: (MR model structure is in blue, correct model in yellow). SigmaA map looks like the MR model used to calculate it; prime-and-switch map looks like the correct model.**

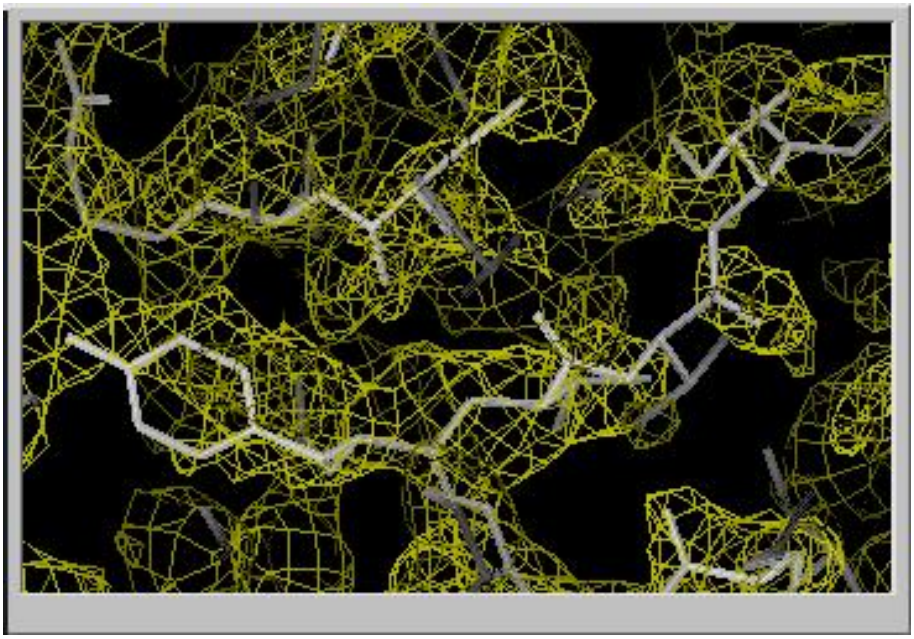


Improvement of experimentally-phased maps

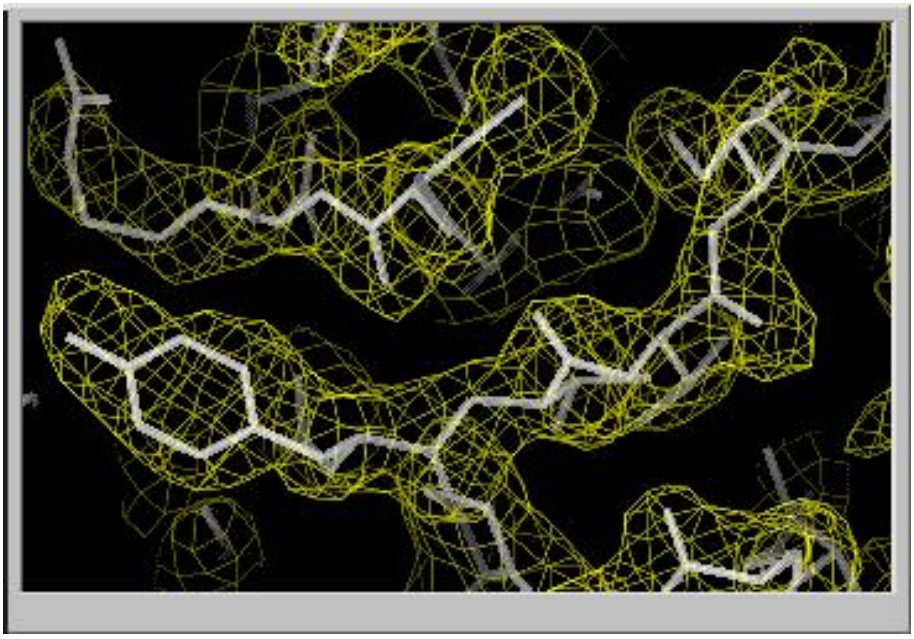
IF5A MAD map

(See: Peat, T. S., Newman, J., Waldo, G. S., Berendzen, J. & Terwilliger, T. C. (1998) "Structure of translation initiation factor 5A from Pyrobaculum aerophilum at 1.75 A resolution" Structure 15, 1207-1214.

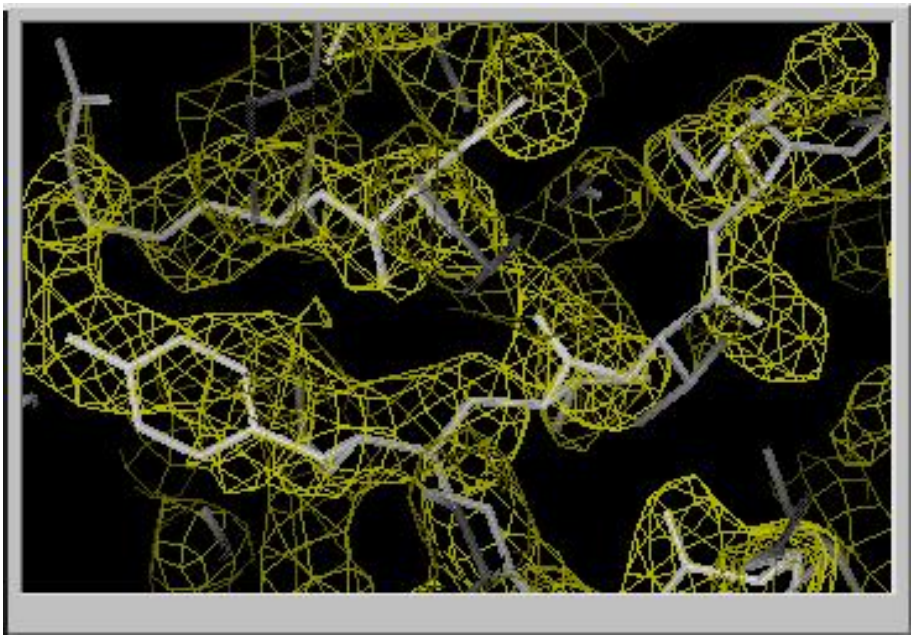
- **MAD dataset on initiation factor 5A**
- **Only 1 of 3 selenium atoms in 147 residues used in phasing (just for this test example)**
- **Solvent content was 60%**
- **Resolution of 2 A, space group I4**
- **Initial correlation of experimental map with one based on refined model was 0.37**
- **Correlation after RESOLVE was 0.79**
- **Correlation after conventional solvent flattening and histogram matching (using dm) was 0.65**



Initial MAD map (CC=0.37)



RESOLVE map (CC=0.79)



Conventional solvent flattening & histogram matching (CC=0.65)

Contents	Index

Sample scripts for SOLVE

Choose one of the following sample control files, edit it to match your data (you may need to check on data formats), and run SOLVE with it. You can watch SOLVE run by looking at the end of the "solve.status" with "tail -30f solve.status". This file will tell you where to look if something got typed in wrong and it will keep you informed about the structure determination as it goes.

If you want to change some parameters that are not listed in this file you should have a look at the list of keywords for SOLVE automated operation.

Each control file also comes with a script you can use to generate a dummy dataset and an output solve.prt file that will be similar to the one you should get if you run the control file on the dummy dataset. (Note: it won't necessarily be identical, because slight differences between machines can lead to somewhat different output files. The overall answers are usually about the same, though. The files here were generated on a DEC alpha.) You can use these to make sure everything is working properly.

Note in particular the keywords

- READDENZO/READFORMATTED/READTREK
- UNMERGED/PREMERGED
- READ_INTENSITIES/READ_AMPLITUDES

  LABIN / HKLIN (for CCP4 files, see example ccp4 MADor MIRscripts)

ADDSOLVE and ANALYZE_SOLVE. Note also if you know some of your sites at the beginning you can tell SOLVE to use them:

- under "lambda 1" or "derivative 1" specify
  - atomname xx ! xx is your atom
  - xyz x y z ! coordinates of this atom
  - .. more xyz values
- before ANALYZE_MAD or ANALYZE_MIR or SAD specify:
  - ADDSOLVE ! ADDSOLVE (add on more sites) or ANALYZE_SOLVE (refine input sites)
- SOLVE will refine these sites and go on from there.
- See sample solve.com file for ADDSOLVE.
- See sample solve.com file for ANALYZE_SOLVE.

Single datasets

Command files for running SOLVE with...

- 1 SAD dataset: generate one or solve one or look at the solve.prt output from generate followed by solve.
- 1 MAD dataset: generate one or solve one or look at the solve.prt output from generate followed by solve.
- 1 MIR dataset: generate one or solve one or look at the solve.prt output from generate followed by solve.
- 1 MIR dataset with anomalously scatter ing atoms in the native: generate one or solve one (requires version 1.10 or higher) or look at the solve.prt output from generate followed by solve.

Multiple datasets

If you have several types of datasets on the same crystal, then you can combine them into one pseudo-MIR dataset (see also the documentation on COMBINE). You can also analyze MAD data with more than one type of anomalous scatterer with this method. Edit these control files for combining datasets (Note: all these require version 1.10 or higher):

- MAD + MIR datasets: generate one or solve one or look at the solve.prt output from generate followed by solve.
- 2 MAD datasets: generate one or solve one or look at the solve.prt output from generate followed by solve.
- 2 MIR datasets: generate one or solve one or look at the solve.prt output from generate followed by solve.
- 1 MAD dataset with 2 anomalously scattering atoms: generate one or solve one or look at the solve.prt output from generate followed by solve.

Table of Contents for SOLVE on-line manual version 2.09

- SOLVE FAQS page
- Introduction to SOLVE
  - What SOLVE does
  - Getting Started
    - Scripts
    - Useful commands
    - Running more than one SOLVE job at a time
    - Symmetry files
    - Data files
  - Your license
  - The SOLVEDIR, SOLVETMPDIR, and CCP4_OPEN environmental variables
  - Getting more help
- KEYWORD and COMMAND LISTS
  - List of most commonly-used keywords that apply to SOLVE automated operation
  - List of all SOLVE keywords
  - List of all SOLVE commands
- Automated structure determination with SOLVE
  - What you will need to input to SOLVE
  - How to run SOLVE
  - Data formats for automated structure determination with SOLVE
  - Sample scripts for SOLVE
  - How to interpret the solve.status file
  - What to do next after you've run SOLVE
  - How to interpret the output from SOLVE
  - How to export your output from SOLVE to other packages
  - Commands for controlling how SOLVE works in automated mode
  - How SOLVE works
  - Solve examples
    - Gene 5 protein
    - Armadillo repeat of beta-catenin
    - Granulocyte-macrophage colony stimulating factor
- GENERATE: generating and solving model datasets with errors
- SCALING and ANALYSIS OF MIR and MAD data

    - SCALE_MAD: Scaling MAD data
    - ANALYZE_MAD: Analyzing MAD data
    - MADMRG: Converting MAD data to pseudo SIRAS
    - MADBST: Bayesian Fa estimates
    - SCALE_NATIVE: Scaling native data
    - SCALE_MIR: Scaling derivative data
    - ANALYZE_MIR: Analyzing MIR data
    - COMBINE: Combining multiple MAD or MIR datasets
    - HASSP: Searches for solutions to Patterson
    - HEAVY: Heavy atom refinement and phasing
- Local scaling and merging of data
    - Local scaling of data
    - Merging data to the asymmetric unit
    - Completeness of datasets
- DATA formats, and import/export
    - Binary formats used by SOLVE
    - Importing and Exporting data in SOLVE
    - File merging
- Working with maps
    - Calculating maps
    - Peaksearching
    - MAPtoASYM: map a PDB file to asymmetric unit or near an object
    - RHO: electron density at coordinates in a PDB file
    - Omit_map: calculate overlapping omit maps
    - Avg_omit: average overlapping omit maps
- Miscellaneous commands
    - HA_PDB: Write out a heavy atom solution in PDB format
    - COMPARE_SOLN: Comparing two heavy-atom input files
    - MATH: Perform simple operations on datasets
    - GETISO: Get isomorphous differences
    - GETANOM: Get anomalous differences and Fbar
    - GETPHASES: Get phases and F from A, B
    - Weights: weighting for macromolecular refinement
    - Bayesian difference refinement
    - CART_TO_FRACT and FRACT_TO_CART: conversion from fractional to cartesian coordinates
- New features for versions 1.18/1.19/2.01/2.02/2.03/2.04/2.05/2.06/2.08/2.09/2.10 of SOLVE/RESOLVE
- References for SOLVE

- Copyright statement for SOLVE

SOLVE Examples

This page gives three examples of what SOLVE can do. Each is an automatic structure determination carried out starting with raw data and a minimal amount of information from the user. In each case the top solution was correct and had the correct handedness (anomalous differences were used in each structure determination). The examples are:

- Gene 5 protein (MAD data, 2800 reflections to 2.6 A, 87 amino acids, 2 selenium sites)
- Armadillo repeat of beta-catenin (MAD data, 17000 reflections to 2.7 A, 537 amino acids, 15 selenium sites; data courtesy of Andy Huber and Bill Weis)
- Granulocyte-macrophage colony stimulating factor (MIR data, 4800 reflections to 3.5 A, 4 derivatives, 254 amino acids; data courtesy of Kay Diederichs)

For each structure determination, this page shows:

- A description of the structure determination by SOLVE
- The input script file used to run SOLVE
- The summary of the structure solution from the output "solve.prt"
- The end of the solve.status file that showed the progress of the structure solution file

Gene 5 protein

Summary of this structure solution:

This is a dataset with 3-wavelengths of MAD data, 2800 reflections to 2.6 A, 87 amino acids, and 2 selenium sites (Met-1, Met-77). SOLVE found both selenium sites in 34 minutes on an DEC alpha 500 MHz workstation. The Met-1 site has a very high thermal factor.

Solve.setup file listing basic information about the crystals:

```
CELL 76.08 27.97 42.36 90 103.2 90        ! cell params
symfile /usr/local/lib/solve/c2.sym       ! space group symmetry
resolution 2.6 20.0                       ! resolution limits
```

Input script file used to run SOLVE on gene 5 protein

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
```

```
#
solve <<EOD > solve.log
!command file to read in raw MAD data, scale, analyze and solve it----
checksolve                      ! compare solution with known h.a. sites
comparisonfile gvp.fft          ! FFT map using FCALC from refined model
@solve.setup                    ! get our standard information read in
logfile mad.logfile             ! write out most information to this file.
                                ! summary info will be written to solve.prt
readformatted                   ! alternatives are readdenzo, readtrek
premerged                       ! alternative is unmerged
read_intensities                ! alternative is read_amplitudes
refscattfactors                 ! alternative is fixscattfactors


mad_atomname se                 ! anomalously scattering atom is Se

lambda 1                       ! info on wavelength #1 follows
label Wavelength #  1          ! a label for this wavelength
rawmadfile test_wva.fmt        ! datafile with h k l Intensity sigma or
                               ! h k l I+ sigma+ I- sigma-
wavelength 0.9000              ! wavelength value
fprimv_mad  -1.6               ! f' value at this wavelength
fprprv_mad  3.4                ! f doubleprime value at this wavelength

! input refined h.a. coordinates (used only for comparison in "checksolve")
atomname se
 XYZ   0.4813319      0.9972169      9.4140753E-02
 XYZ   0.9731338      0.2875228      0.9446641

lambda 2
rawmadfile test_wvb.fmt
wavelength 0.9794
fprimv_mad  -8.5
fprprv_mad  4.8

lambda 3
rawmadfile test_wvc.fmt
wavelength 0.9797
fprimv_mad  -9.85
fprprv_mad  2.86
premerged
readformatted
nres 100                   [approx # of residues in protein molecule]
nanomalous 2               [approx # of anomalously scattering atoms per protein]
SCALE_MAD                  ! read in and localscale the data
ANALYZE_MAD                ! run MADMRG and MADBST and analyze all the Pattersons
SOLVE                      ! Solve the structure
EOD
```

Summary information from the "solve.prt" output file produced after completion of the automated structure determination

Correlation of anomalous differences. These indicate that the data beyond about 2.7 A are not contributing much to the phasing, as the correlation is less than 0.3.

```
            CORRELATION FOR

           WAVELENGTH PAIRS

 DMIN    1 VS 2   1 VS 3   2 VS 3

 5.20     0.90     0.72     0.80

 3.90     0.76     0.54     0.69

 3.64     0.69     0.44     0.69

 3.44     0.65     0.44     0.52

 3.25     0.59     0.27     0.50

 3.12     0.57     0.34     0.47

 2.99     0.58     0.18     0.39

 2.86     0.42     0.30     0.46

 2.73     0.33     0.17     0.31

 2.60     0.19     0.09     0.32
```

List of sites analyzed for compatibility with difference Patterson

```
  PEAK         X         Y         Z     OPTIMIZED
                                            RELATIVE OCCUPANCY
     1     0.484     1.000     0.090      91.024
     2     0.026     0.292     0.062      37.211

 Evaluation of this test soln with    2 sites after optimizing
 occupancy of each site

 Cross-vectors for sites  1 and  1 (excluding origin; 1000 = 1 sigma):
  #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
   1   -0.969   0.000  -0.181   16389.9     16570.9          2

 Cross-vectors for sites  2 and  1 (excluding origin; 1000 = 1 sigma):
```

```
 #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
  1   -0.458  -0.708  -0.028   4053.98     3387.13          1
  2   -0.510  -0.708  -0.153   4583.90     3387.13          1

Cross-vectors for sites  2 and  2 (excluding origin; 1000 = 1 sigma):
 #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
  1   -0.052   0.000  -0.125   1640.05     2769.35          2

Overall quality of this Patterson soln =  7815.40
Overall quality of the fit to patterson = 0.798126
Avg normalized peak height =  3495.15
```

Selenium atom occupancy, coordinates, and thermal factors, and Cross-validation fouriers calculated with all heavy atoms in all derivs except the site being evaluated and any sites equivalent to it.

(Peak height is height of peak at this position/rms of map)

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.484	0.997	0.093	0.526	49.596	4.90
2	0.028	0.285	0.059	0.369	60.000	3.56

Re-refinement of f' and f" values:

Final refined values of f-prime and f"

Wavelength	------- f-prime -------- last refinement	Refined	--------f"---------- last refinement	Refined
1	0.327	0.327	3.817	2.928
2	-8.419	-7.357	6.146	5.860
3	-9.609	-8.668	2.272	1.774

Figure of merit versus resolution, and anomalous and dispersive FH/E versus resolution

FIGURE OF MERIT WITH RESOLUTION

DMIN:	TOTAL	8.81	5.75	4.55	3.88	3.44	3.12	2.88	2.68
N:	2668	153	234	301	343	388	417	452	380
MEAN FIG MERIT:	0.46	0.65	0.64	0.62	0.56	0.52	0.37	0.34	0.26

RMS ANOMALOUS FH/E [f" PART OF FH / RMS ANO ERROR]:

LAMBDA: 1	0.5	0.7	0.8	0.6	0.6	0.5	0.3	0.3	0.2
LAMBDA: 2	0.6	0.8	0.8	0.7	0.7	0.6	0.5	0.4	0.3

```
LAMBDA:  3          0.3    0.6    0.6    0.5    0.3    0.3    0.2    0.2    0.1

RMS DISPERSIVE FH/E  [Delta-f-prime PART OF FH / RMS DISPERSIVE ERROR]:

L1 VS L2:           0.8    1.0    1.1    1.0    0.9    0.8    0.5    0.4    0.3
L1 VS L3:           0.9    1.2    1.3    1.0    1.0    0.9    0.6    0.5    0.4
L2 VS L3:           0.2    0.4    0.4    0.3    0.2    0.2    0.1    0.1    0.1
```

The summary of scoring for this solution

```
Summary of scoring for this solution:
                          -- over many solutions--    -- this solution --
Criteria                       MEAN          SD         VALUE        Z-SCORE
Pattersons:                    3.11         1.86         5.83         1.46
Cross-validation Fourier:      3.26         2.44         7.05         1.55
NatFourier CCx100:             27.4         5.39         31.3        0.729
Mean figure of meritx100:    0.000E+00      8.26         58.5         7.08
Correction for Z-scores:                                             -2.85

Overall Z-score value:                                                7.97
```

Note that the Patterson and cross-validation Fourier scores are 1.5 sigma above the starting solutions, but native fourier analysis is just 1 sigma above. This is both because the asymmetric unit is small and the map is fairly noisy.

The end of the solve.status file:

```
*********************************************************************
                    SOLVE STATUS      07-oct-00 10:34:53

TIME ELAPSED:    34 MIN

---------------------------------------------------------------------
CURRENT STEP:SOLVE MAIN PROGRAM
STATUS:   DONE
---------------------------------------------------------------------
---------------------------------------------------------------------
    ---TOP SOLUTION FOUND BY SOLVE  (<m> = 0.59; score =   8.03) ---

          X        Y        Z         OCCUP     B          HEIGHT/SIGMA

  2     0.484    0.997    0.093     0.526     49.6              4.3
  2     0.028    0.285    0.059     0.369     60.0              4.4

       TIME REQUIRED TO OBTAIN THIS SOLUTION:    28 MIN
---------------------------------------------------------------------
CURRENT RESOLUTION:   2.6 A.    FINAL RESOLUTION:   2.6 A.
```

Armadillo repeat region of beta-catenin **(data courtesy of Andy Huber and Bill Weis)**

Summary of this structure solution:

This is a dataset with 4 wavelengths of MAD data, 17000 reflections to 2.7 A, 537 amino acids, and 15 selenium sites. SOLVE found 14 selenium sites in 2 hours on a DEC Alpha 500 MHz workstation. The remaining 2 sites (one selenium has 2 positions) are very weak and were not included by SOLVE.

Solve.setup file listing basic information about the beta-catenin crystals:

```
resolution 2.7 20
symfile /usr/local/lib/solve/c2221.sym
cell 64.1 102.0 187.0 90 90 90
```

Input script file used to run SOLVE on beta-catenin

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#
solve <<EOD > solve.log

!command file to read in raw MAD data, scale, analyze and solve it----
title armadillo repeat of beta catenin 4-wavelength MAD data
logfile mad.logfile            ! write out most information to this file.
                               ! summary info will be written to "solve.prt"
@solve.setup                   ! get our standard information read in
readformatted                   ! or: readdenzo, readtrek, readccp4_unmerged
unmerged                        ! or; premerged

mad_atom se

refscattfactors               ! do not refine scattering factors (you can if
                              ! you want though)

        !  Comment out next line if you don't know any sites
checksolve                    ! compare solutions to the one input below

        !  Comment out next lines if you don't know the structure
! native.fft is fft calculated from catenin_y.pdb (offset +0.5 in y)
comparisonfile native.fft

lambda 1                      ! info on wavelength #1 follows
label Wavelength #  1         ! a label for this wavelength
```

```
rawmadfile l1.int
wavelength 0.9000            ! wavelength value
fprimv_mad  -1.6             ! f' value at this wavelength
fprprv_mad  3.4              ! f" value at this wavelength

! list of all SE positions in refined beta-catenin
! structure (offset by 0.5 in y from PDB file). Only if you know them

atomname se
xyz  0.2631041      0.6633824      2.8978506E-02
xyz  0.4166300      0.6113137      7.7325497E-03
xyz  0.4765674      0.7249608      2.5320712E-02
xyz  0.4591554      0.7427059      0.4719517
xyz  0.4083922      0.7455686      0.1403100
xyz  0.4416372      0.8393628      7.6342024E-02
xyz  0.1327285      0.4970000      0.4364171
xyz  9.6379094E-02  0.5855882      0.3802352
xyz  7.6066948E-02  0.6245000      0.3974865
xyz  0.1150683      0.7795883      0.3715025
xyz  0.1385160      0.7238529      0.4098982
xyz  9.2073016E-02  0.7063529      0.4022779
xyz  0.2152710      0.8265882      0.3764597
xyz  0.3304202      0.6161765      0.2311389
xyz  0.1806852      0.8512745      0.1618233

lambda 2
rawmadfile l2.int
wavelength 0.9794
fprimv_mad  -11.44
fprprv_mad  8.74

lambda 3
rawmadfile l3.int
wavelength 0.9797
fprimv_mad  -12.83
fprprv_mad  2.56

lambda 4
rawmadfile l4.int
wavelength 0.9897
fprimv_mad -2.42
fprprv_mad 1.13

nres 700                   [approx # of residues in protein molecule]
nanomalous 15               [approx # of anomalously scattering atoms per protein]
acceptance 0.10
SCALE_MAD                  ! read in and localscale the data
ANALYZE_MAD                ! run MADMRG and MADBST and analyze all the Pattersons
SOLVE                      ! Solve the structure
EOD
```

Summary information from the "solve.prt" output file produced after completion of the automated structure determination of beta-catenin

Correlation of anomalous differences. These indicate that the data all the way to about 2.7 A are contributing to the phasing, as the correlation >0.3 for wavelengths 1 vs 2.

CORRELATION FOR

WAVELENGTH PAIRS

DMIN	1 VS 2	1 VS 3	1 VS 4	2 VS 3	2 VS 4	3 VS 4
5.40	0.84	0.66	0.42	0.79	0.41	0.35
4.05	0.75	0.53	0.36	0.69	0.35	0.33
3.78	0.65	0.43	0.21	0.60	0.23	0.19
3.58	0.67	0.38	0.24	0.58	0.27	0.22
3.38	0.56	0.31	0.19	0.50	0.19	0.17
3.24	0.53	0.28	0.12	0.40	0.14	0.14
3.11	0.48	0.21	0.14	0.36	0.18	0.16
2.97	0.44	0.25	0.11	0.38	0.18	0.11
2.84	0.41	0.21	0.08	0.32	0.13	0.06
2.70	0.33	0.11	0.10	0.25	0.13	0.11
ALL	0.63	0.37	0.22	0.52	0.24	0.19

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.833	0.115	0.231	66.078
2	0.424	0.125	0.102	60.406
3	0.944	0.337	0.076	55.988
4	0.368	0.997	0.062	56.693

```
     5     0.681     0.354     0.162      49.508
     6     0.403     0.083     0.118      54.275
     7     0.049     0.243     0.028      41.397
     8     0.972     0.226     0.025      52.006
     9     0.389     0.281     0.127      42.309
    10     0.292     0.326     0.125      31.510
    11     0.410     0.212     0.095      27.341
    12     0.361     0.226     0.090      39.093
    13     0.910     0.250     0.144      22.225
    14     0.889     0.111     0.012      24.421

 Evaluation of this test soln with   14 sites after optimizing
 occupancy of each site

 Cross-vectors for sites  1 and  1 (excluding origin; 1000 = 1 sigma):
  #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
   1   -1.667  -0.229   0.500   7041.00     8732.56          2
   2   -1.667   0.000   0.037   6518.80     8732.56          2
   3    0.000  -0.229  -0.463   6185.17     8732.56          2

 Cross-vectors for sites  2 and  1 (excluding origin; 1000 = 1 sigma):
  #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
   1   -0.410   0.010  -0.130   6028.85     3991.49          1
   2   -1.257  -0.240   0.370   6627.21     3991.49          1
   3   -1.257   0.010   0.167   7920.79     3991.49          1
   4   -0.410  -0.240  -0.333   5973.46     3991.49          1

 Cross-vectors for sites  2 and  2 (excluding origin; 1000 = 1 sigma):
  #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
   1   -0.847   0.000   0.296   4925.09     7297.73          2
   2    0.000  -0.250  -0.204   5302.47     7297.73          2
```

(etc. for many many more cross-vectors)

Selenium atom occupancy, coordinates, and thermal factors, and Cross-validation fouriers calculated with all heavy atoms in all derivs except the site being evaluated and any sites equivalent to it.

(Peak height is height of peak at this position/rms of map)

```
  Site    x       y       z       occ       B     -- PEAK  HEIGHT --

    1   0.830   0.116   0.231   0.691  38.214            30.54
    2   0.422   0.124   0.103   0.671  44.433            25.98
    3   0.943   0.338   0.076   0.706  42.818            25.65
    4   0.367   0.996   0.063   0.527  15.000            24.49
    5   0.679   0.353   0.162   0.641  60.000            20.66
    6   0.406   0.084   0.119   0.574  32.160            22.10
    7   0.045   0.243   0.028   0.539  48.318            17.88
```

8	0.970	0.225	0.026	0.764	60.000	18.08
9	0.386	0.281	0.128	0.306	15.000	15.63
10	0.289	0.326	0.125	0.303	33.767	12.07
11	0.409	0.211	0.095	0.310	36.022	10.80
12	0.362	0.225	0.091	0.192	15.000	12.00
13	0.910	0.250	0.145	0.289	31.524	8.22
14	0.891	0.110	0.011	0.456	60.000	8.36

Re-refinement of f' and f'' values:

Final refined values of f-prime and f"

Wavelength	------- f-prime -------- last refinement	Refined	--------f"---------- last refinement	Refined
1	-2.206	-2.206	5.365	4.357
2	-10.957	-11.069	11.971	7.525
3	-12.631	-12.740	3.032	2.000
4	-2.714	-2.507	1.232	0.563

Figure of merit versus resolution, and anomalous and dispersive FH/E versus resolution

FIGURE OF MERIT WITH RESOLUTION

DMIN:	TOTAL	9.09	5.96	4.72	4.03	3.57	3.24	2.99	2.79
N:	17155	946	1466	1815	2122	2386	2623	2798	2999
MEAN FIG MERIT:	0.55	0.70	0.74	0.67	0.59	0.56	0.52	0.47	0.41

RMS ANOMALOUS FH/E [f" PART OF FH / RMS ANO ERROR]:

LAMBDA: 1	0.6	1.1	1.2	0.9	0.7	0.6	0.5	0.4	0.4
LAMBDA: 2	1.0	1.2	1.3	1.2	1.1	1.0	0.9	0.8	0.6
LAMBDA: 3	0.3	0.4	0.6	0.4	0.3	0.3	0.3	0.2	0.2
LAMBDA: 4	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1

RMS DISPERSIVE FH/E [Delta-f-prime PART OF FH / RMS DISPERSIVE ERROR]:

L1 VS L2:	1.0	1.4	1.5	1.3	1.1	1.0	0.8	0.7	0.6
L1 VS L3:	1.1	1.5	1.6	1.4	1.2	1.1	0.9	0.8	0.7
L1 VS L4:	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
L2 VS L3:	0.3	0.6	0.5	0.4	0.4	0.3	0.2	0.2	0.1
L2 VS L4:	1.0	1.3	1.5	1.3	1.1	1.0	0.9	0.8	0.6
L3 VS L4:	1.2	1.4	1.7	1.4	1.2	1.1	1.0	0.9	0.8

.1

The summary of scoring for this solution

```
Summary of scoring for this solution:
                          -- over many solutions--    -- this solution --
Criteria                       MEAN          SD          VALUE        Z-SCORE
Pattersons:                    4.89        0.745         14.2         12.5
Cross-validation Fourier:      21.7         4.61         185.         35.5
NatFourier CCx100:             11.0         5.29         58.9         9.05
Mean figure of meritx100:    0.000E+00      5.00         69.3         13.9
Correction for Z-scores:                                             -12.1

Overall Z-score value:                                                58.8
```

Note that the Z-score for this solution (59) is much higher than for the gene 5 protein example, even though the phasing is about the same. This is because SOLVE scoring gets higher for datasets with more sites.

The end of the solve.status file:

```
*****************************************************************************
              SOLVE STATUS      07-oct-00 11:46:31

 DATASET TITLE: armadillo repeat of beta catenin 4-wavelength MAD data
 TIME ELAPSED:     2 HR

 ---------------------------------------------------------------------------
 CURRENT STEP:SOLVE MAIN PROGRAM
 STATUS:   DONE
 ---------------------------------------------------------------------------
 ---------------------------------------------------------------------------
    ---TOP SOLUTION FOUND BY SOLVE  (<m> = 0.69; score =  58.80) ---

        X       Y       Z       OCCUP    B        HEIGHT/SIGMA

  2    0.830   0.116   0.231   0.691    38.2        30.5
  2    0.422   0.124   0.103   0.671    44.4        26.0
  2    0.943   0.338   0.076   0.706    42.8        25.7
  2    0.367   0.996   0.063   0.527    15.0        24.5
  2    0.679   0.353   0.162   0.641    60.0        20.7
  2    0.406   0.084   0.119   0.574    32.2        22.1
  2    0.045   0.243   0.028   0.539    48.3        17.9
  2    0.970   0.225   0.026   0.764    60.0        18.1
  2    0.386   0.281   0.128   0.306    15.0        15.6
  2    0.289   0.326   0.125   0.303    33.8        12.1
  2    0.409   0.211   0.095   0.310    36.0        10.8
  2    0.362   0.225   0.091   0.192    15.0        12.0
  2    0.910   0.250   0.145   0.289    31.5         8.2
  2    0.891   0.110   0.011   0.456    60.0         8.4

     TIME REQUIRED TO OBTAIN THIS SOLUTION:    38 MIN
```

```
 ---------------------------------------------------------------------------
 CURRENT RESOLUTION:   2.7 A.    FINAL RESOLUTION:   2.7 A.
~
```

Granulocyte-macrophage colony stimulating factor

Summary of this structure solution:

This is an MIR dataset with 4800 reflections to 3.5 A, 4 derivatives, and 254 amino acids. The data is courtesy of Kay Diederichs. The derivatives are not that good and the overall figure of merit of the structure is only 0.54 to 3.5 A. Using all the data and including anomalous differences, SOLVE took 18 minutes to solve this MIR problem on a DEC Alpha 500 MHz workstation. Solve only used 3 of the 4 derivatives in phasing.

Solve.setup file listing basic information about the crystals:

```
cell 47.6 59.1 126.7 90 90 90
symfile p212121.sym
resolution 20 3.5
```

Input script file used to run SOLVE on Granulocyte-macrophage colony stimulating factor

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
solve <<EOD > solve.log
! solve.com for gmf 7-25-97
! include known h.a. sites for comparison and fft map as well

logfile mir.logfile ! write out most information to this file.
@solve.setup
title gm native + 4 derivatives
!
readformatted
premerged
checksolve
comparisonfile gm_offset.fft
rawnativefile gmnat.fmt
derivative 1
```

```
inano
noanorefine
label deriv 1  gm18 pcmbs
rawderivfile gm18.fmt
 ATOMNAME Hg
 nsolsite_deriv 2
 XYZ   0.4069387      0.5974227      0.1901610
 XYZ   0.4322624      0.5161777      0.2020042
derivative 2
inano
noanorefine
label deriv 2 gmPt(EtNH2)2Cl2 derivative #40
rawderivfile gm40.fmt
 ATOMNAME Pt
 nsolsite_deriv 6
derivative 3
inano
noanorefine

label mersalyl acid # 52
rawderivfile gm52.fmt
 ATOMNAME Hg
 nsolsite_deriv 2
 XYZ   0.9070011      0.4427668      0.1972121
 XYZ   0.4240658      0.6003970      0.1951167
derivative 4
inano
noanorefine
label HgI2 #57
rawderivfile gm57.fmt
 ATOMNAME Hg
 nsolsite_deriv 3
 XYZ   0.9747854      0.4725027      0.2089491
 XYZ   0.3438405      0.6067868      0.1840420
acceptance 0.35       ! accept new sites with ~50% of height of avg
scale_native
scale_mir
analyze_mir
SOLVE
EOD
```

Summary information from the "solve.prt" output file produced after completion of the automated structure determination

Selenium atom occupancy, coordinates, and thermal factors, andCross-validation fouriers calculated with all heavy atoms inall derivs except the site being evaluated and any sites equivalent to it.

```
(Peak height is height of peak at this position/rms of map)

  Site    x       y       z       occ       B     -- PEAK  HEIGHT --
```

```
Deriv 1:
    1   0.404   0.600   0.192   0.130  60.000             14.80
    2   0.923   0.440   0.200   0.112  60.000             12.27

Deriv 2:

Deriv 3:
    1   0.423   0.600   0.195   0.169  60.000             21.90
    2   0.908   0.443   0.197   0.223  60.000             24.02

Deriv 4:
    1   0.973   0.481   0.210   0.126  60.000             12.80
    2   0.326   0.568   0.188   0.052  16.111              9.09
    3   0.357   0.637   0.182   0.040  25.935              3.11
```

Figure of merit versus resolution

```
 DMIN:           TOTAL   11.16   7.54   6.04   5.18   4.61   4.19   3.87   3.61
 N:                4801    297    418    525    589    663    713    778    818
 MEAN FIG MERIT:   0.54   0.70   0.76   0.73   0.64   0.52   0.48   0.42   0.34
```

List of sites analyzed for compatibility with difference Patterson

(Height is 1000 x height of peak in Patterson/rms of map. Predicted height is expected height based on occupancy of sites)

```
Derivative 1:

   PEAK         X         Y         Z     OPTIMIZED
                                             RELATIVE OCCUPANCY
      1     0.406     0.597     0.191      47.139
      2     0.927     0.438     0.198      40.744

 Evaluation of this test soln with    2 sites after optimizing
 occupancy of each site

 Cross-vectors for sites  1 and  1 (excluding origin; 1000 = 1 sigma):
  #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
   1   -0.312  -1.194   0.500   4539.24     4444.25          2
   2   -0.812   0.500   0.118   2537.75     4444.25          2
   3    0.500  -0.694  -0.382   4848.71     4444.25          2

 Cross-vectors for sites  2 and  1 (excluding origin; 1000 = 1 sigma):
  #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
   1    0.521  -0.160   0.007   4568.46     1920.64          1
   2   -0.833  -1.035   0.507   4088.19     1920.64          1
   3   -1.333   0.340   0.111   3195.59     1920.64          1
   4    1.021  -0.535  -0.389   2322.98     1920.64          1
```

```
Cross-vectors for sites  2 and  2 (excluding origin; 1000 = 1 sigma):
 #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
  1   -1.354  -0.875   0.500   1348.34     3320.13          2
  2    0.500  -0.375  -0.396   3381.48     3320.13          2
Total of            1 of           11 patterson peaks used more than once.

Overall quality of this Patterson soln =  5946.31
Overall quality of the fit to patterson =  1.38812
Avg normalized peak height =  1792.88
```

(... etc for derivatives 2,3,4).

.

Summary of scoring for this solution:

```
Summary of scoring for this solution:
                          -- over many solutions--    -- this solution --
Pattersons:                   1.87        0.500         3.82         3.89
Cross-validation Fourier:     6.63         3.49         36.3         8.51
NatFourier CCx100:            12.4         5.31         32.8         3.83
Mean figure of meritx100:    0.000E+00     7.53         53.5         7.11
Correction for Z-scores:                                             -3.92

Overall Z-score value:                                                19.4
```

Tail end of the solve.status file:

```
*************************************************************************
                    SOLVE STATUS      07-oct-00 10:18:51

DATASET TITLE: gm native + 4 derivatives
TIME ELAPSED:    18 MIN

-------------------------------------------------------------------------
CURRENT STEP:SOLVE MAIN PROGRAM
STATUS:   DONE
-------------------------------------------------------------------------
-------------------------------------------------------------------------
    ---TOP SOLUTION FOUND BY SOLVE  (<m> = 0.54; score =  19.43) ---

Deriv     X        Y        Z         OCCUP     B          HEIGHT/SIGMA


  1     0.404    0.600    0.192     0.130     60.0             14.8
  1     0.923    0.440    0.200     0.112     60.0             12.3

  3     0.423    0.600    0.195     0.169     60.0             21.9
  3     0.908    0.443    0.197     0.223     60.0             24.0
```

```
  4     0.973    0.481    0.210     0.126     60.0            12.8
  4     0.326    0.568    0.188     0.052     16.1             9.1
  4     0.357    0.637    0.182     0.040     25.9             3.1

       TIME REQUIRED TO OBTAIN THIS SOLUTION:    16 MIN
-------------------------------------------------------------------------
CURRENT RESOLUTION:   3.5 A.    FINAL RESOLUTION:   3.5 A.
```

e!doctype html public "-//w3c//dtd html 4.0 transitional//en">

All keywords for SOLVE

This is a list of all the keywords for SOLVE. Keywords set values of parameters (number of heavy atom sites, f' value), while commands cause SOLVE to do something (solve a structure, scale data, search for heavy atoms, draw a map)

Also see the list of common keywords that apply to automated SOLVE operation and the list of all commands.

The keywords for SOLVE are listed here in the following groups:

- Overall control parameters
- Crystal data (cell, resolution, etc)
- Reading in,scaling and rejecting data
- Specifying file names
- Column numbers for data in data files
- Control parameters for SOLVE structure determination
- Defining heavy atom scattering factors
- Input of heavy atom parameters
- Control parameters for heavy atom refinement and phasing
- Control parameters for HASSP
- Working with maps
- Grids used for FFT
- Control parameters for GENERATE

Overall control parameters

```
VERBOSE              write out a lot of output to logfile
NSHELLS n            Number of shells for analysis [default=10]. Does not
                     apply to heavy atom refinement (fixed at 8).
SAVE_FILES           Save scratch files instead of cleaning up
```

Crystal data (cell, resolution, etc)

```
SYMFILE xxxxx       symmetry file for this space group
CELL   a b c alpha beta gamma
RESOLUTION dmin dmax
TITLE       xxxxx      CCP4 overall Title (60 characters, spaces allowed)
PROJECTNAME xxxxx      CCP4 Project Name (20 characters, no spaces)
CRYSTALNAME  xxxxx      CCP4 Crystal Name (20 characters, no spaces)
DATASETNAME xxxxx      CCP4 Dataset Name (20 characters, no spaces)
```

```
RES_PHASE   XX       Just use data up to XX in phasing and heavy-atom
                     searches, write out all data defined by dmin and dmax
NRES   n             # of residues in asymmetric unit [default=100].  Used to
                     estimate overall scale and (along with nanomalous) how big
                     Fa values might be.

NANOMALOUS n         # of anomalously scattering atoms in asymmetric unit.
                     Used to estimate how big the Fa values might be


SN_MIN  xx           Identify working resolution as the point where signal-
                     to-noise in the data goes down to about XX. Default =0.5

SN_RATIO_MIN  xx     Identify working resolution as the point where signal-
                     to-noise in the data goes down to about XX times its
                      maximum value. Larger of value of S/N obtained by SN_MIN and
                      SN_RATIO_MIN used. Default = 0.1
```

Reading in scaling, and rejecting data

```
READ_INTENSITITES  (default)  The raw data files contain intensity measurements
READ_AMPLITUDES    The raw data files contain amplitudes (F) not intensities (I)
                   (This is valid only with READFORMATTED)

PREMERGED          The data in all RAWMADFILEs have H K L and 4 other columns:
                   I+/F+, sigma, I-/F-, sigma
UNMERGED           The data in all RAWMADFILEs have H K L and 2 other columns:
                   I/F, sigma

READDENZO        All datafiles are written by Scalepack.  For unmerged data
                 they will be read with the formatting:(6i4,i6,2i2,i3,2f8.0) and
                 nsym*2+1 lines are skipped at the top of the file. For
                 merged data the formatting is:  (3i4,4f8.0) and
                 3 lines are skipped at the top of the file.

READFORMATTED      All datafiles will be read with "*" formatting and
                   contain H K L I/F sigma or H K L I+/F+ sigma I-/F- sigma

READTREK          The datafiles were written by d*trek and contain columns
                  with intensities

READCCP4_UNMERGED  All datafiles will be read as CCP4 mtz files assuming that
                   LABIN is defined as I=I SIGI=SIGI. No LABIN lines are allowed
                   (you cannot redefine LABIN).

LABIN             specify column assignments for HKLIN in standard CCP4
                  fashion (FC=FC1 PHIC=PHIC FOM=FOM) etc

HKLIN  xxx.mtz    mtz file containing scaled amplitudes
```

```
PHASES_FORMATTED xxx.fmt   File  xxx.fmt contains H K L FC PHIC FOM and
                     the phases and fom will be used in SOLVE with
                     difference Fouriers to find initial sites

PHASES_LABIN         specification of column assignments for PHASES_MTZ file
                     Normal use is FC=FC PHIC=PHIC FOM=FOM. NOTE: MUST
                     come before PHASES_MTZ!

PHASES_MTZ xxx.mtz    as PHASES_FORMATTED, but mtz-file. FC PHIC FOM required

NSKIP n             Skip exactly n lines at the top of each data file
NSKIP 0             Do not skip any lines at the top of each data file
NSKIP -1            Skip 0 lines at the top of each data file
                    unless the keywords READDENZO and PREMERGED
                    are set in which case the default number of lines
                    are skipped (see above)

RATMIN  xx          Minumum ratio of F/sig or I/sig to read in data for a
                    reflection at all is xx [default=2.0].  This is
                    used to eliminate weak data.

SIGMA_I_RATIO XX    Data read in with "premerged" will be given sigmas of
                    at least XX. Default=0.0

FPFM_ONLY           Toss all acentric reflections where either F+
                    or F- is missing [this is the default for MAD data]
FP_OR_FM            Use F+ or F- as an estimate of Fbar if F+ and
                    F- are not both present.

SWAP_ANO            Swap H K L -> -H -K -L as data are read in to
                    SOLVE in scale_native, scale_derivative, and
                    scale_mad. This is to correct for a detector or indexing
                    that swapped F+ for F-
OVERALLSCALE        Do not do local scaling; just an overall
                    scale factor for F+, F- at each wavelength.
                    Use this if you already have scaled the data
                    and you don't want any more scaling done.
KEEPALL            keep reflections even with high differences
TOSSBAD            (default)Toss reflections if differences between native and
                   derivative are more than 3 * the rms found for other
                   reflections.
                   Note: KEEPALL and TOSSBAD apply to MERGE, LOCALSCALE,
                   SCALE_MAD, SCALE_MIR, SCALE_NATIVE. This is the
                   place to reject derivative reflections with very large del F
                   if you want to reject them at all.
ANCUT              minimum # of reflections to use to scale a reflection (30.)
RATMIN             minimum ratio of F/sigma to include (default=2)
NOBFACTOR          if specified, do not apply overall Wilson scaling before doing
                   local scaling. Generally used only along with DAMPING=0.
BFACTOR            undoes NOBFACTOR. Do apply Wilson scaling before local scaling
DAMPING xx         scale factor (after Wilson scaling) is damped by taking it
                   to the power xx. Generally used with NOBFACTOR and a value of
```

```
                  0 to not do any scaling at all.
NODAMPING         undoes DAMPING by resetting damping factor to 1.0
OVERALLSCALE      just get 1 scale factor for the whole dataset. No local
                  scaling, no wilson scaling. Same as NOBFACTOR + DAMPING 0.0
NOOVERALLSCALE    undoes OVERALLSCALE. SAME AS BFACTOR + DAMPING 1.0
```

Specifying file names

```
LOGFILE  xx       name of output file for summary of results is

SYMFILE      xxxxx       symmetry file for this space group
INFILE  data.drg    Standard datafile for input.  Some routines require that
                    other input file names are specified.
OUTFILE  data.xplor Standard output file.   Some routines require that
                    other output file names are specified.

RAWMADFILE     xxx.int   read in xxx.int as data for the current mad wavelength
RAWNATIVEFILE  xxx.int   read xxx.int as data for the native
RAWDERIVFILE   xxx.int   read xxx.intas data for the current derivative
EXPORTFILE   xxx    Formatted file with Fp,phase, m, and Hendrickson-
                    Lattman coefficients will be written to this
                    file at the end  of SOLVE.  Only the top solution
                    will be written out in this way.  The file has
                    a header that you can edit and make into a little
                    script file that will read the data into ccp4
                    format. Default is phases-hl.export

PHASEFILE           binary .drg file with the same data as the EXPORTFILE
                    Default is phases-hl.drg

NEWSCRIPTFILE       script file that will run HEAVY and write out a
                    new EXPORTFILE,PHASEFILE, and NEWSCRIPTFILE.
                    Default name is "phases-hl.script".  All
                    this script file does is calculate phases.  The file
                    does contain the final heavy atom parameters. If you
                    want to take the heavy atom parameters and continue
                    in SOLVE with them (i.e., running ADDSOLVE or
                    ANALYZE_SOLVE), you should copy these heavy atom
                    parameters into the solve_mad.script or
                    solve_mir.script file that was
                    written by "ANALYZE_MAD" or
                    "ANALYZE_MIR" and make
                    a new script file this way.
MADFBARFILE xx.scl  Output file from SCALE_MAD with (Fbar,sigma,DelAno,sigma)
                    for each wavelength will be xx.scl
                    (DEFAULT="mad_fbar.scl")

MADFPFMFILE yy.scl Output file from SCALE_MAD with (F+,sigma,F-,sigma) for each
                    wavelength will be yy.scl
                    (DEFAULT="mad_fpfm.scl")
```

```
madmrgfile xxx.out    SIRAS-like MAD dataset from MADMRG
                    [     default="madmrg.out"]

madbstfile yyy.out  coefficients for a Bayesian Patterson to yyy.out from MADBST
                          [default="madbst.out"]

SOLVEDATAFILE  xxx  Output datafile with MADMRG and MADBST data

NEWFILE  xx       file with updated script file for HEAVY
OUTFILE   xx     output file name for HEAVY, required if KOUT is not 0


FFTFILE   xx        name of FFT-containing file
EZDMAPFILE   xx       name of output EZD format map file
CCP4MAPFILE   xx       name of output ccp4 format map file
MAPVIEWFILE xx      name of output MAPVIEW format file
COMPARISONFILE   xx   name of file with FFT to be compared with all native
                    fouriers from trial solutions (goes with checksolve).

FILETITLE xxx       optional title for a file
```

Column numbers for data in data files

```
NNATF n           column # for F of native data
NNATS n           column # of sigma of F of native data
NDERF n           column # for F of deriv data
NDERS n           column # for sigma of F of deriv data
NANOF n           column # of anomalous difference (Fplus-Fminu) of deriv data
NANOS n           column # of sigma of anomalous difference
NCOLF_MERGE n     column number in input file for F (default = 1)
NCOLSIG_MERGE n   column number in input file for sigma of F (default =2)

NCOLFBAR   n        Fbar for this wavelength
NCOLSFBAR  n        sigma of Fbar
NCOLDELF   n        Del F ano (Fplus - Fminus)
NCOLSDELF  n        sigma of del F ano
NCOLFPLUS               column number for Fplus
NCOLSIGPLUS             column number for sigma of Fplus
NCOLFMINUS              column number for Fminus
NCOLSIGMINUS            column number for sigma of Fminus
NCOLFC n                WT Fc (WT Fcalc) for Fdiff
NCOLFOWT n              WT Fo (WT Fobs) for Fdiff
NCOLSWT n               sigma of WT Fo for Fdiff
NCOLFOMUT n             MUT Fo (MUT Fobs) for Fdiff
NCOLSMUT n              sigma of MUT Fo for Fdiff
NCOLFCMUT n             MUT Fc for Fdiff
NCOLRTEST n             RTEST indicator (0 if missing) for Fdiff
NXPLORF  n          column for F in drgtoxplor
NXPLORSIG n         column for sigma in drgtoxplor
```

```
JSTD    n           wavelength ID for wavelength to be considered the STANDARD
```

Control parameters for SOLVE structure determination

```
new_dataset        Used to identify the beginning of a new dataset
                       in combination with COMBINE_ALL_DATA.
                       Cannot be used with mtz files.
ntopfour xx         Number of Fourier peaks to pick from a map
ntopderiv xx        Number of Fourier peaks to be tested for
nsolsite xx         Maximum number of sites in a derivative unless overridden by
                    nsolsite_deriv
nseedtest xx        Number of seeds per derivative to try (before sorting)
nseedsolve xx       Number of seeds (total) to try after sorting them
ntopsolve xx        Number of solutions to print out at the end and number
                    of solutions to keep track of at any one time
addsolve            Add on to solution that is input[default=off]
checksolve          Compare all solutions to input solution [default=off]

analyze_solve       Analyze input solution without doing anything else
                    [default=off]

no_fom             do not use figure of merit in solve scoring
                   (this is default in P1 where figure of merit is poorly
                   estimated)
no_patterson       do not use patterson in solve scoring
no_fourier         do not use cross-validation difference fouriers in scoring
no_native_fourier  do not use analysis of native fourier in solve scoring

[no]delete          do [not] check out all solutions by testing
                    all one-site deletions [default=delete]

[no]inverse         do [not] check out all solutions by testing
                    their inverses (does not apply if a solution
                    is centrosymmetric or if anomalous differences
                    are not used). [default=inverse]
full_inverse        if inverse is set, calculate all scores for inverse, do not
                    copy over patterson and cross-validation difference fourier

SCORING_TABLE (8 values)  Scoring table (usually generated by SOLVE) consisting
                        of mean and standard deviation of scores for trial
                        solutions for Pattersons, Cross-fouriers, Native Fourier
                        maps, and mean figure of merit.  This keyword is useful
                        when you are running SOLVE after modifying the script
                        file it writes out at the end.

QUICK                  once a plausible solution is found, don't keep looking,
                       just add on sites to it and check it at the end.

QUICKER                Go with the best solution at each stage, but try all seeds ,
```

unless a very good solution is found.

VERYQUICK SOLVE will go with the best solution at each step, even if it is not very good. This will speed up SOLVE a lot for good data.

THOROUGH keep looking anyways until a limit set by ntopsolve, nseedsolve, etc is reached. Opposite of QUICK.

FINISH control parameter for solve.control file indicating that SOLVE should finish up as soon as possible without looking for any new solutions.

RESOLUTION_STEPS number of steps of resolution to use in the search for heavy-atom solutions. Default=3. If you specify 0 or 1 it will go right to the highest resolution available

NTOL_SITE a site within ntol_site grid units of an existing site is considered to be a duplicate and is ignored. [default=8]

NTOL_SOLN a heavy-atom solution for which every site matches another solution within ntol_soln grid units is considered to be a duplicate and is ignored. [default=2]

ACCEPTANCE xx the weighting function for scoring patterson and free-difference fourier peak heights is adjusted so that a new site with height relative to the previous average height of ACCEPTANCE or higher will generally give a solution with a higher score than the solution without this site. [default =0.2]

CUT_DELETE xx Only sites with free-difference Fourier peak heights less than XX *sigma of map will be will be considered for removal in generating new test solutions. Default = 5.0

bayes Use Bayesian MAD phasing at the very end of SOLVE. (This is the default)

nobayes Use the compressed MADMRG datafile for all phasing when program gets to SOLVE.

NO_ANISOTROPIC_B no anisotropic b used in automated structure determination

no_duplicate_sites Do not allow the same site to appear in more than one derivative.(default=off)

CONTINUE_SAD Continuing a SAD dataset (keyword used by SOLVE in solve_fast_sad.script

CONTINUE_MAD Continuing a SAD dataset (keyword used by SOLVE in

```
                      solve_mad.script

USE_INPUT_PHASES     Using input phases (must be defined in input file)
                     (keyword used by SOLVE in solve_mad.script and solve_mir.script)
```

Defining heavy atom scattering factors

```
The atom types recognized by SOLVE are:
H, H-1, He, Li, Li+1, Be, Be+2, B, C, Cv, N, O, O-1,
F, F-1, Ne, Na, Na+1, Mg, Mg+2, Al, Al+3, Si, Siv, Si+4,
P, S, Cl, Cl-1, Ar, K, K+1, Ca, Ca+2, Sc, Sc+3, Ti, Ti+2,
Ti+3, Ti+4, V, V+2, V+3, V+5, Cr, Cr+2, Cr+3, Mn, Mn+2, Mn+3,
Mn+4, Fe, Fe+2, Fe+3, Co, Co+2, Co+3, Ni, Ni+2, Ni+3, Cu,
Cu+1, Cu+2, Zn, Zn+2, Ga, Ga+3, Ge, Ge+4, As, Se, Br,
Br-1, Kr, Rb, Rb+1, Sr, Sr+2, Y, Y+3, Zr, Zr+4, Nb, Nb+3,
Nb+5, Mo, Mo+3, Mo+5, Mo+6, Tc, Ru, Ru+3, Ru+4, Rh, Rh+3,
Rh+4, Pd, Pd+2, Pd+4, Ag, Ag+1, Ag+2, Cd, Cd+2, In, In+3,
Sn, Sn+2, Sn+4, Sb, Sb+3, Sb+5, Te, I, I-1, Xe, Cs, Cs+1,
Ba, Ba+2, La, La+3, Ce, Ce+3, Ce+4, Pr, Pr+3, Pr+4, Nd,
Nd+3, Pm, Pm+3, Sm, Sm+3, Eu, Eu+2, Eu+3, Gd, Gd+3, Tb,
Tb+3, Dy, Dy+3, Ho, Ho+3, Er, Er+3, Tm, Tm+3, Yb, Yb+2,
Yb+3, Lu, Lu+3, Hf, Hf+4, Ta, Ta+5, W, W+6, Re, Os, Os+4,
Ir, Ir+3, Ir+4, Pt, Pt+2, Pt+4, Au, Au+1, Au+3, Hg, Hg+1,
Hg+2, Tl, Tl+1, Tl+3, Pb, Pb+2, Pb+4, Bi, Bi+3, Bi+5, Po,
At, Rn, Fr, Ra, Ra+2, Ac, Ac+3, Th, Th+4, Pa, U, U+3, U+4,
U+6, Np, Np+3, Np+4, Np+6, Pu, Pu+3, Pu+4, Pu+6, Am, Cm, Bk, Cf

newatomtype xxxx    define scattering properties of atom xxxx
aval a1 a2 a3 a4    4 real numbers (a1,a2,a3,a4)  from International Tables for
                    the most recently defined newatomtype
bval b1 b2 b3 b4    b values for newatomtype (NOTE: not the same as BVALUE).
cval c              c value for newatomtype
fprimv xx           f' value for newatomtype
fprprv xx           f" value for newatomtype

                    Where the fo scattering from this atom is:
       cval +  sum_{i=1 to 4} (aval(i)*exp(-bval(i)*(sin(theta)/lambda)**2)
       and the f" value is fprimv and the f"" value is fprprv

    For CLUSTER compounds, you may wish to specify instead fprimv, fprprv,
plus 3 separate formulas, one for the f_o part of the scattering factor,
1 for NORMALIZED f", and one for NORMALIZED f"".
All three of these quantities will depend strongly on scattering angle.

NOTE: when you put in a cluster compound, it is a good idea to look at
the form factors as a function of sin(theta)/lambda using the keyword

plot_formfactors xxxx


The formulas used are:
```

```
                    f_o formula:

The "clus_aval",
"clus_bval","clus_cval(1)" values
will be used almost as for regular atoms, but with 2 additional
factors cval(2) and cval(3) that put in a sinc function as described
in Fu et al., Cell 98, 799 (1999):

clus_aval           4 real numbers (a1,a2,a3,a4)
clus_bval           4 real numbers
clus_cval           3 real numbers

 fo= clus_cval(1) +
 sinc(clus_cval(2)*(sin(theta)/lambda)**clus_cval(3))**2  *
   [sum_{i=1 to 4} (clus_aval(i)*exp(-bval(i)*(sin(theta)/lambda)**2)]

Note the sinc function multiplies the whole sum over the aval/bval terms
but NOT the cval(1) term (it is unclear how this was done
in the Fu et al paper).

                    f" and f"" formulas

clus_fp_aval           4 real numbers (a1,a2,a3,a4) for f"
clus_fp_bval           4 real numbers
clus_fp_cval           3 real numbers

clus_fpp_aval           4 real numbers (a1,a2,a3,a4) for f""
clus_fpp_bval           4 real numbers
clus_fpp_cval           3 real numbers


The formulas for f" and f"" are a little different so as to
preserve compatibility with the definitions for normal atoms.  The
definitions here are not used in the Fu et al article mentioned above.
In each case, the value used for f" and f""
in all the routines is equal to their INPUT values (fprimv and fprprv or
fprimv_mad and fprprv_mad)
TIMES an overall form factor given by (for f"):

 form_factor= clus_fp_cval(1) +
 clus_fp_cval(2)*sinc((sin(theta)/lambda)**clus_fp_cval(3))**2  *
   [sum_{i=1 to 4} (clus_fp_aval(i)*exp(-bval(i)*(sin(theta)/lambda)**2)]

and for f"":

 form_factor= clus_fpp_cval(1) +
 clus_fpp_cval(2)*sinc((sin(theta)/lambda)**clus_fpp_cval(3))**2  *
   [sum_{i=1 to 4} (clus_fpp_aval(i)*exp(-bval(i)*(sin(theta)/lambda)**2)]


Example of CLUSTER form factor input:
```

```
newatomtype wclu

fprimv 100.0    ! fprime value, to be multiplied by clus_fp form factor
fprprv 50.0   ! fprime value, to be multiplied by clus_fpp form factor

clus_aval 2093 5109.4 -1197.1 5254.3   ! form factors for f_o scattering
clus_bval 509.3 -37.8 849.4 108.5
clus_cval 184 30 1.2

clus_fp_aval 0.185886 0.453782 -0.10632 0.466651  ! form factors for fprime
clus_fp_bval 509.3 -37.8 849.4 108.5
clus_fp_cval 184 30 1.2

clus_fpp_aval 0.185886 0.453782 -0.10632 0.466651  !form factors, fdoubleprime
clus_fpp_bval 509.3 -37.8 849.4 108.5
clus_fpp_cval 184 30 1.2


Example of CLUSTER form factor input for mad atom:

Same as above for the newatomtype.  Then specify:
mad_atom wclu

and for each wavelength specify (just as usual)
fprimv_mad 100.
fprprv_mad 50.

Then the clus_fp_aval etc will be applied to the fprimv_mad value, and the
clus_fpp_aval etc will be applied to the fprprv_mad value as a function
of sin(theta)/lambda.

mad_atom  xxxx      name of the anomalously scattering atom is xxxx.

fprimv_mad       f' value for anomalously scattering atom at a particular
                 wavelength (must be input after each wavelength)
fprprv_mad       f" value for a particular wavelength

FIXSCATTFACTORS     Fix scattering factors at their input values [default]
REFSCATTFACTORS     refine scattering factors f' and f".
```

Heavy atom parameters

```
derivative n        begin input of information for derivative/wavelength n
                    This command is used to start entering information on a
                    derivative.  If you want to modify something after you've
                    gone on to another derivative then you need to use the
                    command GOTODERIV
lambda n            identical to derivative n
gotoderiv n         go to previously specified derivative (wavelength) n
                    and get readyto read some modifications of the parameters
                    for this derivative.
```

```
LABEL   text        label for this  wavelength

IEGROUP n           Group of correlated derivatives that this derivative
                    belongs to.  Determined automatically if GETGROUPS is set
DERSCALE        Dividing scale factor applied to all this derivative data
                  after overall scale factor has been applied.  DEFAULT=1.0
DERTEMP         Dividing B-factor to apply to deriv data.  DEFAULT =0.
SIGDERSCALE     Scale factor to apply to derivative sigmas after all above
                  scaling is applied. DEFAULT = 1.0


ATOMNAME XXXX   XXXX is the atom type of an atom to be refined.
                Please note:  the f' and f" values in SOLVE's database are
                for lambda=1.54 A.  If you collected MIR data at a
                synchrotron then you should define a new atom type with
                NEWATOMTYPE and input the correct f' and f" values.
                HINT:  you can get the aval, bval, and cval values for an
                atom recognized by SOLVE by typing madatom [atom name].

                When you type ATOMNAME, SOLVE assumes you are typing in a new
                atom and it zeroes out all the parameters for this new atom.
                If you want to go back to this atom later (i.e., in another
                cycle) use the keywords GOTODERIV and GOTOATOM to identify
                this atom.

                When you have multiple sites for a particular derivative,
                use ATOMNAME XXXX for the first, then input all the data on
                that site, then start the next site with ATOMNAME YYYY, and
                so forth.
gotoatom n      go to the n'th atom in this derivative/wavelength
                and get ready to read some modifications of its parameters


OCCUPANCY x     Fractional occupancy of this atom
                Note: if occupancy is equal to 0.000 or refines to 0.000,
                the atom is ignored in refinement.

BVALUE  b       Temperature factor for this atom.  Anisotropic temperature
                factors are also supported (just input 6 numbers.)
                NOTE: bval is not the same as bvalue.  BVAL refers to
                scattering factors for a newatomtype.

XYZ     x y z   Fractional coordinates of this atom
                Note: if coordinates move dramatically during refinement,
                the occupancy is set to zero and the atom ignored.

PDB_XYZ_IN        PDB file with orthogonal A coordinates
                of all heavy-atoms for this derivative/wavelength
                NOTE: you cannot use XYZ/OCC/BVALUE/REFINEMENT parameters along with
pdb_xyz_in.
```

Control parameters for HEAVY atom refinement and phasing

```
NO_SIM            Do not use Sim weighing with heavy-atom structure factors in
SAD
CORRELPHASE       Use Bayesian correlated MIR phasing (default)
GETGROUPS         Automatically get groups of correlated derivatives for
                  correlated MIR phasing
IMADPHASE    n    wavelength to phase using Bayesian MAD phasing
                  (default =0, no Bayesian MAD phasing). See  description below.
                  n refers to the wavelength to be defined as a reference.
                  Note that you cannot refine with madphase, you can only phase.

Note: Flags for refinement of a heavy atom are cumulative, so you can
refine x and y using REFINEX and REFINEY.

Note: Flags for refinement of a heavy atom do not apply when the keyword
SOLVE is used (only with HEAVY)

REFINENONE      Don't refine anything...reset all the refinement flags to zero
REFINEALL       Refine x,y,z,occupancy, and B
REFINEOCCB      Refine occupancy and B
REFINEXYZ       Refine x,y,z
REFINEX         Refine x
REFINEY         Refine y
REFINEZ         Refine z
REFINEOCC       Refine occupancy
REFINEB         Refine B


NOREFINESCALE   Do not refine overall scale factor. Default = refined
REFINETEMP      Refine B-factor applied to deriv data. DEFAULT= not refined

nsolsite_deriv      Maximum number of sites for this derivative only
cutoff_deriv 200. 3.5       resolution limits for this derivative/wavelength only
RES_PHASE 2.2   high-resolution limit for phasing only
SN_MIN    Set RES_PHASE so that signal-to-noise is bigger than this
SN_RATIO_MIN    Set RES_PHASE so that signal-to-noise is bigger than this
                    ratio times the value at low-resolution
INPHASE             include this wavelength/derivative in phasing.
NOINPHASE           do not this include this derivative/wavelength in phasing
INANO               include anomalous differences for current wavelength/deriv

noanorefine        use anomalous differences in phasing but not
                   refinement for this derivative.
                   (this is usually the best option for MIR
                   unless your anomalous differences are really
                   big, as from a synchrotron MIR dataset at an
                   absorption edge).  Note: you still have to specify
                   for each derivative "inano" to include anomalous
                   differences for that derivative.

anorefine          For this derivative with "inano" specified, use
```

```
                    anomalous differences in both refinement and phasing.
                    This is best for MAD data. (This is the default also).
                    Applies to current derivative/wavelength

ISOONLY             use only isomorphous differences in phasing and
                    refinement this deriv
ANOONLY             use only anomalous differences in phasing and refinement
                    this deriv
ISOANO              use both isomorphous and anomalous differences this deriv
                    (cancels ISOONLY and ANOONLY)(default)

KOUT     n      type of output from HEAVY if any. DEFAULT = 0 (no binary output)

                KOUT...TYPE OF OUTPUT
        0.... NONE
        2.... DIFFERENCE FOURIER FOR KDER
           A=m(Fder-Fnat)cos(PhiBest)
           B=m(Fder-fnat)sin(PhiBest)
        3.... ANOM DIFF FOURIER FOR KDER
           A=m(DelAno)cos(PhiBest-90)
           B=m(DelAno)sin(PhiBest-90)
        4.... RESIDUAL MAP FOR KDER
           A=m(Fder-|Fnat+FH|)cos(PhiBest)
           B=m(fder-|Fnat+FH|)sin(PhiBest)
            (where Fnat+FH is the vector sum of Fnat
              and the heavy atom FH)
        6.... NATIVE FOURIER
           A=m(Fnat)cos(PhiBest)
           B=m(Fnat)sin(PhiBest)
        7.... PHASES AND FIGURE OF MERIT and fnat
              PhiBest (in degrees), PhiMostProbable,
                and figure of merit, and Fnat
        8.... Fnat,phibest, phi most probable, figure of
              merit, HENDRICKSON-LATTMAN COEFFS
        9.... HEAVY ATOM S. FACTORS FOR KDER
           A, B= real and imaginary parts of normal scattering from heavy atom.
           C, D= real and imaginary parts of anomalous scattering
         NOTE: m=the figure of merit, PhiBest is the "Best" phase,
          PhiMostProbable is the the most probable phase.

KDER    n       derivative n is to be included in output

IANGLE   x      phasing angle, minimum=5, default=5
INANAL   n      PHASE ANALYSIS.  DEFAULT=0
                  1 for printing of extensive heavy atom statistics

INRESD   n      RESIDUAL AND STATISTICS.  DEFAULT = 0
                -1  No residuals or statistics calculated.
                0   zeroth cycle added before first refinement
                    cycle. During zeroth cycle residuals and
                    statistics are calculated and printed.
                    No statistics are calculated on other cycles.
```

```
                1   Residuals and statistics calculated every
                    cycle and printed according to INPRNT.
                    Note: residuals are only calculated for
                    derivatives with INPHAS = 1.
INOSIG  n       USE OF SIGMAS.  DEFAULT = 0 (use sigmas).
                 1  if sigmas from input data file are not to be used.
INHEND  n       USE OF HENDRICKSON-LATTMAN COEFFICIENTS.  DEFAULT=0 (don't use)
                 1  if Hendrickson-Lattman coefficients are to be calculated and
                    used for phasing instead of Blow-Crick phasing.
                    HEAVY does not do phase combination.  If KOUT=8 and
                    INHEND=1, then standard H-L ABCD are output.  If KOUT=8 and
                    INHEND=0, then HL ABCD are fit by least squares to the
                    phase probability distribution.
INPRNT   n      PRINTING OF SHIFTS. DEFAULT =0 (don't print)
                 1  if shifts (and statistics, if any) are
                    to be printed on every cycle.  Default
                    is to print statistics on first cycle,
                    shifts and statistics on last.
JALT     n      USE OF PHASES IN REFINEMENT.  DEFAULT = 0 (Patterson refinement)
                   0 is to use origin-removed Patterson refinement.
                   1 is to use phase refinement at most probable phase
                   JALT and KALT are set automatically to 0
                   if you use a procedure (IHEAVYPROC > 0)

KALT     n      USE OF DERIVATIVE BEING REFINED IN PHASING. DEFAULT=0(don't use)
                   0 is not to use derivative being refined in phases
                   1 is to use all available derivatives in phasing
                   JALT and KALT are set automatically to 0
                   if you use a procedure (IHEAVYPROC > 0)

NCYCLE    n     Number of cycles of refinement to be carried out if a
                PROCEDURE is NOT used (see IHEAVYPROC).  Maximum = 30
                Default = 0
IREFCY    n     List of derivative numbers to be refined during the NCYCLE
                cycles of refinement if a procedure is NOT used. Default = 0
                  i.e., 1,1,1,1,0 means refine deriv #1 on cycles 1-4 and
                  calculate phases, get residuals, figure of merit, etc
                  on cycle #5.  Note that you don't get these statistics on
                  cycles in which you refine with Patterson refinement.

IHEAVYPROC  n   RUN a procedure with HEAVY.  Default =5 (refine everything)
                Available procedures:
                1 = NREP cycles of refinement of each deriv that has INPHASE
                  specified, refining only occupancy.
                2 = as 1, but refining only xyz.  Fixes coordinates of best
                  atom in each deriv in polar space group in polar directions
                  unless another atom is already fixed by user.
                3 = as 2, but refining xyz and occ
                4 = as 3, but refining xyz, occ, B
                5 = 1, then 2, then 3, then 4
                6 = phased refinement to obtain relative coordinates among
                  derivatives for polar directions in polar space groups.
```

```
                  Fix and phase with best derivative.  Refine just coordinates
                  in polar directions for all other derivatives with INPHASE
                  specified.  This should be followed by #5 again.
                Note: when you use this procedure you must still set which
                refinement flags you want to (ever) be refined.  The
                procedures only turn OFF flags, they do not turn on ones
                you have never set.

NREP    n       # of refinements of each deriv in procedures with
                 IHEAVYPROC > 0 (default=5)

SMALL   xx       minimum ratio of derivative structure factor amplitude
                 (F Deriv) to RMS lack-of-closure for use in
                 refinement or residuals. DEFAULT=0.
FMIN    xx      minimum native F for any action. Default=0.
FOMMIN  xx      minimum figure of merit for use in phased
                   portion of refinement. Default =0.
BMAX              maximum allowed bvalue default = 60
BMIN              minimum allowed bvalue default = 1


THR     xx      Keywords to set threshold and damping factors for shifts:
ACL     xx      if SHIFT > THR *sigma of SHIFT, SHIFT=SHIFT*ACL
                         Defaults are 0. and 0.5

FSIGMIN  xx     MINIMUM ratio of F/sig to include. DEFAULT =1.0

NNATF     n     column number in input file for native F
NNATS     n     column number for sigma of native F

NNATF_DERIV n   column number for native F that pairs with this derivative
NNATS_DERIV n   sigma of native F that pairs with this derivative

NBST     n      optional column number for "best" phase in input file
NMP      n      optional column number for most probable phase in input file
NFIGM    n      optional column number for figure of merit in input file
                  Note: if you set NFIGM for routine HEAVY it will be
                  applied in MAPS too.  Be sure you reset it to the value you
                  want.

INOLD    n      Flag for using phases from input file in phasing when they
                  are not available from current data. default = 0.  To use
                  input phases, inold=1

ANATSCALE xx   Overall scale factor applied to ALL data before any
                  other scaling.  DEFAULT = 1.0
SIGNATSCALE  xx   Scale factor applied to native sigmas after overall scaling
                  DEFAULT = 1.0
EIS             Optional list of estimated rms isomorphous lack-of-closure
                  residuals in 8 resolution ranges
EAD             Optional list of estimated rms anomalous lack-of-closure
                  residuals in 8 resolution ranges
```

```
FPHBAR          Optional list of estimated rms derivative F in 8
                  resolution ranges
FHBAR           Optional list of estimated rms heavy atom F in 8
                  resolution ranges
SIGBAR          Optional list of estimated rms derivative sigma in 8
                  resolution ranges

USE_F_BINS      Use |F| and resolution to set bins for phasing statistics

                 (only used internally, does not affect printout)

NO_USE_F_BINS      Use only resolution to set bins for phasing statistics

N_F_BINS        Number of bins (for each resolution range) based on F for

                phasing
```

Control parameters for HASSP

```
searchregion            region to search (xs,xe,ys,ye,zs,ze) (default = region
                            covered by the FFTGRID)

ihassptype n            control of what is to be done (default=0)
                        0    search for single-site solutions, then 2-sites
                        2    search for single-site solutions
                        5    same as 0
                       -5    use trial solutions given in keyword trialsite
                              as trial cross-vectors
                       -6    use trial sites given in trialsite as a
                              trial solution. evaluate it and look for more
                              sites

discrm  xx              ratio of peak height over surroundings to use
                          (default=1.0)
icrmax     n            maximum # of peaks to try in 2-site search (default=10)
nospec     n            control over ignoring symmetry #'s of special positions
                          (default=0, do not ignore)
nsignf     n            0 if significance of peaks is to be tested (default=0)
spat       xx            minimum probability for non-randomness to keep a peak
                          in routine "patpk". (default=0.0)
ssin       xx           as spat, but for single-site searches (default=0.0)
sdub       xx           as spat, for two-site searches (default=0.95)
strp       xx           as spat, for 3-site searches (default=0.0)
ssft       xx           as spat, for sifting through 3-site solutions
                         (default=0.95)

ihassplist              print out value of patterson function
                          at every predicted self- and cross-vector for every
                          solution.  this is very helpful for analyzing a
                          solution in detail. default =0 (not to print out).
```

```
trialsite   x y z       fractional coordinates of a trial site or cross vector
                          (read in if ihassptype < 0)

ntophassp       n       a maximum of n solutions will be saved in routines
                        that use hassp output to build up solutions

sigma_remove       n       peaks labelled with uvw_remove will be set to this value
uvw_remove u v w       n      remove this uvw peak from the Patterson (set to
sigma_remove*sigma)
```

Working with maps

```
PATTERSON            this is a Patterson map
FOURIER              this is a fourier map
```

Grids used for FFT calculations

```
FFTGRID  xs xe xtotal ys ye ytotal zs ze ztotal    grid for FFT calculations
PATTGRID xs xe xtotal ys ye ytotal zs ze ztotal    grid for Patterson
EZDGRID  xs xe    ys ye    zs ze                   grid for NEWEZD map
BOSSGRID  xs xe    ys ye    zs ze                  grid for MAPVIEW map .
```

Control parameters for GENERATE

```
coordinatefile          pdb file with coordinates for GENERATE
percent_error           % error added to intensities in GENERATE
cell_derivative a b c alpha beta gamma  (only for generate_mir) cell parameters for
this derivative.
```

Contents	Index

How SOLVE works

MAD structure determination and overall SOLVE operation

The real power of the SOLVE package is the automated solution of MAD and MIR datasets. For a MAD dataset, all you need to do is tell the program about your space group and cell dimensions, where the unscaled intensity data files are, and what the scattering factors are for the MAD dataset. The program scales the data using localscaling in SCALE_MAD , calculates difference Patterson maps, compresses the MAD data into SIRAS-like data using MADMRG, calculates an optimized Bayesian heavy-atom Patterson using MADBST, and iteratively builds up and scores potential heavy-atom solutions for the MAD dataset with the SOLVE routine. Potential solutions are scored on the basis of (1) agreement with the Patterson, (2) "free" difference Fouriers, (3) the presence of "solvent" and "protein" regions in a native electron density map, and (4) the figure of merit of phasing.

SOLVE can either look exhaustively for solutions (trying all possible additions/deletions/inversions etc) or else just follow the best solution and keep adding on to it. The default is to follow the best solution (keyword is "mediumquick"), and SOLVE keeps looking at seeds until this process leads to a result with a figure of merit > 0.5 and a Z-score over 10. If you set the "veryquick" keyword, SOLVE will just look at one seed this way.

Final phases for the top solutions are written out along with Hendrickson-Lattman coefficients for calculating maps and solvent flattening. SOLVE refines scattering factors for the MAD data after heavy-atom parameters are found by comparing the effective occupancies of sites when refined against each possible set of dispersive or anomalous differences. SOLVE calculates final phases using Bayesian correlated MAD phasing.

SAD structure determination

SAD data is single-wavelength data with anomalous differences. SOLVE treats SAD data as a native (the averaged F+ and F-) and anomalously-scattering derivative (F+, F-) where the f' value is zero and the f" value is non-zero. The heavy-atom sites are found using the anomalous differences and phasing uses just the anomalous differences. Scattering factors are not refined. SAD phases require solvent flattening or other density modification before they are useful. RESOLVE works very well for SAD data.

MIR structure determination

MIR structure determination is almost the same as for MAD structure determination. The program scales the data using localscaling in SCALE_NATIVE and SCALE_MIR , calculates difference Patterson maps, and iteratively builds up and scores potential heavy-atom solutions for the MAD dataset with the SOLVE routine.

Combinations of multiple MAD or MIR datasets

SOLVE can treat multiple MAD or MIR datasets by converting them into one super-dataset that uses each native dataset other than the first one as a pseudo- derivative with no heavy atom. Then it uses correlated bayesian phasing to take into account any non-isomorphisms among the different datasets. In this way you can combine several MAD datasets, several MIR datasets, or MAD and MIR datasets. See the command COMBINE.

Datasets with anomalously-scattering native

SOLVE treats native datasets with anomalously-scattering atoms as two datasets. You enter the same data twice, once as a "native", and again as a "derivative" for which you specify a new heavy atom type with a zero real part and a non-zero imaginary part.

Contents	Index

Data formats for automated structure determination with SOLVE

Your choices about input data include:

- Premerged or unmerged data?
- One data file or more per dataset?
- What data format? Intensities or amplitudes? (Scalepack, formatted, CCP4 mtz, d*trek)
- An additional file with phases (i.e. from MR)?

These choices are discussed below in more detail. See also the SAMPLE SCRIPTS.

Should you merge your data to the asymmetric unit before running SOLVE?

- SOLVE can read unmerged data or data merged to the asymmetric unit.
  - PREMERGED data is best if your data is already well scaled
  - UNMERGED data is best if your data has not been thoroughly scaled already

Can you input more than one data file for a native, derivative, or wavelength?

- For each native, derivative, or wavelength dataset, you can input one or more separate data files.
  - If a dataset has just one data file, just read in the datafile
  - If a dataset consists of several data files, just read them in one after another

What data format? Amplitudes or intensities?

- if you have DENZO/SCALEPACK output as your raw data...
  - ...and the data is NOT MERGED to the asymmetric unit, you will use the flags:
    - READDENZO
    - UNMERGED
    - READ_INTENSITIES
  - if the data is ALREADY MERGED to the asymmetric unit, substitute the flag:
    - PREMERGED

- if you have FREE-FORMAT intensities or amplitudes as your raw data...
  - ...and the data looks like: H K L I SIGMA, use the flags
    - READFORMATTED
    - UNMERGED

        - READ_INTENSITIES
    - if the data looks like: H K L I+ SIGMA+ I- SIGMA-, substitute the flag:
        - PREMERGED
    - if you have free-format F(hkl) instead of intensities:
        - substitute the flag READ_AMPLITUDES

- if you have a CCP4 MTZ file with amplitudes scaled and reduced to the asymmetric unit as your raw data...
    - You will have to make sure that this mtz file contains only the data you want and not lots of other columns of data
    - Note what you have called your data columns
    - The column names that SOLVE will want assigned are:
        - MAD data:
            - FPH1 (amplitude at wavelength 1)
            - SIGFPH1 (sigma of FPH1)
            - DPH1 (anomalous difference wavelength 1)
            - SIGDPH1 (sigma of DPH1)
            - FPH2 (etc for wavelength 2, 3 ...)
        - MIR data:
            - FP (amplitude for native)
            - SIGFP (sigma of FP)
            - FPH1 (amplitude for deriv 1)
            - SIGFPH1 (sigma of FPH1)
            - DPH1 (anomalous difference deriv 1)
            - SIGDPH1 (sigma of DPH1)
            - FPH2 (etc for derivs 2, 3 ...)
    - use the flags LABIN and HKLIN to tell SOLVE how to read your mtz file. You can use multiple LABIN statements if you can't fit it all on one line. A sample LABIN statement where native F is called FP and sigma is SIG and deriv F is called FHG and sig of deriv F is SIGHG and anom diff for deriv is called DELHG and its sigma is SIGDELHG and with an input file of input.mtz is:
        - LABIN FP=FP SIGFP=SIG FPH1=FHG SIGFPH1=SIGHG
        - LABIN DPH1=DELHG SIGDPH1=SIGDELHG
        - HKLIN input.mtz
        - NOTE: use uppercase letters (unless your column names are lowercase) because case matters here
    - SOLVE figures out if this is MIR or MAD data based on whether or not you define FP and SIGFP.
    - When SOLVE reads the HKLIN statement it will read in the file using the information in all previous LABIN statements. HKLIN can be specified only once in a solve run.
    - You do not need to input cell dimensions or space group if you use HKLIN. The values read from the mtz file are used unless you change them with a keyword after the HKLIN statement. SOLVE writes out a symmetry file in the local directory based on the symmetry

information in the mtz file that you can use later if you wish. It is named with the space group name.
    - NOTE: remove the SCALE_MAD command from your script file as your data is assumed to be scaled already

- if you have a set of CCP4 MTZ files with unmerged intensities (LABIN I=I SIGI=SIGI)use the flag:
    - READCCP4_UNMERGED !(instead of readdenzo or readformatted or readtrek)
    - Enter data file names just as for readdenzo or premerged
    - You may not specify a LABIN line with this option. Your mtz file must contain I and SIGI as the column labels.

- if you have a d*TREK file with intensities as your raw data...
    - use the flag READTREK (just one flag needed)

What if I have phases from molecular replacement?

- If you have an "mtz" file containing FC PHIC FOM then specify (myFC is your column name for FC, etc):
    - PHASES_LABIN FC=myFC PHIC=myPHIC FOM=myFOM
    - PHASES_MTZ xxxx.mtz
- If you have a formatted file with H K L FC PHIC FOM (one record per line; there can be text in between the numbers, such as in CNS or X-PLOR formatted files), then specify:
    - PHASES_FORMATTED xxxxx.fmt
- That's it. Put these lines somewhere in your input file before "SOLVE" and SOLVE will read in these phases and use them in initial difference Fouriers to find sites.

Contents	Index

Introduction to SOLVE

What SOLVE does

SOLVE is a program that can carry out all the steps of structure determination for MAD and MIR, starting with raw intensities (e.g., Scalepack output), and ending with phases and a NEWEZD electron density map that you can read right into "O". SOLVE can make all the decisions in this process and come up with a map all by itself if you want. On the other hand, if you know something about your system that SOLVE doesn't seem to realize, you can guide it through the process by editing script files that SOLVE writes out and continuing in the fashion that you choose.

To use SOLVE on your data, first read the rest of this introduction and the Getting Started section that follows. Then go right on to Automated structure determination to solve your structure.

To get an idea of what SOLVE does, you might want to have a look at some examples of MAD and MIR structure solutions using SOLVE.

Then you might want to try SOLVE out on some test data in your space group using the GENERATE feature. This allows you to create a MAD or MIR dataset with any heavy atom sites you want and then run SOLVE on it. If you start with a PDB coordinate file, you can generate a dataset, solve it, and use "O" to display the EZD electron density map that SOLVE creates along with the correct structure.

Getting Started

[Scripts | Useful commands | Symmetry files | Data files]

[Your license |SOLVEDIR, SOLVETMPDIR, CCP4_OPEN environmental variables| More help]

Scripts

This manual is set up to provide you with short scripts that you can edit to carry out MIR and MAD structure determinations and to carry out any of the other functions of SOLVE.

If you want, you can put the standard information for your dataset in to a file "solve.setup":

```
!------------------ solve.setup: standard setup for a dataset --------------
SYMFILE /usr/local/lib/solve/c2.sym     ! symmetry file for this space group
                                        !  (most common space groups should be in
                                        !   this directory)
CELL  76.08 27.97 42.08 90 103.2 90     ! a, b, c, alpha, beta, gamma
RESOLUTION 3 20                         ! Resolution limits in A.
!-------------------------------------------------------------------------------
```

Now each time you start up SOLVE, just type:@solve.setup and the program will read in this information. You can put any keywords or commands into script files that you can type at the terminal. You can also put this information right into your scripts for running SOLVE if you prefer.

Useful commands and information about running SOLVE

The "solve.status" file

SOLVE continuously updates a file called "solve.status" in your working directory. The tail end of the file shows the current status of your SOLVE job. You may find it convenient in unix to open a window, get in this directory, and type "tail -30f solve.status". This will continuously update your screen with the current status of SOLVE. In general, good datasets take a lot less time than poor ones because SOLVE recognizes a correct partial solution very early in a good dataset and can just follow it until all the sites are found.

The "solve.control" file

SOLVE will read a file called "solve.control" in your working directory while it is running. This file can be used to tell SOLVE to finish up as soon as practical if you think it has found the best solution already and you don't want it to keep looking any longer. Once SOLVE has read the file it writes the keyword IGNORE at the end of the file. This tells it not to read the file again (until you delete or comment out the keyword IGNORE). The most useful commands that can be read from this file are:

```
!-------- solve.control: control solve operation in real time--------------
FINISH          ! finish up cleanly as soon as possible (do not look for any new
solutions)
NTOPSOLVE  n    ! only consider n solutions at a time
NSEEDSOLVE  n   ! only use n seeds as starting points for SOLVE
NTOPDERIV n     ! only try out up to n peaks from a difference Fourier when adding
                            ! on new sites to a solution
```

Useful commands

Some commands that you might find useful right away are "Help", "?" "@" and "History".

HELP will print out a list of available commands.

? [keyword or command]

?ALL will print out a list of all keywords applying to each command and the keyword values.

? followed by a command will print out a list of all keywords that apply to that command and their values. Example: "?maps" will list all keywords that apply to "MAPS".

? followed by a keyword will print out the value of that keyword. Example: "?cell" will print out the current cell constants a, b, c, alpha, beta, gamma. This is useful for making sure that the values are what you think they are and for finding out

what the defaults are.

@ filename

The "@" symbol is used to read in the contents of a file as commands. In the above example, "filename" will be opened and interpreted as keywords and commands just as if they were entered from the terminal.

HISTORY

This command will print out a list of the most recent commands and keywords input. The number listed is controlled by the keyword NLIST. Example:

```
nlist 100
history
```

will print out the 100 most recent commands and keywords.

Abbreviating commands

You can abbreviate commands as long as what you type is unique. That is, you can type "hist" and SOLVE will recognize this as "history". Any text appearing after a "!" or a "[" on a keyword or command line is ignored. This lets you enter comments in your script files. SOLVE reads commands one line at a time. This means any information for a keyword has to appear on the same line as the keyword. Also you can only enter one keyword per line. You can end the program by typing "stop".

Alias

If you don't like a command name SOLVE uses, you can create an alias of your choice. For example, if you tell SOLVE:

```
alias quit stop
```

then any time you type "quit", SOLVE will interpret this as "stop" (and it will stop). If you want to know what the current aliases are, type,

```
?alias
```

and SOLVE will list them for you.

Running more than one SOLVE job at a time.

You can run more than one SOLVE job on your computer at one time, but they must be in separate directories. This is because SOLVE writes out lots of temporary files with standard names and they would be mixed up if two jobs were running simultaneously in one directory.

Symmetry files

You need to tell SOLVE about your space group symmetry with a symmetry file using the information in the International Tables. The directory /usr/local/lib/solve (or else the directory named if you type "echo $SOLVEDIR") on

your computer should contain the most common space groups as they are supplied with the installation of SOLVE.

Here is the symmetry file for space group C2 which you would find in "/usr/local/lib/solve/c2.sym":

```
4          !  Line 1 of the file is the number of symmetry elements to follow
x,y,z
-x,y,-z
x+1/2,y+1/2,z
-x+1/2,y+1/2,-z
```

Enter the identity "x,y,z" as the first operation always. For centered cells, enter the members of the centered sets in the same order for each centering operation.

NOTE: dashes (-) and underscores (_) in the space group file name have meanings. Don't confuse r3.sym with r-3.sym!

For space groups like P2/m that contain slashes, the space group file name has an underscore instead of the slash (p2_m.sym is P2/m). Similarly, for space groups like P1-bar, the file name for the space group has a dash (p-1.sym is p1-bar)

Data Files

The data files used by SOLVE are binary. To look at one, you can use the keywords "INFILE", "NLIST", and "VIEW". Here is a script to look at the file "data.drg":

```
!-----------------------------------------------------------------------------
! script file to VIEW a .drg file
@solve.setup
INFILE data.drg
NLIST 20
VIEW
!-----------------------------------------------------------------------------
```

If you want to convert SOLVE datafiles to another format, then you can use the IMPORT and EXPORT commands. See the "Importing and Exporting data" section for details on this and for details on converting binary maps to other formats.

If you want to look at all occurrences of the reflection "-1 3 6" and its symmetry equivalents, you can use the keyword "HKL_VIEW" with "NLIST -1":

```
!-----------------------------------------------------------------------------
! script file to VIEW a reflection and sym equivalents in  .drg file
@solve.setup
INFILE data.drg
NLIST -1
HKL -1 3 6
VIEW
!-----------------------------------------------------------------------------
```

For automated analysis of MAD and MIR data, you can start with Scalepack output or other formatted data. If you used mosflm and have an .mtz file with your data, you can read that right into SOLVE as well.

For other operations of SOLVE, you will need to IMPORT your data into "dorgbn-style" files. See the description of "IMPORT" later in this documentation. Missing data is represented by either "-1.0" or "0.0" in this package.

Your license

Users of SOLVE are issued an "solve2.access" license file that looks like:

```
License for SOLVE expiring 3-jan-1999
AX9481VA991
```

You should name your license file "solve2.access" and put it in the directory named in the environmental variable "SOLVEDIR" (see below). That is, if you type "echo $SOLVEDIR" and your computer responds "/usr/local/lib/solve" then your license file should be located in "/usr/local/lib/solve/solve2.access".

The SOLVEDIR, SOLVETMPDIR and CCP4_OPEN environmental variables

SOLVE expects to find the file "solve2.access" and your symmetry files (e.g., c2.sym) in your working directory or else in the directory named by the environmental variable "SOLVEDIR". If you put these files in the directory "/usr/local/lib/solve" then you should add a line something like the following to your .cshrc or .cshrc_custom login file (or the corresponding files if you use another shell):

```
setenv SOLVEDIR /usr/local/lib/solve
```

You should then be able to check that the variable is set by typing

```
echo $SOLVEDIR
```

and your computer should respond

```
/usr/local/lib/solve
```

By default, SOLVE writes temporary files in your working directory. If you are running on one machine and your directories are on another, this can slow SOLVE down. SOLVE now allows you to set a SOLVETMPDIR environmental variable that specifies where these files are written. You can set it to a directory on your local machine to speed up SOLVE. (If the directory does not exist, solve will stop.)

```
setenv SOLVETMPDIR /var/tmp
```

SOLVE now uses ccp4 library routines. To allow overwriting of existing files, you need to set the CCP4_OPEN environmental variable:

```
setenv CCP4_OPEN UNKNOWN
```

Getting more help

You can get more help with this program and report bugs by emailing the SOLVE/Resolve mailing list. You can also email the author at "terwilliger@LANL.gov". Before you do this, however, please check that you have the most recent version of the program and documentation as your bug may already have been corrected.

Contents	Index

Answers to SOLVE Frequently-asked questions

Where can I find all the questions and answers that have been sent to the SOLVE/RESOLVE newsgroup?" ?
You can find them archived at Jiansheng Jiang's very nice site at http://asdp.bnl.gov. On that site, click the blue button "BBnML" (Bulletin Boards and Mailing Lists) and then click "solve" under the Archive column.

Where is SYMINFO?
SOLVE/RESOLVE versions 2.08 and higher use the CCP4 version 5.0 libraries. These require both SYMOP and SYMINFO to be defined. (They are both symmetry libraries). They can be defined (if you are using csh, for example) with:

```
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
```

What should I try if SOLVE cannot solve my MAD dataset?
Try the SAD script right away if using your MAD data fails. In many cases, the data for the wavelengths collected later are severely affected by crystal decay, and in these cases using just the first wavelength collected (hopefully at the peak wavelength) may work much better than using all the data.

What is the best way to tell if my data are good?
For MAD data , have a look at the correlation of anomalous differences. See the table in the beta-catenin dataset for example. For MIR data, check that your R-factors between derivative and native start out large at low resolution, get smaller, and then finally get bigger again (the last rise is due to the errors in measurement and indicate where to cut off).

For SOLVE version 2.01 and higher, you can look at the analysis of signal-to-noise in the data listed in solve.prt. SOLVE estimates the noise from the sigmas in the data and the signal from MAD anomalous or dispersive differences, and from SAD anomalous differences. Note: If the sigmas in the data are clearly overestimated (rms(sigma) > rms (difference) then SOLVE rescales all the sigmas so as to yield rms(sigma)=rms(difference) in the highest resolution shell.

Can I input sites that I already know into SOLVE? Yes, you can. You just put them right in under the correct wavelength or derivative, add the keywords "addsolve" or "analyze_solve" before the scaling command, and SOLVE will use them to find new sites (addsolve) or to refine and calculate phases (analyze_solve). See addsolve and analyze_solve instructions for examples.

Can I use a solution at low resolution to run SOLVE at high resolution? Yes, you can. The easiest way is with addsolve and analyze_solve.

What do "checksolve" and "comparisonfile" do? Checksolve tells SOLVE to compare all the solutions it gets with the one that you input. SOLVE finds the origin (and hand, if you do not have anomalous data) that best matches its trial solutions with the one you entered, and reports the solution relative to this origin and hand. Comparisonfile allows you to input an FFT that SOLVE has previously calculated (at the same resolution as SOLVE is working); in combination with checksolve, SOLVE will calculate the correlation coefficient of every map that it examines to the one you input. This is handy when you have used "generate" to create a dataset.

Will SOLVE give me the right hand for my structure? Usually if you have good anomalous differences and MAD data, then yes, SOLVE will give you the correct hand. For SAD data, it is common for SOLVE not to get the correct hand. Run RESOLVE or the RESOLVE_BUILD script. after SOLVE, and if it fails to build anything useful, rerun SOLVE with analyze_solve and use the heavy-atom sites located in "solve_inverse.xyz" which are just the inverse of the sites in your original SOLVE run. In very rare cases your anomalous differences might be reversed (due to incorrect analysis of data or detector hooked up backwards). In that case you can use "swap_ano" to reverse the signs of the differences.

How do I get a bigger version of SOLVE? The distribution comes with the regular sized SOLVE and solve_giant and solve_huge. Try these first. If you need even a bigger version, then email me at terwilliger@lanl.gov and I'll give you the source so you can compile a bigger version. You will need the CCP4 library file libccp4.a to compile SOLVE.

Do I need a new access file for a new version of SOLVE? No, the same access file is good for all versions from version 2.0 through 2.99. If you are upgrading from version 1 to version 2, yes you do need a new access file (and the new one goes in "solve2.access").

Why will SOLVE read my solve2.access file but not RESOLVE ? This is a bug triggered by not having a carriage return after the access code on the second line of solve2.access. Just put in the carriage return and it should work for RESOLVE too. Sorry!

Why do I have to set BMIN=0 for high-resolution SAD data or other high-resolution data? You just need to do this for SOLVE versions earlier than 2.03 (20-Sept-2002). The reason is that the default minimum B value for heavy-atom site is B=15. For high-resolution data, typically B values are 5-10 so this default is much too high. If you set BMIN=0 then SOLVE can properly refine these B values. For version 2.03 and higher, the default is 2.

Where do I get f' and f" scattering factors? The best place to get f' and f" values for your MAD experiment is from the beamline staff where you collected your data. They will usually have made careful measurements of these for standard settings on their beamline, so if you do a Se experiment, for example, their values should be very good. You can also measure X-ray fluorescence from your own crystal and use the Kramers-Kronig transformation to estimate these values with the same programs the beamline staff used for their standard cases.

SOLVE does use the f' and f" values and they are very important. The wavelength values are not used in any important ways by SOLVE

Where do I get scattering factors for atoms that SOLVE has not heard of? They are on pp. 500-501 of Volume C of the international tables. For example (Nb:)

```
NEWATOMTYPE NB
AVAL 17.6142 12.0144 4.04183 3.53346
BVAL  1.18865 11.7660 0.204785 69.7957
CVAL 3.75591
FPRIMV -.248
FPRPRV 2.48
```

Why are the figures of merit in the solve.status file not quite the same as the final values? The reason that the final phases look better for MAD data than the ones reported in the solve.status file is that SOLVE calculates phases at the very end using bayesian correlated mad phasing, which gives much better phases than the SIRAS-like phases used during the main part of the run (when the solve.status file is being written). The reason the full phasing is not used all the time is that it is very slow.

Should I use all my data, or just the good data? Though it would be nice to use all the data, it is far better to use just the good data. Unless your sigmas are perfect and the statistics were done perfectly, it is really hard to get rid of the interference caused by data containing noise and essentially no signal.

Will SOLVE use NCS? Regrettably, no.

Why should I use NO MERGE ORIGINAL INDEX in scalepack? You should use "no merge original index" in scalepack so that SOLVE can re-scale the data with local scaling. This flag tells scalepack to write out the place in reciprocal space that each reflection was measured. Then SOLVE can compare it to its neighbors in reciprocal space.

Can I compare Z-scores for SOLVE runs in different space groups? At different resolutions? No, Z-scores are relative and therefore cannot be compared for different space groups or resolutions.

Can I read in data in 2 different formats? Unfortunately not.

Can I convert solve files like mad_fbar.scl into mtz files? Yes, you can. You will need to use "export" to export the data to a flat file, then use the ccp4 routine f2mtz to import into mtz.

Can I look at my patterson maps? Yes, you can. SOLVE writes some of them out as ".ezd" files which you can read right into "O" or convert to anything else with "mapman". Others you can convert to ezd with "fft toezd".

Why do I get an execution error with no output when I try to run SOLVE? On an SGI, if you run a version of solve that does not match your computer, you get an "exec error". Try a version of solve for a lower version of your machine (i.e., r5000 instead of r12000).

Why does SOLVE say CELL DIMENSION <1 OR > 1000 FOUND? This happens if you try to use a really huge unit cell that SOLVE didn't expect. You'll have to cut back on the resolution a bit if it happens.

Why does SOLVE say /sbin/loader: Fatal Error: set_program_attributes failed to set heap start? This is an error that your Compaq Alpha might give you if you don't have enough memory allocated to you. The solution is to add a line to your .cshrc file that just says: "unlimit". This tells the system to give you all available resources.

Why doesn't COMBINE_ALL work for me? For combine_all to work, you have to be sure and input two or more complete datasets, separated by "new_dataset".

Why can't I use COMBINE_ALL_DATA with mtz files? SOLVE won't let you read in more than one mtz file, unfortunately. Sorry about this limitation. This means you can't use COMBINE with mtz files. You would need to dump your mtz data into flat files and read it in with "readformatted" instead.

Why can't SOLVE find 2 sites that are close together? SOLVE won't let you find sites that are closer than a specified number of grid units. The distance depends on the grid size, which is typically 1/3 the resolution. The default ("ntol_site") is 8, or about 2 to 3x the resolution. You can decrease it if you want; in which case SOLVE will have to consider more solutions and may have trouble identifying the best.

What does it mean when SOLVE says "error in reading this file" when reading a scalepack file ? When SOLVE encounters "********" in a data file it will give you this error message. In scalepack (.sca) files this occurs if there is a reflection with a very large intensity that does not fit in the format of the file. One solution is just to edit the .sca file to remove these lines. Another is to re-run scalepack, specifying a scale factor to apply to all the intensities. You can do this with the keyword:

```
scale factor 10.0
```

in Scalepack.

On-line manual for RESOLVE

SOLVE/RESOLVE home page	SOLVE Table of Contents

RESOLVE Table of Contents

Quick Start

- ***How to use RESOLVE and how to run RESOLVE right after SOLVE***
- ***How to use RESOLVE to remove model bias from a map***

Introduction to RESOLVE

- ***Why another density modification approach?***
- ***Problems with the phase recombination approach to density modification.***
- ***A statistical approach to density modification***
- ***Using all the available information for density modification***
- ***Removing model bias with prime-and-switch phasing***
- ***Using NCS in RESOLVE***
- ***Automated model-building and iterative model-building with RESOLVE***
- ***Automated fitting of flexible ligands to electron density maps***

- ***Merging of NCS-related models***

Running RESOLVE

- ***Summary***
- ***Input and output mtz files***
- ***Keywords***
- ***Sample scripts***

RESOLVE output

References for RESOLVE

- ***Reciprocal-space solvent flattening***
- ***Statitstical density modification (previously called maximum-likelihood density modification)***
- ***Pattern recognition (finding helices and strands)***
- ***Map-probability phasing (prime-and-switch)***

RESOLVE example maps

- ***Removal of model bias with prime-and-switch phasing***
- ***Improvement of experimentally-phased maps***

RESOLVE FAQ page

- ***What do I do if RESOLVE says..***
- ***Can RESOLVE..***

electronic reprint

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Maximum-likelihood density modification

Thomas C. Terwilliger

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Maximum-likelihood density modification

Thomas C. Terwilliger

Structural Biology Group, Mail Stop M888, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence e-mail: terwilliger@lanl.gov

Received 6 December 1999
Accepted 17 April 2000

A likelihood-based approach to density modification is developed that can be applied to a wide variety of cases where some information about the electron density at various points in the unit cell is available. The key to the approach consists of developing likelihood functions that represent the probability that a particular value of electron density is consistent with prior expectations for the electron density at that point in the unit cell. These likelihood functions are then combined with likelihood functions based on experimental observations and with others containing any prior knowledge about structure factors to form a combined likelihood function for each structure factor. A simple and general approach to maximizing the combined likelihood function is developed. It is found that this likelihood-based approach yields greater phase improvement in model and real test cases than either conventional solvent flattening and histogram matching or a recent reciprocal-space solvent-flattening procedure [Terwilliger (1999), *Acta Cryst.* D**55**, 1863–1871].

1. Introduction

The phase information obtained from experimental measurements on macromolecules using either multiple isomorphous replacement or multiwavelength anomalous diffraction is often insufficient by itself for constructing a electron-density map useful for model building and interpretation. Many density-modification methods have been developed in recent years for improving the quality of electron-density maps by incorporation of prior knowledge about the features expected in these maps when they are obtained at high or moderate resolution (2–4 Å). Among the most powerful of these methods are solvent flattening, non-crystallographic symmetry averaging, histogram matching, phase extension, molecular replacement, entropy maximization and iterative model building (Abrahams, 1997; Bricogne, 1984, 1988; Cowtan & Main, 1993, 1996; Giacovazzo & Siliqi, 1997; Goldstein & Zhang, 1998; Gu *et al.*, 1997; Lunin, 1993; Perrakis *et al.*, 1997; Podjarny *et al.*, 1987; Prince *et al.*, 1988; Refaat *et al.*, 1996; Roberts & Brünger, 1995; Rossmann & Arnold, 1993; Vellieux *et al.*, 1995; Wilson & Agard, 1993; Xiang *et al.*, 1993; Zhang & Main, 1990; Zhang, 1993; Zhang *et al.*, 1997). The fundamental basis of density-modification methods is that there are many possible sets of structure-factor amplitudes and phases that are all reasonably probable based on the limited experimental data, and those structure factors that lead to maps that are most consistent with both the experimental data and the prior knowledge are the most likely overall. In these methods, the choice of prior information that is to be used and the procedure for combining prior infor-

© 2000 International Union of Crystallography
Printed in Denmark – all rights reserved

electronic reprint

mation about electron density with experimentally derived phase information are crucial parts.

Until recently, density modification, the combination of knowledge about expected features of an electron-density map with experimental phase information, has generally been carried out in a two-step procedure that is iterated until convergence. In the first step, an electron-density map obtained experimentally is modified in real space in order to make it consistent with expectations. This can consist of flattening solvent regions, averaging non-crystallographic symmetry-related regions or histogram matching, for example. In the second step, phases are calculated from the modified map and are combined with the experimental phases to form a new phase set.

The disadvantage of this real-space modification approach is that it is not at all clear how to weight the observed phases with those obtained from the modified map. This is a consequence of the fact that the modified map contains some of the same information as the original map and some new information. This difficulty has been recognized for a long time and a number of approaches have been designed to improve the relative weighting from these two sources, recently including the use of maximum-entropy methods and the use of weighting optimized using cross-validation (Xiang *et al.*, 1993; Roberts & Brünger, 1995; Cowtan & Main, 1996) and 'solvent flipping' (Abrahams, 1997).

2. Density modification by reciprocal-space-based likelihood optimization

We have recently developed a very different approach to combinining experimental phase information with prior knowledge about expected electron-density distributions in maps. Our approach is based on maximization of a combined likelihood function (Terwilliger, 1999). The fundamental idea is to express our knowledge about the probability of a set of structure factors $\{F_{\mathbf{h}}\}$ in terms of two quantities: (i) the likelihood of having measured the observed set of structure factors $\{F_{\mathbf{h}}^{\rm OBS}\}$ if this structure-factor set were correct and (ii) the likelihood that the map resulting from this structure-factor set $\{F_{\mathbf{h}}\}$ is consistent with our prior knowledge about this and other macromolecular structures.

When set up in this way, the overlap of information that occurred in the real-space modification methods is not present because the experimental and prior information are kept separate. Consequently, proper weighting of experimental and prior information only requires estimates of probability functions for each source of information.

The likelihood-based density-modification approach has a second very important advantage. This is that the derivatives of the likelihood functions with respect to individual structure factors can be readily calculated in reciprocal space by FFT-based methods. As a consequence, density modification simply becomes an optimization of a combined likelihood function by adjustment of structure factors. This makes density modification a remarkably simple but powerful approach, only requiring that suitable likelihood functions be constructed for each aspect of prior knowledge that is to be incorporated. We previously showed that such an approach could be applied to solvent flattening and that the resulting algorithm was greatly improved over methods depending on real-space modification and phase recombination (Terwilliger, 1999).

Here, we extend the idea of likelihood-based density modification to include prior information on the electron-density distribution from a wide variety of potential sources and demonstrate it on both the electron density in the solvent region and the region occupied by a macromolecule. First, we describe the mathematics of likelihood-based density modification in a practical formulation that is modified somewhat from the one we used for reciprocal-space solvent flattening (Terwilliger, 1999). We then show how a likelihood function for a map that includes information on both the solvent- and macromolecule-containing regions can be constructed and used.

3. Likelihood-based density modification

The basic idea of our likelihood-based density-modification procedure is that there are two key kinds of information about the structure factors for a crystal of a macromolecule. The first is the experimental phase and amplitude information. This can be thought of in terms of a likelihood (or log-likelihood) function $\mathrm{LL}^{\rm OBS}(F_{\mathbf{h}})$ for each structure factor $F_{\mathbf{h}}$, where the probability distribution for the structure factor $p^{\rm OBS}(F_{\mathbf{h}})$ is given by

$$p^{\rm OBS}(F_{\mathbf{h}}) = \exp\{\mathrm{LL}^{\rm OBS}(F_{\mathbf{h}})\}. \quad (1)$$

For reflections with accurately measured amplitudes, the chief uncertainty in $F_{\mathbf{h}}$ will be in the phase, while for unmeasured or poorly measured reflections it will be in both phase and amplitude.

The second kind of information about structure factors in this formulation is the likelihood of the map resulting from them. For example, for most macromolecular crystals a set of structure factors $\{F_{\mathbf{h}}\}$ that leads to a map with a flat region corresponding to solvent is more likely to be correct than one that leads to a map with uniform variation everywhere. This map-likelihood function describes the probability that the map obtained from a set of structure factors is compatible with our expectations,

$$p^{\rm MAP}(F_{\mathbf{h}}) = \exp\{\mathrm{LL}^{\rm MAP}(F_{\mathbf{h}})\}. \quad (2)$$

We then combine our two principal sources of information along with any prior knowledge of the structure factors to yield the likelihood of a particular set of structure factors,

$$\mathrm{LL}(\{F_{\mathbf{h}}\}) = \mathrm{LL}^{o}(\{F_{\mathbf{h}}\}) + \mathrm{LL}^{\rm OBS}(\{F_{\mathbf{h}}\}) + \mathrm{LL}^{\rm MAP}(\{F_{\mathbf{h}}\}), \quad (3)$$

where $\mathrm{LL}^{o}(\{F_{\mathbf{h}}\})$ includes any structure-factor information that is known in advance, such as the distribution of intensities of structure factors (Wilson, 1949).

electronic reprint

3.1. Approximating the likelihood function to simplify the procedure

In order to maximize the overall likelihood function in (3) we are going to need to know how the map-likelihood function changes in response to changes in structure factors. In the case of the map-likelihood function $\mathrm{LL}^{\mathrm{MAP}}(\{F_{\mathbf{h}}\})$ this can be thought of as two separate relationships, the response of the likelihood function to changes in electron density and the changes in electron density as a function of changes in structure factors. In principle, the likelihood of a particular map is a complicated function of the electron density over the entire map. Furthermore, the value of any structure factor affects the electron density everywhere in the map. To simplify the mathematics, we explicitly use a low-order approximation to the likelihood function for a map instead of attempting to evaluate the function precisely. As Fourier transformation is a linear process, each reflection contributes independently to the electron density at a given point in the cell. Although the log-likelihood of the electron density might have any form, we expect that for sufficiently small changes in structure factors, a first-order approximation to the log-likelihood function would apply and each reflection would also contribute relatively independently to changes in the log-likelihood function.

Consequently, we construct a local approximation to the map-likelihood function, neglecting correlations among different points in the map and between reflections, expecting that it might describe reasonably accurately how the likelihood function would vary in response to small changes in structure factors.

By neglecting correlations among different points in the map, we can write the log-likelihood for the whole electron-density map as the sum of the log-likelihoods of the densities at each point in the map, normalized to the volume of the unit cell and the number of reflections used to construct it (Terwilliger, 1999),

$$\mathrm{LL}^{\mathrm{MAP}}(\{F_{\mathbf{h}}\}) \simeq (N_{\mathrm{REF}}/V)\int_V \mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\,\mathrm{d}^3\mathbf{x}. \qquad (4)$$

Additionally, by treating each reflection as independently contributing to the likelihood function, we can write a local approximation to the log-likelihood of the density at each point. This approximation is given by the sum over all reflections of first few terms of a Taylor's series expansion around the value obtained with the starting structure factors $\{F^o_{\mathbf{h}}\}$ used in a cycle of density modification, $\mathrm{LL}[\rho(\mathbf{x}, \{F^o_{\mathbf{h}}\})]$,

$$\begin{aligned}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})] \simeq\ & \mathrm{LL}[\rho(\mathbf{x}, \{F^o_{\mathbf{h}}\})] \\ &+ \sum_{\mathbf{h}}\Bigg\{\Delta F_{\mathbf{h},\parallel}\frac{\partial}{\partial F_{\mathbf{h},\parallel}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})] \\ &+ \frac{1}{2}\Delta F^2_{\mathbf{h},\parallel}\frac{\partial^2}{\partial F^2_{\mathbf{h},\parallel}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})] \\ &+ \Delta F_{\mathbf{h},\perp}\frac{\partial}{\partial F_{\mathbf{h},\perp}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})] \\ &+ \frac{1}{2}\Delta F^2_{\mathbf{h},\perp}\frac{\partial^2}{\partial F^2_{\mathbf{h},\perp}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})] + \ldots\Bigg\}, \qquad (5)\end{aligned}$$

where $\Delta F_{\mathbf{h},\parallel}$ and $\Delta F_{\mathbf{h},\perp}$ are the differences between $F_{\mathbf{h}}$ and $F^o_{\mathbf{h}}$ along the directions of $F^o_{\mathbf{h}}$ and $iF^o_{\mathbf{h}}$, respectively.

Combining (4) and (5), we can write an expression for the map log-likelihood function,

$$\begin{aligned}\mathrm{LL}^{\mathrm{MAP}}(\{F_{\mathbf{h}}\}) \simeq\ & \mathrm{LL}^{\mathrm{MAP}}[\rho(\mathbf{x}, \{F^o_{\mathbf{h}}\})] \\ &+ (N_{\mathrm{REF}}/V)\sum_{\mathbf{h}}\Delta F_{\mathbf{h},\parallel}\int_V \frac{\partial}{\partial F_{\mathbf{h},\parallel}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\,\mathrm{d}^3\mathbf{x} \\ &+ \frac{1}{2}\Delta F^2_{\mathbf{h},\parallel}\int_V \frac{\partial^2}{\partial F^2_{\mathbf{h},\parallel}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\,\mathrm{d}^3\mathbf{x} \\ &+ \Delta F_{\mathbf{h},\perp}\int_V \frac{\partial}{\partial F_{\mathbf{h},\perp}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\,\mathrm{d}^3\mathbf{x} \\ &+ \frac{1}{2}\Delta F^2_{\mathbf{h},\perp}\int_V \frac{\partial^2}{\partial F^2_{\mathbf{h},\perp}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\,\mathrm{d}^3\mathbf{x}. \qquad (6)\end{aligned}$$

3.2. FFT-based calculation of the reciprocal-space derivatives of log-likelihood of electron density $\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]$

The integrals in (6) can be rewritten in a form that is suitable for evaluation by an FFT-based approach. Considering the first integral in (6), we use the chain rule to write that

$$\frac{\partial}{\partial F_{\mathbf{h},\parallel}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})] = \frac{\partial}{\partial \rho(\mathbf{x})}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\frac{\partial}{\partial F_{\mathbf{h},\parallel}}\rho(\mathbf{x}) \qquad (7)$$

and note that the derivative of $\rho(\mathbf{x})$ with respect to $F_{\mathbf{h},\parallel}$ for a particular index $\mathbf{h}$ is given by

$$\frac{\partial}{\partial F_{\mathbf{h},\parallel}}\rho(\mathbf{x}) = \frac{2}{V}\mathrm{Re}[\exp(i\varphi_{\mathbf{h}})\exp(-2\pi i\mathbf{h}\cdot\mathbf{x})]. \qquad (8)$$

Now we can rearrange and rewrite the first integral in (6) in the form

$$\int_V \frac{\partial}{\partial F_{\mathbf{h},\parallel}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\,\mathrm{d}^3\mathbf{x} = \frac{2}{V}\mathrm{Re}[\exp(i\varphi_{\mathbf{h}})a^*_{\mathbf{h}}], \qquad (9)$$

where the complex number $a_{\mathbf{h}}$ is a term in the Fourier transform of $\{\partial/[\partial\rho(\mathbf{x})]\}\ \mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]$,

$$a_{\mathbf{h}} = \int_V \frac{\partial}{\partial \rho(\mathbf{x})}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\exp(2\pi i\mathbf{h}\cdot\mathbf{x})\,\mathrm{d}^3\mathbf{x}. \qquad (10)$$

In space groups other than $P1$, only a unique set of structure factors need to be specified to calculate an electron-density map. Taking space-group symmetry into account, (9) can be generalized (Terwilliger, 1999) to read

$$\int_V \frac{\partial}{\partial F_{\mathbf{h},\parallel}}\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\,\mathrm{d}^3\mathbf{x} = \frac{2}{V}\sum_{\mathbf{h}'}\mathrm{Re}[\exp(i\varphi_{\mathbf{h}'})a^*_{\mathbf{h}'}], \qquad (11)$$

where the indices $\mathbf{h}'$ are all indices equivalent to $\mathbf{h}$ owing to space-group symmetry.

A similar procedure can be used to rewrite the second integral in (6), yielding the expression

electronic reprint

$$\int_V \frac{\partial^2}{\partial F_{\mathbf{h},\parallel}^2} \mathrm{LL}[\rho(\mathbf{x}, \{F_\mathbf{h}\})] \mathrm{d}^3\mathbf{x} = \frac{2}{V^2} \sum_{\mathbf{h}',\mathbf{k}'} \mathrm{Re}[\exp(-i\varphi_{\mathbf{h}'}) \exp(i\varphi_{\mathbf{k}'}) b_{\mathbf{h}'-\mathbf{k}'} + \exp(-i\varphi_{\mathbf{h}'}) \exp(-i\varphi_{\mathbf{k}'}) b_{\mathbf{h}'+\mathbf{k}'}], \quad (12)$$

where the indices $\mathbf{h}'$ and $\mathbf{k}'$ are each all indices equivalent to $\mathbf{h}$ owing to space-group symmetry and where the coefficients $b_\mathbf{h}$ are again terms in a Fourier transform, this time of the second derivative of the log-likelihood of the electron density,

$$b_\mathbf{h} = \int_V \frac{\partial^2}{\partial \rho(\mathbf{x})^2} \mathrm{LL}[\rho(\mathbf{x}, \{F_\mathbf{h}\})] \exp(2\pi i \mathbf{h} \cdot \mathbf{x}) \, \mathrm{d}^3\mathbf{x}. \quad (13)$$

The third and fourth integrals in (6) can be rewritten in a similar way, yielding the expressions

$$\int_V \frac{\partial}{\partial F_{\mathbf{h},\perp}} \mathrm{LL}[\rho(\mathbf{x}, \{F_\mathbf{h}\})] \, \mathrm{d}^3\mathbf{x} = \frac{2}{V} \sum_{\mathbf{h}'} \mathrm{Re}[i \exp(i\varphi_{\mathbf{h}'}) a_{\mathbf{h}'}^*] \quad (14)$$

and

$$\int_V \frac{\partial^2}{\partial F_{\mathbf{h},\parallel}^2} \mathrm{LL}[\rho(\mathbf{x}, \{F_\mathbf{h}\})] \mathrm{d}^3\mathbf{x} = \frac{2}{V^2} \sum_{\mathbf{h}',\mathbf{k}'} \mathrm{Re}[\exp(-i\varphi_{\mathbf{h}'}) \exp(i\varphi_{\mathbf{k}'}) b_{\mathbf{h}'-\mathbf{k}'} - \exp(-i\varphi_{\mathbf{h}'}) \exp(-i\varphi_{\mathbf{k}'}) b_{\mathbf{h}'+\mathbf{k}'}]. \quad (15)$$

The significance of (4) through (15) is that we now have a simple expression (6) describing how the map-likelihood function $\mathrm{LL}^{\mathrm{MAP}}(\{F_\mathbf{h}\})$ varies when small changes are made in the structure factors. Evaluating this expression only requires that we be able to calculate the first and second derivatives of log-likelihood of the electron density with respect to electron density at each point in the map and carry out an FFT. Furthermore, maximization of the (local) overall likelihood function (3) becomes straightforward, as every reflection is treated independently. It consists simply of adjusting each structure factor to maximize its contribution to the approximation to the likelihood function through (3) to (15).

In practice, instead of directly maximizing the overall likelihood function, we use it here to estimate the probability distribution for each structure factor (Terwilliger, 1999) and then integrate this probability distribution over the phase (or phase and amplitude) of the reflection to obtain a weighted mean estimate of the structure factor. Using (3) to (15), the probability distribution for an individual structure factor can be written as

$$\begin{aligned} \ln p(F_\mathbf{h}) \simeq\ & \mathrm{LL}^o(F_\mathbf{h}) + \mathrm{LL}^{\mathrm{OBS}}(F_\mathbf{h}) \\ & + (2N_{\mathrm{REF}}/V^2) \Delta F_{\mathbf{h},\parallel} \sum_{\mathbf{h}'} \mathrm{Re}[\exp(i\varphi_{\mathbf{h}'}) a_{\mathbf{h}'}^*] \\ & + (2N_{\mathrm{REF}}/V^3) \Delta F_{\mathbf{h},\parallel}^2 \sum_{\mathbf{h}',\mathbf{k}'} \mathrm{Re}[\exp(-i\varphi_{\mathbf{h}'}) \exp(i\varphi_{\mathbf{k}'}) b_{\mathbf{h}'-\mathbf{k}'} \\ & + \exp(-i\varphi_{\mathbf{h}'}) \exp(-i\varphi_{\mathbf{k}'}) b_{\mathbf{h}'+\mathbf{k}'}] \\ & + (2N_{\mathrm{REF}}/V^2) \Delta F_{\mathbf{h},\perp} \sum_{\mathbf{h}'} \mathrm{Re}[i \exp(i\varphi_{\mathbf{h}'}) a_{\mathbf{h}'}^*] \\ & + (2N_{\mathrm{REF}}/V^3) \Delta F_{\mathbf{h},\perp}^2 \sum_{\mathbf{h}',\mathbf{k}'} \mathrm{Re}[\exp(-i\varphi_{\mathbf{h}'}) \exp(i\varphi_{\mathbf{k}'}) b_{\mathbf{h}'-\mathbf{k}'} \\ & - \exp(-i\varphi_{\mathbf{h}'}) \exp(-i\varphi_{\mathbf{k}'}) b_{\mathbf{h}'+\mathbf{k}'}], \end{aligned} \quad (16)$$

where, as above, the indices $\mathbf{h}'$ and $\mathbf{k}'$ are each all indices equivalent to $\mathbf{h}$ owing to space-group symmetry and the coefficients $a_\mathbf{h}$ and $b_\mathbf{h}$ are given in (10) and (13). Also as before, $\Delta F_{\mathbf{h},\parallel}$ and $\Delta F_{\mathbf{h},\perp}$ are the differences between $F_\mathbf{h}$ and $F_\mathbf{h}^o$ along the directions of $F_\mathbf{h}^o$ and $iF_\mathbf{h}^o$, respectively. All the quantities in (16) can be readily calculated once a likelihood function for the electron density and its derivatives are obtained.

4. Likelihood function for an electron-density map with errors

A key step in likelihood-based density modification is the decision as to the likelihood function for values of the electron density at a particular location in the map. For the present purpose, an expression for the log-likelihood of the electron density $\mathrm{LL}[\rho(\mathbf{x}, \{F_\mathbf{h}\})]$ at a particular location $\mathbf{x}$ in a map is needed that depends on whether the point $\mathbf{x}$ is within the solvent region or the protein region. In general, this function might depend on whether the point satisfies any of a wide variety of conditions, such as being at a certain location in a known fragment of structure or being at a certain distance from some other feature of the map. We discussed previously (Terwilliger, 1999) how one might incorporate information on the environment of $\mathbf{x}$ by writing the log-likelihood function as the log of the sum of conditional probabilities dependent on the environment of $\mathbf{x}$,

$$\mathrm{LL}[\rho(\mathbf{x}, \{F_\mathbf{h}\})] = \ln\{p[\rho(\mathbf{x})|\mathrm{PROT}] p_{\mathrm{PROT}}(\mathbf{x}) + p[\rho(\mathbf{x})|\mathrm{SOLV}] p_{\mathrm{SOLV}}(\mathbf{x})\}, \quad (17)$$

where $p_{\mathrm{PROT}}(\mathbf{x})$ is the probability that $\mathbf{x}$ is in the protein region and $p[\rho(\mathbf{x})|\mathrm{PROT}]$ is the conditional probability for $\rho(\mathbf{x})$ given that $\mathbf{x}$ is in the protein region, and $p_{\mathrm{SOLV}}(\mathbf{x})$ and $p[\rho(\mathbf{x})|\mathrm{SOLV}]$ are the corresponding quantities for the solvent region. The probability that $\mathbf{x}$ is in the protein or solvent regions is estimated by a modification of the methods of Wang (1985) and Leslie (1987) as described previously (Terwilliger, 1999). If there were more than just solvent and protein regions that identified the environment of each point, then (17) could be modified to include those as well.

In developing (13) to (15), the derivatives of the likelihood function for electron density were intended to represent how the likelihood function changed when small changes in one structure factor were made. Surprisingly, the likelihood function that is most appropriate for our present

electronic reprint

purposes in this case is not a globally correct one. Instead, it is a likelihood function that represents how the overall likelihood function varies in response to small changes in one structure factor, keeping all others constant. To see the difference, consider the electron density in the solvent region of a macromolecular crystal. In an idealized situation with all possible reflections included, the electron density might be exactly equal to a constant in this region. The goal in using (16) is to obtain the relative probabilites for each possible value of a particular unknown structure factor $F_{\mathbf{h}}$. If all other structure factors were exact, then the globally correct likelihood function for the electron density (zero unless the solvent region is perfectly flat) would correctly identify the correct value of the unknown structure factor. Now suppose we had imperfect phase information. The solvent region would have a significant amount of noise and its value would no longer be a constant. If we use the globally correct likelihood function for the electron density, we would assign a zero probability to any value of the structure factor that did not lead to an absolutely flat solvent region. This is clearly unreasonable, because all the other (incorrect) structure factors are contributing noise that exists regardless of the value of this structure factor.

This situation is very similar to the one encountered in structure refinement of macromolecular structures where there is a substantial deficiency in the model. The errors in all the other structure factors in the present discussion correspond to the deficiency in the macromolecular model in the refinement case. The appropriate variance to use as a weighting factor in refinement includes the estimated model error as well as the error in measurement (*e.g.* Terwilliger & Berendzen, 1996; Pannu & Read, 1996). Similarly, the appropriate likelihood function for electron density for use in the present method is one in which the overall uncertainty in the electron density arising from all reflections other than the one being considered is included in the variance.

A likelihood function of this kind for the electron density can be developed using a model in which the electron density arising from all reflections but one is treated as a random variable (Terwilliger & Berendzen, 1996; Pannu & Read, 1996). Suppose that the true value of the electron density at $\mathbf{x}$ was known and was given by ρ_T. Then consider that we have estimates of all the structure factors, but that substantial errors exist in each one. The expected value of the estimate of this electron density obtained from current estimates of all the structure factors ($\rho_{\rm OBS}$) will be given by $\langle\rho_{\rm OBS}\rangle = \beta\rho_T$ and the expected value of the variance by $\langle(\rho_{\rm OBS} - \beta\rho_T)^2\rangle = \sigma^2_{\rm MAP}$. The factor β represents the expectation that the calculated value of ρ will be smaller than the true value. This is for two reasons. One is that such a estimate may be calculated using figure-of-merit weighted estimates of structure factors, which will be smaller than the correct ones. The other is that phase error in the structure factors systematically leads to a bias towards a smaller component of the structure factor along the direction of the true structure factor. This is the same effect that leads to the D correction factor in maximum-likelihood refinement (Pannu & Read, 1996).

A probability function for the electron density at this point that is appropriate for assessing the probabilities of values of the structure factor for one reflection can now be written as

$$p(\rho) = \exp\left[-\frac{(\rho - \beta\rho_T)^2}{2\sigma^2_{MAP}}\right]. \qquad (18)$$

In a slightly more complicated case, where the value of ρ_T is not known exactly but rather has an uncertainty σ_T, (18) becomes

$$p(\rho) = \exp\left[-\frac{(\rho - \beta\rho_T)^2}{2(\beta^2\sigma^2_T + \sigma^2_{\rm MAP})}\right]. \qquad (19)$$

Finally, in the case where only a probability distribution $p(\rho_T)$ for ρ_T is known, (18) becomes

$$p(\rho) = \int_{\rho_T} p(\rho_T)\exp\left[-\frac{(\rho - \beta\rho_T)^2}{2\sigma^2_{\rm MAP}}\right]{\rm d}\rho_T. \qquad (20)$$

4.1. Likelihood function for solvent- and macromolecule-containing regions of a map

Using (19) and (20), we are now in a position to use a histogram-based approach (Goldstein & Zhang, 1998; Lunin, 1993; Zhang & Main, 1990) to develop likelihood functions for the solvent region of a map and for the macromolecule-containing region of a map. The approach is simple. The probability distribution for true electron density in the solvent or macromolecule regions of a crystal structure is obtained from an analysis of model structures and represented as a sum of Gaussian functions of the form

$$p(\rho_T) = \sum_k w_k \exp\left[-\frac{(\rho_T - c_k)^2}{2\sigma^2_k}\right]. \qquad (21)$$

If the values of β and $\sigma_{\rm MAP}$ were known for an experimental map with unknown errors but identified solvent and protein regions, then using (19) we could write the probability distribution for electron density in the each region of the map as

$$p(\rho_T) = \sum_k w_k \exp\left[-\frac{(\rho_T - \beta c_k)^2}{2(\beta^2\sigma^2_k + \sigma^2_{\rm MAP})}\right], \qquad (22)$$

with the appropriate values of β and $\sigma_{\rm MAP}$. In practice, the values of β and $\sigma_{\rm MAP}$ are estimated by a least-squares fitting of the probability distribution given in (22) to the one found in the experimental map. This procedure has the advantage that the scale of the experimental map does not have to be accurately determined. Then (22) is used with the refined values of β and $\sigma_{\rm MAP}$ as the probability function for electron density in the corresponding region (solvent or macromolecule) of the map.

5. Evaluation of maximum-likelihood density modification with model and real data

To evaluate the utility of maximum-likelihood density modification as described here, we carried out tests using the same

electronic reprint

model and experimental data that we previously analyzed using reciprocal-space solvent flattening and by real-space solvent flattening (Terwilliger, 1999). The first test case consisted of a set of phases constructed from a model with 32–68% of the volume of the unit cell taken up by protein. The initial effective figure of merit of the phases overall [$\langle\cos(\Delta\varphi)\rangle$] was about 0.40. In our previous tests, we showed that both real-space and reciprocal-space solvent flattening improved the quality of phasing considerably. In the current tests, the real-space density modification included both solvent flattening and histogram matching to be as comparable as possible to the maximum-likelihood density modification we have developed.

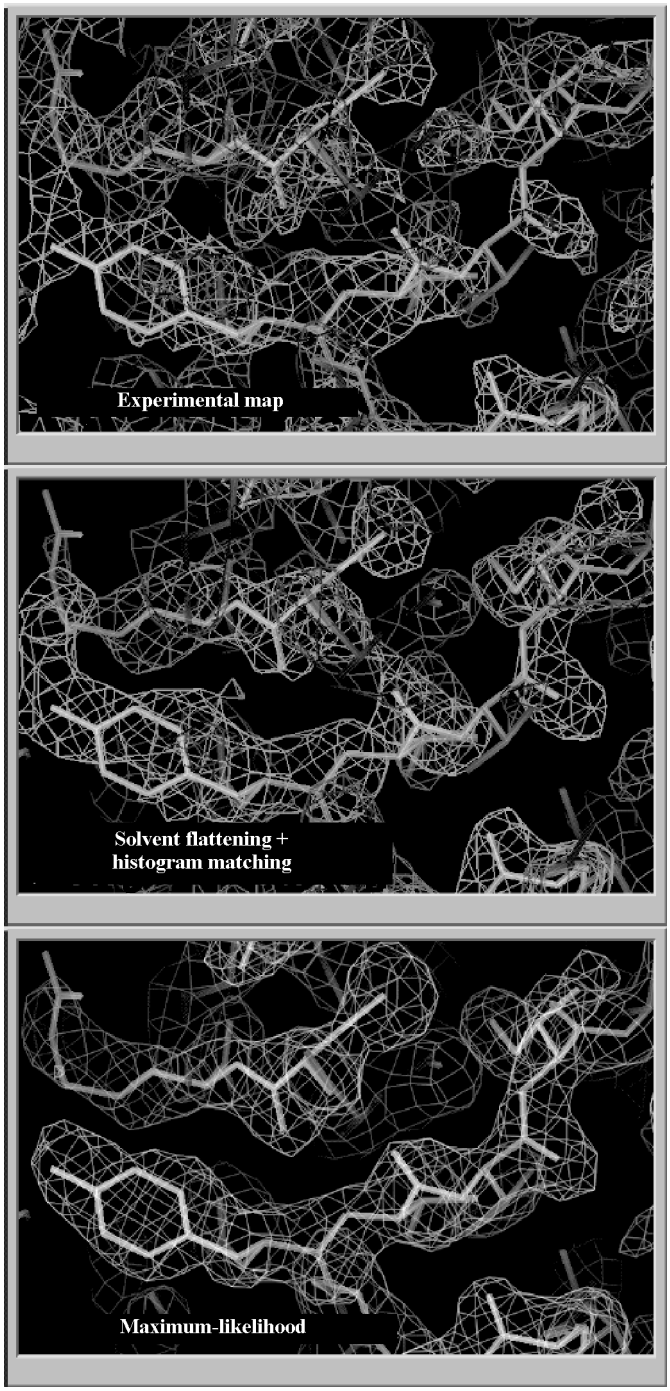


Figure 1
Sections of electron density obtained before and after density modification of phases obtained for IF-5A (Peat *et al.*, 1998) phased using one Se atom in the asymmetric unit. Density modification was carried out by real-space solvent flattening and histogram matching or by maximum-likelihood solvent flattening. Values for real-space density modification were carried out using the program *dm* (Cowtan & Main, 1996), version 1.8, using solvent flattening with histogram matching. Starting phases were calculated with *SOLVE* (Terwilliger & Berendzen, 1999). The correlation coefficient between the map calculated based on the refined model of IF-5A and the starting map was 0.37, for the real-space modifed map it was 0.65 and for the maximum-likelihood map it was 0.79.

Table 1 shows the the quality of phases obtained after each method for density modification was applied to this model case. In all cases, maximum-likelihood density modification of this map resulted in phases with an effective figure of merit [$\langle\cos(\Delta\varphi)\rangle$] higher than any of the other methods. When the fraction of solvent in the model unit cell was 50%, for example, maximum-likelihood density modification yielded an effective figure of merit of 0.83, while real-space solvent flattening and histogram matching resulted in an effective figure of merit of 0.62 and reciprocal-space solvent flattening gave an effective figure of merit of 0.67.

The utility of maximum-likelihood density modification was also compared with real-space density modification and with reciprocal-space solvent flattening using experimental multiwavelength (MAD) data on initiation factor 5A (IF-5A) recently determined in our laboratory (Peat *et al.*, 1998). IF-5A crystallizes in space group *I*4, with unit-cell parameters $a = 114$, $b = 114$, $c = 33$ Å, one molecule in the asymmetric unit and a solvent content of about 60%. The structure was solved using MAD phasing based on three Se atoms in the asymmetric unit at a resolution of 2.2 Å. For purposes of testing density-modification methods, only one of the three selenium sites was used in phasing here, resulting in a starting map with a correlation coefficient to the map calculated using the final refined structure of 0.37. The resulting electron-density map was improved by real-space density modification using solvent flattening and histogram matching with *dm* (Cowtan & Main, 1996), by real-space density modification using solvent flipping (Abrahams, 1997) and after maximum-likelihood density modification. The 'experimental' map, the *dm*-modified map and the maximum-likelihood map are shown in Fig. 1. As anticipated, the real-space modified map obtained with *dm* is improved over the starting map; it has a correlation coefficient of 0.65. Density modification including solvent flipping yielded a similar improvement, with a correlation coefficient of 0.61 to the model map. The maximimum-likelihood modified map was much more substantially improved, with a correlation coefficient to the map based on a refined model of 0.79.

6. Discussion

We have shown here that a maximum-likelihood approach can be used to carry out density modification on macromolecular crystal structures and that this approach is much more powerful than either conventional density modification based on solvent flattening and histogram matching or our recent reciprocal-space solvent-flattening procedure (Terwilliger,

electronic reprint

Table 1
Correlation of density-modified phases with true phases [$\langle\cos(\Delta\varphi)\rangle$] for model data in a unit cell containing 32–68% solvent.

Data and analysis using reciprocal-space solvent flattening are from Terwilliger (1999). Phases with simulated errors for 6906 data from ∞ to 3.0 Å were constructed using a model consisting of coordinates from a dehalogenase enzyme from *Rhodococcus* species ATCC 55388 (American Type Culture Collection, 1992) determined recently in our laboratory (Newman *et al.*, 1999; PDB entry 1bn7), except that some of the atoms were not included in order to vary the fraction of solvent in the unit cell. The cell was in space group $P2_12_12$, with unit-cell parameters a = 94, b = 80, c = 43 Å and one molecule in the asymmetric unit. Phases with simulated errors were generated by adding phase errors as described in Terwilliger (1999) to yield an average value of the cosine of the phase error (*i.e.* the true figure of merit of the phasing) of $\langle\cos(\Delta\varphi)\rangle$ = 0.42 for acentric and 0.39 for centric reflections. The model data with simulated errors was then density modified by the maximum-likelihood method described here, by reciprocal-space solvent flattening (Terwilliger, 1999) and by a real-space method as implemented in the program *dm* (Cowtan & Main, 1996), version 1.8, using solvent flattening and histogram matching.

Fraction protein (%)	Starting	Real-space solvent flattening	Reciprocal-space solvent flattening	Maximum-likelihood solvent flattening
32	0.41	0.64	0.85	0.87
42	0.40	0.62	0.67	0.83
50	0.41	0.54	0.56	0.77
68	0.42	0.48	0.41	0.53

1999). The reason the approach works so well is that the relative weighting of experimental phase information and of expected electron-density distributions is taken care of automatically by keeping the two sources of information clearly delineated and by defining suitable probability distributions for each.

The maximum-likelihood approach to improvement of crystallographic phases has been developed extensively by Bricogne and others (*e.g.* Bricogne, 1984, 1988; Lunin, 1993). The importance of the present work and of our recent work on reciprocal-space solvent flattening (Terwilliger, 1999) is that we have developed a simple, effective and general way to carry it out.

Although we have demonstrated here only two sources of expected electron-density distributions (probability distributions for solvent regions and for protein-containing regions), the methods developed here can be applied directly to a wide variety of sources of information. For example, any source of information about the expected electron density at a particular point in the unit cell that can be written in a form such as the one in (22) can be used in our procedure to describe the likelihood that a particular value of electron density is consistent with expectations.

Sources of expected electron-density information that are especially suitable for application to our method include non-crystallographic symmetry and the knowledge of the location of fragments of structure in the unit cell. In the case of non-crystallographic symmetry, the probability distribution for electron density at one point in the unit cell can be written using (22) with a value of ρ_T equal to the weighted mean at all non-crystallographically equivalent points in the cell. The value of σ_T could be calculated based on their variance and the value of $\sigma_{\rm MAP}$. In the case of knowledge of locations of fragments in the unit cell, this knowledge can be used to calculate estimates of the electron-density distribution for each point in the neighborhood of the fragment. These electron-density distributions can then in turn be used just as described above to estimate ρ_T and σ_T in this region. An iterative process, in which fragment locations are identified by cross-correlation or related searches (density modification) is applied and additional searches are carried out to further generate a model for the electron density, could even be developed, in an extension of the iterative chain-tracing methods described by Wilson & Agard (1993). Such a process could potentially even be used to construct a complete probabilistic model of a macromolecular structure using structure-factor estimates obtained from molecular replacement with fragments of macromolecular structures as a starting point. In all these cases, the electron-density information could be included in much the same way as the probability distributions we used here for the solvent and protein regions of maps. In each case, the key is an estimate of the probability distribution for electron density at a point in the map that contains some information that restricts the likely values of electron density at that point. The procedure could be further extended by having probability distributions describing the likelihood that a particular point in the unit cell is within protein, within solvent, within a particular location in a fragment of protein structure, within a non-crystallographically related region and so on. These probability distributions could be overlapping or non-overlapping. Then for each category of points, the probability distribution for electron density within that category could be formulated as in (22) and our current methods applied.

The procedure described here differs from the reciprocal-space solvent-flattening procedure described previously (Terwilliger, 1999) in two important ways. One is that the expected electron-density distribution in the non-solvent region is included in the calculations and a formalism for incorporating information about the electron-density map from a wide variety of sources is developed. The second is that the probability distribution for the electron density is calculated using (22) for both solvent and non-solvent regions and values of the scaling parameter β and the map uncertainty $\sigma_{\rm MAP}$ are estimated by a fitting of model and observed electron-density distributions. This fitting process makes the whole procedure very robust with respect to scaling of the experimental data, which otherwise would have to be very accurate in order that the model electron-density distributions be applicable.

Software for carrying out maximum-likelihood density modification ('*Resolve*') and complete documentation is available on the WWW at http://resolve.lanl.gov.

The author would like to thank Joel Berendzen for helpful discussions and the NIH and the US Department of Energy for generous support.

electronic reprint

References

Abrahams, J. P. (1997). *Acta Cryst.* D**53**, 371–376.
American Type Culture Collection (1992). *Catalogue of Bacteria and Bacteriophages*, 18th ed., pp. 271–272.
Bricogne, G. (1984). *Acta Cryst.* A**40**, 410–445.
Bricogne, G. (1988). *Acta Cryst.* A**44**, 517–545.
Cowtan, K. D. & Main, P. (1993). *Acta Cryst.* D**49**, 148–157
Cowtan, K. D. & Main, P. (1996). *Acta Cryst.* D**52**, 43–48.
Giacovazzo, C. & Siliqi, D. (1997). *Acta Cryst.* A**53**, 789–798.
Goldstein, A. & Zhang, K. Y. J. (1998). *Acta Cryst.* D**54**, 1230–1244.
Gu, Y., Zheng, Ch., Zhao, Y., Ke, H. & Fan, H. (1997). *Acta Cryst.* D**53**, 792–794.
Leslie, A. G. W. (1987). *Proceedings of the CCP4 Study Weekend*, pp. 25–31. Warrington: Daresbury Laboratory.
Lunin, V. Y. (1993). *Acta Cryst.* D**49**, 90–99.
Newman, J., Peat, T. S., Richard, R., Kan, L., Swanson, P. W., Affholter, J. A., Holmes, I. H., Schindler, J. F., Unkefer, C. J. & Terwilliger, T. C. (1999). *Biochemistry*, **38**, 16105–16114.
Pannu, N. S. & Read, R. J. (1996). *Acta Cryst.* A**52**, 659–668.
Peat, T. S., Newman, J., Waldo, G. S., Berendzen, J. & Terwilliger, T. C. (1998). *Structure*, **15**, 1207–1214.
Perrakis, A., Sixma, T. K., Wilson, K. S. & Lamzin, V. S. (1997). *Acta Cryst.* D**53**, 448–455.
Podjarny, A. D., Bhat, T. N. & Zwick, M. (1987). *Annu. Rev. Biophys. Biophys. Chem.* **16**, 351–373.
Prince, E., Sjolin, L. & Alenljung, R. (1988). *Acta Cryst.* A**44**, 216–222.
Refaat, L. S., Tate, C. & Woolfson, M. M. (1996). *Acta Cryst.* D**52**, 252–256.
Roberts, A. L. U. & Brünger, A. T. (1995). *Acta Cryst.* D**51**, 990–1002.
Rossmann, M. G. & Arnold, E. (1993). *International Tables for Crystallography.* Vol. B, edited by U. Shmueli, pp. 230–258. Dordrecht/Boston/London: Kluwer Academic Publishers.
Terwilliger, T. C. (1999). *Acta Cryst.* D**55**, 1863–1871.
Terwilliger, T. C. & Berendzen, J. (1996). *Acta Cryst.* D**51**, 609–618.
Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**, 849–861.
Vellieux, F. M. D. A. P., Hunt, J. F., Roy, S. & Read, R. J. (1995). *J. Appl. Cryst.* **28**, 347–351.
Wang, B.-C. (1985). *Methods Enzymol.* **115**, 90–112.
Wilson, A. J. C. (1949). *Acta Cryst.* **2**, 318–321.
Wilson, C. & Agard, D. A. (1993). *Acta Cryst.* A**49**, 97–104.
Xiang, S., Carter, C. W. Jr, Bricogne, G. & Gilmore, C. J. (1993). *Acta Cryst.* D**49**, 193–212.
Zhang, K. Y. J. (1993). *Acta Cryst.* D**49**, 213–222.
Zhang, K. Y. J., Cowtan, K. D. & Main, P. (1997). *Methods Enzymol.* **277**, 53–64.
Zhang, K. Y. J. & Main, P. (1990). *Acta Cryst.* A**46**, 41–46.

electronic reprint

Licensing for commercial organizations

For commercial institutions, you need to contact our licensing office to obtain the forms. Please contact T. Allen Morris in the Technology Transfer Division at the address:

T. Allen Morris
Industrial Business Development Division Office, MS C334
Los Alamos National Laboratory
Los Alamos, NM 87545

Tel: 505-665-9597
Fax: 505-665-0154

email: tamorris@lanl.gov

Index for SOLVE manual

Note: The links in this index will put you before the position of the word you are looking for. You may need to use FIND (control-F) on your browser to find the word itself.

- a_b_to_f_phi
  - misc.html (math)
  - misc.html (script for math)
- acceptance
  - all_keywords.html (veryquick)
  - auto_keywords.html
  - different.html
  - next.html (addsolve)
  - notes/solve_brief_examples.html (beta-catenin)
- acentrics
  - misc.html (weights)
- addsolve
  - FAQS.html (input)
  - FAQS.html (input2)
  - all_keywords.html (file_names)
  - all_keywords.html (solve_control)
  - auto_keywords.html
- affected
  - FAQS.html (more)
- alias
  - all_commands.html
  - intro.html (commands)
- aliases
  - intro.html (commands)
- all_keywords
  - auto_keywords.html
- allocated
  - FAQS.html (sbin_loader)
- alwyn
  - next.html

- analyze_mad
  - all_keywords.html (file_names)
  - all_commands.html
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - generate.html
- analyze_mir
  - all_keywords.html (file_names)
  - all_commands.html
  - analyze_mir.html
  - analyze_mir.html (script-analyze_mir)
  - generate.html
- analyze_solve
  - FAQS.html (input)
  - FAQS.html (input2)
  - FAQS.html (hand)
  - all_keywords.html (file_names)
  - all_keywords.html (solve_control)
- anatscale
  - all_keywords.html (heavy_control)
- ancut
  - all_keywords.html (read_reject)
  - localscale.html (keywords for localscale)
- anisotropic
  - all_keywords.html (veryquick)
  - all_keywords.html (atom_input)
- ano
  - all_keywords.html (column_numbers)
  - binary.html (dorgbn files)
  - heavy.html (statistical output)
  - localscale.html
  - localscale.html (more on localscale)
- anofourier
  - maps.html
  - maps.html (keywords for maps)
- anom
  - all_keywords.html (heavy_control)
  - all_commands.html
  - formats.html (ccp4 mtz)

    - madbst.html (how madbst works)
- anomalous
    - FAQS.html (good)
    - FAQS.html (checksolve)
    - FAQS.html (hand)
    - all_keywords.html (column_numbers)
    - all_keywords.html (solve_control)
- anomalous difference
    - all_keywords.html (column_numbers)
    - analyze_mad.html
    - analyze_mir.html
    - formats.html (ccp4 mtz)
    - heavy.html (statistical output)
- anomalous differences
    - FAQS.html (good)
    - FAQS.html (hand)
    - all_keywords.html (solve_control)
    - all_keywords.html (heavy_control)
    - analyze_mad.html
- anomalous patterson
    - output.html (patt_ano)
- anomalous pattersons
    - analyze_mad.html
- anomalously
    - all_keywords.html (crystal_info)
    - all_keywords.html (scatter)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - auto_keywords.html
- anoonly
    - all_keywords.html (heavy_control)
- anorefine
    - all_keywords.html (heavy_control)
    - auto_keywords.html
- archive
    - FAQS.html (more)
- archived
    - FAQS.html (more)
- asdp
    - FAQS.html (more)

- asymmetric unit
  - all_keywords.html (crystal_info)
  - all_commands.html
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - auto_keywords.html
- atomname
  - all_keywords.html (atom_input)
  - analyze_mir.html (script-analyze_mir)
  - auto_keywords.html
  - generate.html
  - heavy.html (mad phasing)
- aval
  - FAQS.html (scatt_2)
  - all_keywords.html (scatter)
  - all_keywords.html (atom_input)
  - auto_keywords.html
  - heavy.html (mad phasing)
- aval_mad
  - madbst.html (keywords)
  - madmrg.html (keywords)
- avg_omit
  - all_commands.html
  - maps.html
  - maps.html (omit_map)
  - maps.html (avg_omit)
  - table-of-contents.html
- bayes
  - all_keywords.html (veryquick)
  - analyze_mad.html (keywords-analyze_mad)
  - madbst.html (how madbst works)
  - solve.html (solve_script_mir)
- bayesdiff
  - fdiff.html (script for fdiff)
- bayesian
  - FAQS.html (fom)
  - all_keywords.html (file_names)
  - all_keywords.html (veryquick)
  - all_keywords.html (heavy_control)

    - analyze_mad.html
- bbnml
    - FAQS.html (more)
- beta
    - FAQS.html (good)
    - all_keywords.html (crystal_info)
    - all_keywords.html (generate)
    - auto_keywords.html
    - fdiff.html
- bijvoet
    - scale_mad.html
- bin
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (beta-catenin)
    - notes/solve_brief_examples.html (gmcsf)
- bins
    - all_keywords.html (heavy_control)
- blue
    - FAQS.html (more)
- bmax
    - all_keywords.html (heavy_control)
- bmin
    - FAQS.html (bmin)
    - all_keywords.html (heavy_control)
    - new.html
- bnl
    - FAQS.html (more)
- boards
    - FAQS.html (more)
- boss
    - import_export.html (ffttoccp4)
    - maps.html (keywords for maps)
    - maps.html (peaksearch)
- bossfile
    - import_export.html (ffttoboss)
    - maps.html
    - maps.html (keywords for maps)
- bossgrid
    - all_keywords.html (grids)
    - import_export.html (ffttoccp4)

    - import_export.html (script for ffttomapview)
    - import_export.html (keywords for ffttomapview)
- btof
    - all_commands.html
    - import_export.html
    - import_export.html (btof-ftob)
- building
    - new.html
    - next.html
- bulletin
    - FAQS.html (more)
- button
    - FAQS.html (more)
- bval
    - FAQS.html (scatt_2)
    - all_keywords.html (scatter)
    - all_keywords.html (atom_input)
    - auto_keywords.html
    - heavy.html (mad phasing)
- bval_mad
    - madbst.html (keywords)
- bvalue
    - all_keywords.html (scatter)
    - all_keywords.html (atom_input)
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - generate.html
- cad
    - sample_scripts/ccp4_mir.html
    - sample_scripts/ccp4_mad.html
- cart
    - misc.html (#fract_to_cart)
- cart_to_fract
    - all_commands.html
    - misc.html
    - misc.html (#fract_to_cart)
    - table-of-contents.html
- cartesian
    - all_commands.html
    - misc.html

    - misc.html (#fract_to_cart)
    - table-of-contents.html
- catenin_y
    - notes/solve_brief_examples.html (beta-catenin)
- ccp4
    - FAQS.html (bigger)
    - FAQS.html (convert)
    - all_keywords.html (read_reject)
    - all_keywords.html (file_names)
    - all_commands.html
- ccp4_map
    - next.html
    - output.html
    - output.html (solve_ccp4)
- ccp4_open
    - auto.html (run)
    - intro.html (started)
    - intro.html (solvedir)
    - table-of-contents.html
    - notes/solve_brief_examples.html (gene 5 protein)
- ccp4mapfile
    - all_keywords.html (file_names)
    - import_export.html (ffttoccp4)
- ccx100
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (correlation)
    - notes/solve_brief_examples.html (gmcsf)
- cell
    - FAQS.html (cell_dim_error)
    - all_keywords.html
    - all_keywords.html (crystal_info)
    - all_keywords.html (scatter)
    - all_keywords.html (generate)
- cell_derivative
    - all_keywords.html (generate)
    - generate.html
- cells
    - hassp.html (more about hassp)
    - intro.html (symmetry)
    - maps.html (keywords for maps)

- centered
  - intro.html (symmetry)
- centering
  - intro.html (symmetry)
- centric
  - heavy.html (refinement)
  - heavy.html (statistical output)
  - heavy.html (correlated phasing)
  - misc.html (weights)
- centrosymmetric
  - all_keywords.html (solve_control)
  - auto_keywords.html
- changes
  - new.html
- checksolve
  - FAQS.html (checksolve)
  - all_keywords.html (file_names)
  - all_keywords.html (solve_control)
  - auto_keywords.html
  - generate.html
- clearly
  - FAQS.html (good)
- click
  - FAQS.html (more)
- closer
  - FAQS.html (close sites)
- clus_aval
  - all_keywords.html (scatter)
- clus_bval
  - all_keywords.html (scatter)
- clus_cval
  - all_keywords.html (scatter)
- clus_fp
  - all_keywords.html (scatter)
- clus_fp_aval
  - all_keywords.html (scatter)
- clus_fp_bval
  - all_keywords.html (scatter)
- clus_fp_cval
  - all_keywords.html (scatter)

- clus_fpp
  - all_keywords.html (scatter)
- clus_fpp_aval
  - all_keywords.html (scatter)
- clus_fpp_bval
  - all_keywords.html (scatter)
- clus_fpp_cval
  - all_keywords.html (scatter)
- cluster
  - all_keywords.html (scatter)
- clusters
  - auto_keywords.html
- cns
  - fdiff.html (script for fdiff)
  - formats.html (molecular replacement?)
- cols
  - filemerge.html (script)
  - localscale.html (merge)
- column
  - FAQS.html (more)
  - all_keywords.html
  - all_keywords.html (read_reject)
  - all_keywords.html (column_numbers)
  - all_keywords.html (heavy_control)
- columns
  - all_keywords.html (read_reject)
  - all_commands.html
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - analyze_mir.html
- combine_all
  - FAQS.html (combine_all)
- combine_all_data
  - FAQS.html (combine)
  - all_keywords.html (solve_control)
  - all_commands.html
  - combine.html
  - new.html
- combined
  - all_commands.html

    - analyze_mad.html (keywords-analyze_mad)
    - binary.html (dorgbn files)
    - combine.html
    - madbst.html (keywords)
- compare_soln
    - all_commands.html
    - misc.html
    - misc.html (compare_soln)
    - table-of-contents.html
- comparisonfile
    - FAQS.html (checksolve)
    - all_keywords.html (file_names)
    - generate.html
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (beta-catenin)
- compile
    - FAQS.html (bigger)
- complicated
    - next.html (addsolve)
- compound
    - all_keywords.html (scatter)
- compounds
    - all_keywords.html (scatter)
- compress
    - heavy.html (mad phasing)
- compressed
    - all_keywords.html (veryquick)
    - analyze_mad.html (keywords-analyze_mad)
    - solve.html
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- content
    - new.html
- continue_mad
    - all_keywords.html (veryquick)
- continue_sad
    - all_keywords.html (veryquick)
- contributing
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (correlation)

- conversion
  - import_export.html (btof-ftob)
  - import_export.html (ffttoboss)
  - import_export.html (script for ffttomapview)
  - madmrg.html
  - misc.html (script for math)
- conversions
  - misc.html (math)
- convert
  - FAQS.html (convert)
  - FAQS.html (look)
  - all_commands.html
  - binary.html (map)
  - heavy.html (mad phasing)
- coordinatefile
  - all_keywords.html (generate)
  - generate.html
- coordinates
  - all_keywords.html (atom_input)
  - all_keywords.html (heavy_control)
  - all_keywords.html (hassp)
  - all_keywords.html (generate)
  - all_commands.html
- coords
  - generate.html
  - maps.html (rho)
  - misc.html (#fract_to_cart)
- correlated
  - FAQS.html (fom)
  - all_keywords.html (atom_input)
  - all_keywords.html (heavy_control)
  - auto_keywords.html
  - combine.html
- correlation
  - FAQS.html (good)
  - FAQS.html (checksolve)
  - analyze_mad.html
  - analyze_mir.html
  - fdiff.html
- correlations

    - analyze_mad.html
    - analyze_mir.html
    - heavy.html (correlated phasing)
- correlphase
    - all_keywords.html (heavy_control)
    - heavy.html (correlated phasing)
- cross vector
    - all_keywords.html (hassp)
    - hassp.html (more about hassp)
- cross vectors
    - all_keywords.html (hassp)
    - different.html
    - hassp.html (more about hassp)
    - hassp.html (analyzing a solution)
    - notes/solve_brief_examples.html (gene 5 protein)
- cut_delete
    - all_keywords.html (veryquick)
- cutoff
    - different.html
    - misc.html (compare_soln)
- cutoff_deriv
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - different.html
- cval
    - FAQS.html (scatt_2)
    - all_keywords.html (scatter)
    - all_keywords.html (atom_input)
    - auto_keywords.html
    - heavy.html (mad phasing)
- cval_mad
    - madbst.html (keywords)
- damped
    - all_keywords.html (read_reject)
    - localscale.html (keywords for localscale)
- damping
    - all_keywords.html (read_reject)
    - all_keywords.html (heavy_control)
    - localscale.html (keywords for localscale)
- dash

    - intro.html (symmetry)
- data?
    - FAQS.html (bmin)
    - formats.html
- datafile
    - all_keywords.html (file_names)
    - all_keywords.html (veryquick)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - analyze_mir.html
- dataset
    - FAQS.html (more)
    - FAQS.html (good)
    - FAQS.html (checksolve)
    - all_keywords.html (read_reject)
    - all_keywords.html (file_names)
- dataset?
    - formats.html
- datasets
    - FAQS.html (combine_all)
    - all_commands.html
    - combine.html
    - different.html
    - fdiff.html
- dealt
    - new.html
- decay
    - FAQS.html (more)
- delano
    - all_keywords.html (file_names)
    - all_keywords.html (heavy_control)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - analyze_mir.html
- delanom
    - all_commands.html
    - scale_mir.html
- delfano
    - madbst.html (how madbst works)
- delhg

    - formats.html (ccp4 mtz)
- deliso
    - madmrg.html (notes)
- delmax
    - maps.html
    - maps.html (keywords for maps)
- denzo
    - formats.html (denzo/scalepack)
- depends
    - FAQS.html (close sites)
- deriv
    - all_keywords.html (column_numbers)
    - all_keywords.html (atom_input)
    - all_keywords.html (heavy_control)
    - analyze_mir.html (script-analyze_mir)
    - auto_keywords.html
- derivative
    - FAQS.html (good)
    - FAQS.html (input)
    - all_keywords.html (read_reject)
    - all_keywords.html (file_names)
    - all_keywords.html (solve_control)
- derivatives
    - all_keywords.html (atom_input)
    - all_keywords.html (heavy_control)
    - combine.html
    - different.html
    - generate.html
- derscale
    - all_keywords.html (atom_input)
- dertemp
    - all_keywords.html (atom_input)
- difference fourier
    - all_keywords.html (solve_control)
    - all_keywords.html (veryquick)
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - heavy.html
- difference fouriers
    - all_keywords.html (read_reject)

    - all_keywords.html (solve_control)
    - auto_keywords.html
    - formats.html (molecular replacement?)
    - how_solve_works.html
- difference patterson
    - hassp.html
    - hassp.html (more about hassp)
    - how_solve_works.html
    - how_solve_works.html (refine)
    - madbst.html (keywords)
- difference pattersons
    - analyze_mad.html
    - analyze_mir.html
    - scoring.html
- differencing
    - fdiff.html (script for fdiff)
- dimension
    - FAQS.html (cell_dim_error)
    - new.html
- dimensioning
    - new.html
- dimensions
    - all_commands.html
    - formats.html (ccp4 mtz)
    - how_solve_works.html
    - sample_scripts/ccp4_mir.html
- dismax
    - maps.html (maptoasym)
- dismin
    - maps.html (maptoasym)
- dispersive
    - FAQS.html (good)
    - analyze_mad.html
    - how_solve_works.html (refine)
    - madbst.html (keywords)
    - references.html
- dmax
    - all_keywords.html (crystal_info)
    - auto_keywords.html
    - import_export.html (import)

    - misc.html (math)
- dmin
    - all_keywords.html (crystal_info)
    - auto_keywords.html
    - heavy.html (correlated phasing)
    - import_export.html (import)
    - misc.html (math)
- dna
    - new.html
- dorgbn
    - all_commands.html
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - analyze_mir.html
    - analyze_mir.html (script-analyze_mir)
- doubleprime
    - new.html
    - notes/solve_brief_examples.html (gene 5 protein)
    - sample_scripts/ccp4_mad.html
- dph1
    - formats.html (ccp4 mtz)
    - sample_scripts/ccp4_mir.html
    - sample_scripts/ccp4_mad.html
- dph2
    - sample_scripts/ccp4_mir.html
    - sample_scripts/ccp4_mad.html
- dph3
    - sample_scripts/ccp4_mad.html
- drgtoxplor
    - all_keywords.html (column_numbers)
    - all_commands.html
    - import_export.html
    - import_export.html (dorgbn-to-xplor)
- drgtoxplr
    - import_export.html (dorgbn-to-xplor)
- ead
    - all_keywords.html (heavy_control)
- earlier
    - FAQS.html (bmin)
    - new.html

- eis
  - all_keywords.html (heavy_control)
- eod
  - notes/solve_brief_examples.html (gene 5 protein)
  - notes/solve_brief_examples.html (beta-catenin)
  - notes/solve_brief_examples.html (gmcsf)
  - sample_scripts/ccp4_mir.html
  - sample_scripts/ccp4_mad.html
- exec
  - FAQS.html (exec_error)
- exist
  - intro.html (solvedir)
- export
  - FAQS.html (convert)
  - all_keywords.html (file_names)
  - all_commands.html
  - binary.html (data)
  - import_export.html
- exported
  - binary.html (data)
- exportfile
  - all_keywords.html (file_names)
  - next.html (addsolve)
- exporting
  - binary.html (data)
  - binary.html (map)
  - import_export.html
  - intro.html (data_files)
  - output.html (export)
- ezd
  - FAQS.html (look)
  - all_keywords.html (file_names)
  - generate.html
  - import_export.html (ffttoboss)
  - import_export.html (ffttoccp4)
- ezd_draw
  - import_export.html (ffttoboss)
  - output.html (solve_ezd)
- ezdgrid
  - all_keywords.html (grids)

    - auto_keywords.html
    - import_export.html (ffttoboss)
- ezdmapfile
    - all_keywords.html (file_names)
    - import_export.html (ffttoboss)
- ezdnew
    - all_commands.html
- f2mtz
    - FAQS.html (convert)
- f_o
    - all_keywords.html (scatter)
- f_phi_to_a_b
    - misc.html (math)
- fails
    - FAQS.html (more)
    - FAQS.html (hand)
- faqs
    - new.html
    - table-of-contents.html
- fbar
    - all_keywords.html (read_reject)
    - all_keywords.html (file_names)
    - all_keywords.html (column_numbers)
    - all_commands.html
    - analyze_mad.html
- fc1
    - all_keywords.html (read_reject)
    - auto_keywords.html
- fcalc
    - all_keywords.html (column_numbers)
    - fdiff.html
    - fdiff.html (script for fdiff)
    - fdiff.html (keywords for fdiff)
    - misc.html (weights)
- fder
    - all_keywords.html (heavy_control)
    - binary.html (dorgbn files)
    - heavy.html (statistical output)
    - localscale.html (more on localscale)
    - madmrg.html (notes)

- fderiv
  - analyze_mad.html (keywords-analyze_mad)
  - binary.html (dorgbn files)
  - madmrg.html (notes)
- fdiff
  - all_keywords.html (column_numbers)
  - all_commands.html
  - fdiff.html
  - fdiff.html (script for fdiff)
  - fdiff.html (keywords for fdiff)
- fdoubleprime
  - all_keywords.html (scatter)
- fft
  - FAQS.html (checksolve)
  - all_keywords.html
  - all_keywords.html (file_names)
  - all_keywords.html (grids)
  - all_commands.html
- fftfile
  - all_keywords.html (file_names)
  - hassp.html (script for hassp)
  - import_export.html (ffttoboss)
  - import_export.html (ffttoccp4)
  - import_export.html (script for ffttomapview)
- fftgrid
  - all_keywords.html (hassp)
  - all_keywords.html (grids)
  - auto_keywords.html
  - hassp.html (analyzing a solution)
  - import_export.html (ffttoboss)
- ffttoboss
  - maps.html
  - maps.html (keywords for maps)
  - maps.html (keywords for peaksearch)
- ffttoccp4
  - all_commands.html
  - import_export.html (ffttoezd)
  - import_export.html (ffttoccp4)
- ffttoccp4is
  - import_export.html (ffttoccp4)

- ffttoezd
  - FAQS.html (look)
  - all_commands.html
  - binary.html (map)
  - import_export.html
  - import_export.html (ffttoezd)
- ffttomapview
  - all_commands.html
  - import_export.html
  - import_export.html (ffttoezd)
  - import_export.html (ffttoccp4)
  - import_export.html (script for ffttomapview)
- fhbar
  - all_keywords.html (heavy_control)
- fhg
  - formats.html (ccp4 mtz)
- filemerge
  - all_commands.html
  - fdiff.html
  - filemerge.html
  - filemerge.html (script)
  - import_export.html (import)
- fileout
  - filemerge.html (script)
- filetitle
  - all_keywords.html (file_names)
  - import_export.html (ffttoboss)
  - localscale.html (keywords for localscale)
- fixscattfactors
  - all_keywords.html (scatter)
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - auto_keywords.html
  - different.html
- fluorescence
  - FAQS.html (scatt_factors)
- fmin
  - all_keywords.html (heavy_control)
- fminu
  - all_keywords.html (column_numbers)

    - localscale.html (keywords for localscale)
- fminus
    - all_keywords.html (column_numbers)
    - madmrg.html (keywords)
- fminusfile
    - misc.html (math)
- fnat
    - all_keywords.html (heavy_control)
    - analyze_mir.html
    - analyze_mir.html (script-analyze_mir)
    - heavy.html (refinement)
    - heavy.html (statistical output)
- fnative
    - analyze_mad.html (keywords-analyze_mad)
    - binary.html (dorgbn files)
    - madmrg.html (notes)
    - output.html (solve_mtz)
    - output.html (phases-hl_export)
- fobs_sig_from_f_err
    - misc.html (math)
- fofc
    - maps.html
    - maps.html (keywords for maps)
    - misc.html (script for weights)
- fold
    - hassp.html (more about hassp)
- fommin
    - all_keywords.html (heavy_control)
- form_factor
    - all_keywords.html (scatter)
- formats
    - FAQS.html (two_formats)
    - binary.html
    - binary.html (map)
    - formats.html
    - import_export.html
- formatted
    - all_keywords.html (file_names)
    - all_commands.html
    - auto.html (run)

    - formats.html
    - formats.html (molecular replacement?)
- formatting
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - scale_mir.html
    - scale_native.html (script-scale_native)
- formula
    - all_keywords.html (scatter)
- formulas
    - all_keywords.html (scatter)
- fourcofile
    - maps.html (keywords for maps)
- fourier
    - all_keywords.html (solve_control)
    - all_keywords.html (veryquick)
    - all_keywords.html (heavy_control)
    - all_keywords.html (maps)
    - all_commands.html
- fp_or_fm
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - different.html
    - misc.html (getanom)
    - scale_mir.html
- fpfm_only
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - different.html
    - misc.html (getanom)
    - scale_mir.html
- fph1
    - formats.html (ccp4 mtz)
    - sample_scripts/ccp4_mir.html
    - sample_scripts/ccp4_mad.html
- fph2
    - formats.html (ccp4 mtz)
    - sample_scripts/ccp4_mir.html
    - sample_scripts/ccp4_mad.html
- fph3

    - sample_scripts/ccp4_mad.html
- fphbar
    - all_keywords.html (heavy_control)
- fplus
    - all_keywords.html (column_numbers)
    - localscale.html (keywords for localscale)
    - madmrg.html (keywords)
- fplusfile
    - misc.html (math)
- fprime
    - all_keywords.html (scatter)
- fprimv
    - FAQS.html (scatt_2)
    - all_keywords.html (scatter)
    - analyze_mir.html (script-analyze_mir)
    - auto_keywords.html
    - heavy.html (mad phasing)
- fprimv_mad
    - all_keywords.html (scatter)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - auto_keywords.html
    - generate.html
- fprprv
    - FAQS.html (scatt_2)
    - all_keywords.html (scatter)
    - analyze_mir.html (script-analyze_mir)
    - auto_keywords.html
    - heavy.html (mad phasing)
- fprprv_mad
    - all_keywords.html (scatter)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - auto_keywords.html
    - generate.html
- fract_to_cart
    - all_commands.html
    - misc.html
    - misc.html (#fract_to_cart)

    - table-of-contents.html
- fractional
    - all_keywords.html (atom_input)
    - all_keywords.html (hassp)
    - all_commands.html
    - auto_keywords.html
    - maps.html (keywords for maps)
- fragment
    - new.html
- freedom
    - new.html
- fsigmin
    - all_keywords.html (heavy_control)
- full_inverse
    - all_keywords.html (solve_control)
- generate
    - FAQS.html (checksolve)
    - all_keywords.html
    - all_keywords.html (generate)
    - all_commands.html
    - auto.html (run)
- generate_mad
    - all_commands.html
    - generate.html
- generate_mir
    - all_keywords.html (generate)
    - all_commands.html
    - generate.html
- generated
    - all_keywords.html (solve_control)
    - auto_keywords.html
    - fdiff.html
    - generate.html
    - heavy.html
- genf_phi
    - misc.html (math)
- gensolvemad
    - generate.html
- gensolvemir
    - generate.html

- getanom
  - all_commands.html
  - misc.html
  - misc.html (getanom)
  - table-of-contents.html
- getgroups
  - all_keywords.html (atom_input)
  - all_keywords.html (heavy_control)
  - heavy.html (correlated phasing)
- getiso
  - all_commands.html
  - maps.html (script for maps)
  - misc.html
  - misc.html (getiso)
  - table-of-contents.html
- getphases
  - all_commands.html
  - misc.html
  - misc.html (getphases)
  - table-of-contents.html
- gm_offset
  - notes/solve_brief_examples.html (gmcsf)
- gotoatom
  - all_keywords.html (atom_input)
  - heavy.html (goto)
  - solve.html (solve_script_mir)
- gotoder
  - generate.html
- gotoderiv
  - all_keywords.html (atom_input)
  - analyze_mir.html
  - auto_keywords.html
  - heavy.html (goto)
  - solve.html (solve_script_mir)
- ha_pdb
  - all_commands.html
  - misc.html
  - misc.html (ha_pdb)
  - table-of-contents.html
- hand

    - FAQS.html (checksolve)
    - FAQS.html (hand)
    - intro.html
    - solve.html
- handed
    - misc.html (#fract_to_cart)
- handedness
    - misc.html (compare_soln)
    - notes/solve_brief_examples.html
- handy
    - FAQS.html (checksolve)
- hang
    - new.html
- hard
    - FAQS.html (all_data)
- harker
    - hassp.html (more about hassp)
- hassp
    - all_keywords.html
    - all_keywords.html (hassp)
    - all_commands.html
    - hassp.html
    - hassp.html (script for hassp)
- heap
    - FAQS.html (sbin_loader)
- heard
    - FAQS.html (scatt_2)
- heavy
    - FAQS.html (hand)
    - FAQS.html (bmin)
    - all_keywords.html
    - all_keywords.html (control)
    - all_keywords.html (crystal_info)
- histograms
    - new.html
- hkl_view
    - intro.html (data_files)
- hklin
    - all_keywords.html (read_reject)
    - auto_keywords.html

    - formats.html (ccp4 mtz)
    - sample_scripts/script_list.html
    - sample_scripts/ccp4_mir.html
- hla
    - output.html (solve_mtz)
- hlb
    - output.html (solve_mtz)
- hlc
    - output.html (solve_mtz)
- hld
    - output.html (solve_mtz)
- hopefully
    - FAQS.html (more)
- http
    - FAQS.html (more)
- i_to_f
    - localscale.html (more on merge)
    - misc.html (math)
- iangle
    - all_keywords.html (heavy_control)
- icol
    - filemerge.html
- icrmax
    - all_keywords.html (hassp)
    - different.html
- identification
    - new.html
- identifying
    - FAQS.html (close sites)
- iegroup
    - all_keywords.html (atom_input)
    - heavy.html (correlated phasing)
- ignore
    - all_keywords.html (hassp)
    - intro.html (commands)
- ignored
    - all_keywords.html (veryquick)
    - all_keywords.html (atom_input)
    - auto_keywords.html
    - import_export.html (import)

    - intro.html (commands)
- ignoring
    - all_keywords.html (hassp)
- ihassplist
    - all_keywords.html (hassp)
- ihassptype
    - all_keywords.html (hassp)
- iheavyproc
    - all_keywords.html (heavy_control)
    - heavy.html (typical cycle)
- ikeepflag
    - localscale.html (merge)
- imadphase
    - all_keywords.html (heavy_control)
    - heavy.html (mad phasing)
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- import
    - FAQS.html (convert)
    - all_commands.html
    - binary.html (data)
    - fdiff.html
    - import_export.html
- imported
    - binary.html (data)
- importing
    - binary.html (data)
    - binary.html (map)
    - import_export.html
    - import_export.html (import)
    - intro.html (data_files)
- improves
    - new.html
- improving
    - new.html
- imsb
    - next.html
- inall
    - notes/solve_brief_examples.html (gmcsf)
- inanal

    - all_keywords.html (heavy_control)
- inano
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - different.html
    - generate.html
    - misc.html (compare_soln)
- incorporates
    - new.html
- increased
    - new.html
- infile
    - all_keywords.html (file_names)
    - fdiff.html (script for fdiff)
    - fdiff.html (keywords for fdiff)
    - import_export.html (scripts for import)
    - import_export.html (export)
- inhend
    - all_keywords.html (heavy_control)
- inold
    - all_keywords.html (heavy_control)
    - heavy.html (typical cycle)
- inosig
    - all_keywords.html (heavy_control)
- inphas
    - all_keywords.html (heavy_control)
    - heavy.html (typical cycle)
- inphase
    - all_keywords.html (heavy_control)
    - analyze_mad.html (keywords-analyze_mad)
    - auto_keywords.html
    - heavy.html (typical cycle)
    - solve.html (solve_script_mad)
- inprnt
    - all_keywords.html (heavy_control)
- inresd
    - all_keywords.html (heavy_control)
    - heavy.html (typical cycle)
- installation
    - intro.html (symmetry)

- intensities
  - all_keywords.html (read_reject)
  - all_keywords.html (generate)
  - auto_keywords.html
  - formats.html
  - formats.html (free-format i)
- intensities?
  - formats.html
- interference
  - FAQS.html (all_data)
- internally
  - all_keywords.html (heavy_control)
- introduces
  - new.html
- inverse
  - FAQS.html (hand)
  - all_keywords.html (solve_control)
  - auto_keywords.html
  - misc.html (math)
  - new.html
- inverses
  - all_keywords.html (solve_control)
  - auto_keywords.html
- inversion
  - misc.html (compare_soln)
- inversions
  - how_solve_works.html (speed)
- irefcy
  - all_keywords.html (heavy_control)
  - heavy.html (typical cycle)
- isignf
  - hassp.html (more about hassp)
- isoano
  - all_keywords.html (heavy_control)
- isofourier
  - maps.html
  - maps.html (keywords for maps)
- isomorphism
  - combine.html
  - generate.html

    - heavy.html (statistical output)
    - heavy.html (correlated phasing)
- isomorphisms
    - combine.html
    - how_solve_works.html (refine)
- isomorphous
    - all_keywords.html (heavy_control)
    - analyze_mir.html
    - different.html
    - heavy.html (statistical output)
    - madmrg.html
- isoonly
    - all_keywords.html (heavy_control)
- isymmetry
    - maps.html (keywords for maps)
    - maps.html (keywords for peaksearch)
- iterative
    - new.html
- jalt
    - all_keywords.html (heavy_control)
    - heavy.html (refinement)
    - heavy.html (typical cycle)
- jcol
    - filemerge.html
- jiang
    - FAQS.html (more)
- jiansheng
    - FAQS.html (more)
- jones
    - next.html
- jstd
    - all_keywords.html (column_numbers)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - madmrg.html (keywords)
    - solve.html (solve_script_mad)
- kalt
    - all_keywords.html (heavy_control)
    - heavy.html (refinement)
    - heavy.html (typical cycle)

- kder
  - all_keywords.html (heavy_control)
  - heavy.html (typical cycle)
- keepall
  - all_keywords.html (read_reject)
  - localscale.html (keywords for localscale)
  - localscale.html (keywords for merge)
- kleywegt
  - generate.html
- kout
  - all_keywords.html (file_names)
  - all_keywords.html (heavy_control)
  - heavy.html (typical cycle)
- kramers
  - FAQS.html (scatt_factors)
- kronig
  - FAQS.html (scatt_factors)
- label
  - all_keywords.html (atom_input)
  - analyze_mad.html
  - analyze_mir.html (script-analyze_mir)
  - madmrg.html (keywords)
  - solve.html (solve_script_mad)
- labelled
  - all_keywords.html (hassp)
- labels
  - formats.html (ccp4_unmerged)
- labin
  - all_keywords.html (read_reject)
  - auto_keywords.html
  - formats.html (ccp4 mtz)
  - formats.html (ccp4_unmerged)
  - sample_scripts/script_list.html
- lambda
  - all_keywords.html (scatter)
  - all_keywords.html (atom_input)
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - auto_keywords.html
- lattice

    - misc.html (math)
- letters
    - formats.html (ccp4 mtz)
- libccp4
    - FAQS.html (bigger)
- library
    - FAQS.html (bigger)
    - intro.html (solvedir)
- limitation
    - FAQS.html (combine)
- localscale
    - all_keywords.html (read_reject)
    - all_commands.html
    - heavy.html (rejecting data)
    - import_export.html (import)
    - localscale.html
- localscaling
    - how_solve_works.html
    - how_solve_works.html (refine)
    - localscale.html (script for localscale)
- logfile
    - all_keywords.html (control)
    - all_keywords.html (file_names)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - analyze_mir.html (script-analyze_mir)
- login_custom
    - auto.html (run)
- lowercase
    - formats.html (ccp4 mtz)
- lsort
    - import_export.html (import)
- lswap
    - import_export.html (import)
- machine
    - FAQS.html (exec_error)
    - import_export.html (ffttoccp4)
    - intro.html (solvedir)
- machines
    - sample_scripts/script_list.html

- mad
  - FAQS.html (more)
  - FAQS.html (good)
  - FAQS.html (hand)
  - FAQS.html (scatt_factors)
  - FAQS.html (fom)
- mad_atom
  - all_keywords.html (scatter)
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - auto_keywords.html
  - generate.html
- mad_atomname
  - notes/solve_brief_examples.html (gene 5 protein)
- mad_fbar
  - FAQS.html (convert)
  - all_keywords.html (file_names)
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - binary.html (dorgbn files)
- mad_fpfm
  - all_keywords.html (file_names)
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - binary.html (dorgbn files)
  - scale_mad.html
- madatom
  - all_keywords.html (atom_input)
- madbst
  - all_keywords.html (file_names)
  - all_commands.html
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - binary.html (dorgbn files)
- madbstfile
  - all_keywords.html (file_names)
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - binary.html (dorgbn files)
- madfbarfile

    - all_keywords.html (file_names)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - binary.html (dorgbn files)
    - scale_mad.html
- madfpfmfile
    - all_keywords.html (file_names)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - binary.html (dorgbn files)
    - scale_mad.html
- madmrg
    - all_keywords.html (file_names)
    - all_keywords.html (veryquick)
    - all_commands.html
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
- madmrgfile
    - all_keywords.html (file_names)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - binary.html (dorgbn files)
- madphase
    - all_keywords.html (heavy_control)
    - solve.html (solve_script_mad)
- mailing
    - FAQS.html (more)
    - intro.html (help)
- mapman
    - FAQS.html (look)
    - binary.html (map)
    - generate.html
    - output.html (solve_ezd)
    - output.html (patt_ano)
- maps
    - FAQS.html (look)
    - all_keywords.html
    - all_keywords.html (solve_control)
    - all_keywords.html (heavy_control)

    - all_keywords.html (maps)
- maptoasym
    - all_commands.html
    - maps.html
    - maps.html (maptoasym)
    - table-of-contents.html
- maptoobject
    - all_commands.html
    - maps.html (maptoasym)
- mapview
    - all_keywords.html (file_names)
    - all_keywords.html (grids)
    - all_commands.html
    - import_export.html
    - import_export.html (ffttoccp4)
- mapviewfile
    - all_keywords.html (file_names)
    - import_export.html (script for ffttomapview)
    - import_export.html (keywords for ffttomapview)
    - maps.html
    - maps.html (keywords for maps)
- matters
    - formats.html (ccp4 mtz)
- mediumquick
    - how_solve_works.html (speed)
- memory
    - FAQS.html (sbin_loader)
- mentioned
    - all_keywords.html (scatter)
- merge
    - FAQS.html (nomerge)
    - all_keywords.html (read_reject)
    - all_commands.html
    - formats.html
    - heavy.html (typical cycle)
- merged
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - formats.html
    - formats.html (denzo/scalepack)

    - import_export.html (import)
- merging
    - filemerge.html
    - localscale.html
    - localscale.html (merge)
    - localscale.html (keywords for merge)
    - table-of-contents.html
- meritx100
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (correlation)
    - notes/solve_brief_examples.html (gmcsf)
- mhz
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (beta-catenin)
    - notes/solve_brief_examples.html (gmcsf)
- min
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (correlation)
    - notes/solve_brief_examples.html (gmcsf)
- minor
    - new.html
- mir
    - FAQS.html (good)
    - all_keywords.html (atom_input)
    - all_keywords.html (heavy_control)
    - all_commands.html
    - analyze_mir.html
- mir_fbar
    - analyze_mir.html
    - analyze_mir.html (script-analyze_mir)
    - scale_mir.html
    - solve.html (solve_script_mir)
    - sample_scripts/ccp4_mir.html
- mir_fpfm
    - scale_mir.html
- mirfbarfile
    - analyze_mir.html (script-analyze_mir)
    - scale_mir.html
- mirfpfmfile
    - scale_mir.html

- molecular
  - formats.html (molecular replacement?)
- mosflm
  - intro.html (data_files)
- mtz
  - FAQS.html (convert)
  - FAQS.html (combine)
  - all_keywords.html (read_reject)
  - all_keywords.html (solve_control)
  - auto_keywords.html
- multiplies
  - all_keywords.html (scatter)
- mutually
  - misc.html (#fract_to_cart)
- myfc
  - formats.html (molecular replacement?)
- myfom
  - formats.html (molecular replacement?)
- n_f_bins
  - all_keywords.html (heavy_control)
- n_refl
  - new.html
- nanof
  - all_keywords.html (column_numbers)
  - localscale.html (keywords for localscale)
- nanomalous
  - all_keywords.html (crystal_info)
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - auto_keywords.html
  - different.html
- nanos
  - all_keywords.html (column_numbers)
  - localscale.html (keywords for localscale)
- natfourier
  - maps.html
  - maps.html (keywords for maps)
  - notes/solve_brief_examples.html (gene 5 protein)
  - notes/solve_brief_examples.html (correlation)
  - notes/solve_brief_examples.html (gmcsf)

- native
  - FAQS.html (good)
  - all_keywords.html (read_reject)
  - all_keywords.html (file_names)
  - all_keywords.html (column_numbers)
  - all_keywords.html (solve_control)
- native_f
  - scale_mir.html
  - scale_native.html
  - scale_native.html (script-scale_native)
- nbst
  - all_keywords.html (heavy_control)
- ncol
  - binary.html (dorgbn files)
  - import_export.html (dorgbn files)
  - madbst.html (output)
- ncola
  - misc.html (math)
- ncolb
  - misc.html (math)
- ncoldelf
  - all_keywords.html (column_numbers)
  - madbst.html (keywords)
  - madmrg.html (keywords)
  - solve.html (solve_script_mad)
  - solve.html (solve_script_mir)
- ncolf
  - maps.html (keywords for maps)
  - misc.html (math)
- ncolf_merge
  - all_keywords.html (column_numbers)
  - localscale.html (merge)
  - localscale.html (keywords for merge)
- ncolfa
  - maps.html (keywords for maps)
  - misc.html (getphases)
- ncolfb
  - maps.html (keywords for maps)
  - misc.html (getphases)
- ncolfbar

    - all_keywords.html (column_numbers)
    - madbst.html (keywords)
    - madmrg.html (keywords)
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- ncolfc
    - all_keywords.html (column_numbers)
    - fdiff.html (script for fdiff)
    - fdiff.html (keywords for fdiff)
    - maps.html (keywords for maps)
    - misc.html (script for weights)
- ncolfcmut
    - all_keywords.html (column_numbers)
    - fdiff.html (keywords for fdiff)
- ncolfhcos
    - madbst.html (keywords)
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- ncolfhsin
    - madbst.html (keywords)
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- ncolfigm
    - maps.html
- ncolfm
    - misc.html (getanom)
- ncolfminus
    - all_keywords.html (column_numbers)
    - heavy.html (mad phasing)
    - solve.html (solve_script_mad)
- ncolfobs
    - maps.html (keywords for maps)
- ncolfomut
    - all_keywords.html (column_numbers)
    - fdiff.html (script for fdiff)
    - fdiff.html (keywords for fdiff)
- ncolfowt
    - all_keywords.html (column_numbers)
    - fdiff.html (script for fdiff)
    - fdiff.html (keywords for fdiff)

    - misc.html (script for weights)
    - misc.html (keywords for weights)
- ncolfp
    - misc.html (getanom)
- ncolfplus
    - all_keywords.html (column_numbers)
    - heavy.html (mad phasing)
    - solve.html (solve_script_mad)
- ncoli
    - misc.html (math)
- ncolpatt
    - maps.html (keywords for maps)
    - maps.html (script for maps)
- ncolpattsig
    - maps.html
    - maps.html (keywords for maps)
- ncolphi
    - maps.html (keywords for maps)
    - misc.html (math)
- ncolrtest
    - all_keywords.html (column_numbers)
    - fdiff.html (script for fdiff)
    - fdiff.html (keywords for fdiff)
    - misc.html (script for weights)
    - misc.html (keywords for weights)
- ncolsdelf
    - all_keywords.html (column_numbers)
    - madbst.html (keywords)
    - madmrg.html (keywords)
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- ncolsfbar
    - all_keywords.html (column_numbers)
    - madbst.html (keywords)
    - madmrg.html (keywords)
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- ncolsfm
    - misc.html (getanom)
- ncolsfomut

    - fdiff.html (script for fdiff)
- ncolsfp
    - misc.html (getanom)
- ncolsig
    - misc.html (math)
- ncolsig_merge
    - all_keywords.html (column_numbers)
    - localscale.html (merge)
    - localscale.html (keywords for merge)
- ncolsigi
    - misc.html (math)
- ncolsigminus
    - all_keywords.html (column_numbers)
    - heavy.html (mad phasing)
    - solve.html (solve_script_mad)
- ncolsigplus
    - all_keywords.html (column_numbers)
    - heavy.html (mad phasing)
    - solve.html (solve_script_mad)
- ncolsmut
    - all_keywords.html (column_numbers)
    - fdiff.html (keywords for fdiff)
- ncolswt
    - all_keywords.html (column_numbers)
    - fdiff.html (script for fdiff)
    - fdiff.html (keywords for fdiff)
    - misc.html (script for weights)
    - misc.html (keywords for weights)
- ncs
    - FAQS.html (ncs)
- ncycle
    - all_keywords.html (heavy_control)
    - heavy.html (typical cycle)
- nderf
    - all_keywords.html (column_numbers)
    - localscale.html (script for localscale)
    - localscale.html (keywords for localscale)
    - maps.html (script for maps)
    - misc.html (getiso)
- nders

    - all_keywords.html (column_numbers)
    - localscale.html (script for localscale)
    - localscale.html (keywords for localscale)
    - maps.html (script for maps)
    - misc.html (getiso)
- negativeonly
    - maps.html (keywords for peaksearch)
- nequiv
    - hassp.html (more about hassp)
- nequiv_separate
    - misc.html (math)
- new_dataset
    - FAQS.html (combine_all)
    - all_keywords.html (solve_control)
    - all_commands.html
    - combine.html
- newatomtype
    - FAQS.html (scatt_2)
    - all_keywords.html (scatter)
    - all_keywords.html (atom_input)
    - analyze_mad.html (keywords-analyze_mad)
    - analyze_mir.html (script-analyze_mir)
- newezd
    - all_keywords.html (grids)
    - auto.html (run)
    - auto_keywords.html
    - binary.html (map)
    - generate.html
- newfile
    - all_keywords.html (file_names)
    - heavy.html
- newscriptfile
    - all_keywords.html (file_names)
    - next.html (addsolve)
- newsgroup?
    - FAQS.html (more)
- nfigm
    - all_keywords.html (heavy_control)
    - maps.html (keywords for maps)
- nfile

    - filemerge.html
- nfiles
    - localscale.html (keywords for merge)
- nice
    - FAQS.html (more)
    - FAQS.html (all_data)
- nlist
    - intro.html (commands)
    - intro.html (data_files)
    - maps.html (keywords for maps)
    - maps.html (keywords for peaksearch)
- nmaps
    - maps.html (avg_omit)
- nmp
    - all_keywords.html (heavy_control)
- nnatf
    - all_keywords.html (column_numbers)
    - all_keywords.html (heavy_control)
    - localscale.html (script for localscale)
    - localscale.html (keywords for localscale)
    - localscale.html (complete)
- nnatf_deriv
    - all_keywords.html (heavy_control)
- nnats
    - all_keywords.html (column_numbers)
    - all_keywords.html (heavy_control)
    - localscale.html (script for localscale)
    - localscale.html (keywords for localscale)
    - localscale.html (complete)
- nnats_deriv
    - all_keywords.html (heavy_control)
- no_anisotropic_b
    - all_keywords.html (veryquick)
- no_duplicate_sites
    - all_keywords.html (veryquick)
- no_fom
    - all_keywords.html (solve_control)
- no_fourier
    - all_keywords.html (solve_control)
- no_native_fourier

    - all_keywords.html (solve_control)
- no_patterson
    - all_keywords.html (solve_control)
- no_sim
    - all_keywords.html (heavy_control)
    - auto_keywords.html
- no_use_f_bins
    - all_keywords.html (heavy_control)
- noanorefine
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - different.html
    - notes/solve_brief_examples.html (gmcsf)
    - sample_scripts/ccp4_mir.html
- nobayes
    - all_keywords.html (veryquick)
    - analyze_mad.html (keywords-analyze_mad)
    - auto_keywords.html
    - heavy.html (statistical output)
- nobfactor
    - all_keywords.html (read_reject)
    - localscale.html (keywords for localscale)
- nodamping
    - all_keywords.html (read_reject)
    - localscale.html (keywords for localscale)
- noinano
    - different.html
    - scale_mir.html
- noinphase
    - all_keywords.html (heavy_control)
    - auto_keywords.html
- nolog
    - all_commands.html
- none
    - all_keywords.html (heavy_control)
    - generate.html
- nooverallscale
    - all_keywords.html (read_reject)
    - localscale.html (keywords for localscale)
- norefinescale

    - all_keywords.html (heavy_control)
    - madmrg.html (notes)
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- normalized
    - all_keywords.html (scatter)
    - status.html
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (gmcsf)
- nrep
    - all_keywords.html (heavy_control)
    - heavy.html (typical cycle)
- nres
    - all_keywords.html (crystal_info)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - auto_keywords.html
    - different.html
- nseedsolve
    - all_keywords.html (solve_control)
    - all_keywords.html (veryquick)
    - auto_keywords.html
    - different.html
    - intro.html (commands)
- nseedtest
    - all_keywords.html (solve_control)
    - auto_keywords.html
- nset
    - localscale.html (merge)
- nshells
    - all_keywords.html (control)
    - analyze_mad.html (keywords-analyze_mad)
    - analyze_mir.html (script-analyze_mir)
    - localscale.html (keywords for localscale)
    - localscale.html (keywords for merge)
- nsignf
    - all_keywords.html (hassp)
- nskip
    - all_keywords.html (read_reject)
    - auto_keywords.html

    - scale_mir.html
    - scale_native.html (script-scale_native)
- nsolsite
    - all_keywords.html (solve_control)
    - auto_keywords.html
    - different.html
    - generate.html
    - next.html (addsolve)
- nsolsite_deriv
    - all_keywords.html (solve_control)
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - different.html
    - notes/solve_brief_examples.html (gmcsf)
- nsym
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - hassp.html (more about hassp)
    - scale_mir.html
    - scale_native.html (script-scale_native)
- ntol_site
    - FAQS.html (close sites)
    - all_keywords.html (veryquick)
    - auto_keywords.html
- ntol_soln
    - all_keywords.html (veryquick)
    - auto_keywords.html
- ntopderiv
    - all_keywords.html (solve_control)
    - auto_keywords.html
    - intro.html (commands)
- ntopfour
    - all_keywords.html (solve_control)
    - auto_keywords.html
- ntophassp
    - all_keywords.html (hassp)
- ntopsolve
    - all_keywords.html (solve_control)
    - all_keywords.html (veryquick)

    - auto_keywords.html
    - different.html
    - intro.html (commands)
- numbr
    - hassp.html (more about hassp)
- nxplorf
    - all_keywords.html (column_numbers)
    - import_export.html (dorgbn-to-xplor)
- nxplorsig
    - all_keywords.html (column_numbers)
    - import_export.html (dorgbn-to-xplor)
- occ
    - all_keywords.html (atom_input)
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - generate.html
    - heavy.html (typical cycle)
- occup
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (correlation)
    - notes/solve_brief_examples.html (gmcsf)
- occupancies
    - different.html
    - heavy.html (mad phasing)
    - heavy.html (refinement)
    - heavy.html (typical cycle)
    - how_solve_works.html (refine)
- occupancy
    - all_keywords.html (atom_input)
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - generate.html
    - heavy.html (typical cycle)
- offset
    - notes/solve_brief_examples.html (beta-catenin)
- omit
    - all_commands.html
    - maps.html (omit_map)
    - maps.html (avg_omit)
    - table-of-contents.html

- omit_map
  - maps.html
  - maps.html (omit_map)
  - maps.html (avg_omit)
  - table-of-contents.html
- optimal
  - new.html
- origin
  - FAQS.html (checksolve)
  - all_keywords.html (heavy_control)
  - analyze_mad.html
  - analyze_mir.html
  - generate.html
- origin_removed
  - maps.html
- origin_removed_patt
  - maps.html (keywords for maps)
- origins
  - heavy.html (typical cycle)
- orthogonal
  - all_keywords.html (atom_input)
  - auto_keywords.html
- outfile
  - all_keywords.html (file_names)
  - fdiff.html (script for fdiff)
  - fdiff.html (keywords for fdiff)
  - import_export.html (export)
  - import_export.html (btof-ftob)
- outliers
  - localscale.html (merge)
- overallscale
  - all_keywords.html (read_reject)
  - auto_keywords.html
  - localscale.html (keywords for localscale)
  - scale_mir.html
  - scale_native.html (script-scale_native)
- overestimated
  - FAQS.html (good)
- overwriting
  - intro.html (solvedir)

    - sample_scripts/ccp4_mir.html
- p2_m
    - intro.html (symmetry)
- paper
    - all_keywords.html (scatter)
    - references.html
- partially
    - new.html
- patpk
    - all_keywords.html (hassp)
- patt_ano_orig_removed_der1
    - output.html
    - output.html (patt_ano)
- patt_fa
    - output.html
    - output.html (patt_ano)
- patt_iso_orig_removed_der1
    - output.html
    - output.html (patt_ano)
- pattern
    - new.html
- patterns
    - new.html
- patterson
    - FAQS.html (look)
    - all_keywords.html (file_names)
    - all_keywords.html (solve_control)
    - all_keywords.html (veryquick)
    - all_keywords.html (heavy_control)
- pattersons
    - all_keywords.html (solve_control)
    - analyze_mad.html
    - analyze_mir.html
    - analyze_mir.html (script-analyze_mir)
    - auto_keywords.html
- pattfftfile
    - madbst.html (keywords)
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- pattgrid

    - all_keywords.html (grids)
    - auto_keywords.html
    - import_export.html (ffttoboss)
    - import_export.html (keywords for ffttomapview)
- pdb
    - all_keywords.html (atom_input)
    - all_keywords.html (generate)
    - all_commands.html
    - auto.html (run)
    - auto_keywords.html
- pdb_xyz_in
    - all_keywords.html (atom_input)
    - auto_keywords.html
    - misc.html (compare_soln)
- peakfile
    - maps.html
    - maps.html (keywords for maps)
    - maps.html (script for peaksearch)
    - maps.html (keywords for peaksearch)
- peaksearch
    - all_commands.html
    - heavy.html (typical cycle)
    - maps.html
    - maps.html (peaksearch)
    - maps.html (script for peaksearch)
- peaksearching
    - table-of-contents.html
- percent_error
    - all_keywords.html (generate)
    - generate.html
- phasefile
    - all_keywords.html (file_names)
    - next.html (addsolve)
- phases
    - FAQS.html (input)
    - FAQS.html (fom)
    - all_keywords.html (read_reject)
    - all_keywords.html (file_names)
    - all_keywords.html (veryquick)
- phases_formatted

    - all_keywords.html (read_reject)
    - auto_keywords.html
    - formats.html (molecular replacement?)
- phases_labin
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - formats.html (molecular replacement?)
- phases_mtz
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - formats.html (molecular replacement?)
- phasing
    - FAQS.html (fom)
    - all_keywords.html
    - all_keywords.html (crystal_info)
    - all_keywords.html (veryquick)
    - all_keywords.html (heavy_control)
- phib
    - output.html (solve_mtz)
- phibest
    - all_keywords.html (heavy_control)
- phimostprobable
    - all_keywords.html (heavy_control)
- phosphorous
    - new.html
- plot_formfactors
    - all_keywords.html (scatter)
- positiveonly
    - maps.html (keywords for peaksearch)
- prefer
    - intro.html (scripts)
- preferences
    - all_commands.html
- premerged
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - formats.html
    - formats.html (denzo/scalepack)
    - formats.html (free-format i)
- preserve

    - all_keywords.html (scatter)
- prevented
    - new.html
- preventing
    - new.html
- prime
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (correlation)
- printout
    - all_keywords.html (heavy_control)
- progress
    - notes/solve_brief_examples.html
- properly
    - FAQS.html (bmin)
    - sample_scripts/script_list.html
- proportion
    - new.html
- quantities
    - all_keywords.html (scatter)
- questions
    - FAQS.html
    - FAQS.html (more)
- quicker
    - all_keywords.html (solve_control)
- r12000
    - FAQS.html (exec_error)
- r5000
    - FAQS.html (exec_error)
- rare
    - FAQS.html (hand)
- ratio
    - all_keywords.html (read_reject)
    - all_keywords.html (heavy_control)
    - all_keywords.html (hassp)
    - analyze_mad.html
    - auto_keywords.html
- ratmin
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - different.html

    - fdiff.html
    - fdiff.html (keywords for fdiff)
- rawderivfile
    - all_keywords.html (file_names)
    - auto_keywords.html
    - generate.html
    - scale_mir.html
    - notes/solve_brief_examples.html (gmcsf)
- rawmadfile
    - all_keywords.html (file_names)
    - auto_keywords.html
    - generate.html
    - scale_mad.html
    - notes/solve_brief_examples.html (gene 5 protein)
- rawmadfiles
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - scale_mir.html
    - scale_native.html (script-scale_native)
- rawmirfile
    - scale_mir.html
- rawnativefile
    - all_keywords.html (file_names)
    - auto_keywords.html
    - generate.html
    - scale_mir.html
    - scale_native.html (script-scale_native)
- read_amplitudes
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - formats.html (free-format i)
    - scale_mad.html
    - scale_mir.html
- read_intensities
    - formats.html (denzo/scalepack)
    - formats.html (free-format i)
    - scale_mir.html
    - scale_native.html (script-scale_native)
    - notes/solve_brief_examples.html (gene 5 protein)

- read_intensitites
  - all_keywords.html (read_reject)
  - auto_keywords.html
- readccp4_unmerged
  - all_keywords.html (read_reject)
  - formats.html (ccp4_unmerged)
  - notes/solve_brief_examples.html (beta-catenin)
  - sample_scripts/ccp4_mad.html
- readdenzo
  - all_keywords.html (read_reject)
  - auto_keywords.html
  - formats.html (denzo/scalepack)
  - formats.html (ccp4_unmerged)
  - scale_mir.html
- readformatted
  - FAQS.html (combine)
  - all_keywords.html (read_reject)
  - auto_keywords.html
  - combine.html
  - formats.html (free-format i)
- readtrek
  - all_keywords.html (read_reject)
  - auto_keywords.html
  - formats.html (ccp4_unmerged)
  - formats.html (d*trek)
  - notes/solve_brief_examples.html (gene 5 protein)
- rebuilding
  - new.html
- recalculated
  - new.html
- redefine
  - all_keywords.html (read_reject)
- references
  - references.html
  - table-of-contents.html
- refine
  - FAQS.html (input)
  - FAQS.html (bmin)
  - all_keywords.html (scatter)
  - all_keywords.html (heavy_control)

    - all_commands.html
- refineall
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - heavy.html (typical cycle)
    - solve.html (solve_script_mad)
    - solve.html (solve_script_mir)
- refineb
    - all_keywords.html (heavy_control)
- refined
    - all_keywords.html (atom_input)
    - all_keywords.html (heavy_control)
    - fdiff.html
    - heavy.html (mad phasing)
    - heavy.html (typical cycle)
- refinement
    - all_keywords.html
    - all_keywords.html (control)
    - all_keywords.html (atom_input)
    - all_keywords.html (heavy_control)
    - all_commands.html
- refinenone
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - heavy.html (mad phasing)
    - heavy.html (goto)
- refineocc
    - all_keywords.html (heavy_control)
- refineoccb
    - all_keywords.html (heavy_control)
- refinetemp
    - all_keywords.html (heavy_control)
- refinex
    - all_keywords.html (heavy_control)
- refinexyz
    - all_keywords.html (heavy_control)
- refinexyzb
    - heavy.html (goto)
- refiney
    - all_keywords.html (heavy_control)

- refinez
  - all_keywords.html (heavy_control)
- refscattfactors
  - all_keywords.html (scatter)
  - analyze_mad.html (keywords-analyze_mad)
  - auto_keywords.html
  - different.html
  - madmrg.html (keywords)
- reject
  - all_keywords.html (read_reject)
  - heavy.html (rejecting data)
  - localscale.html (keywords for localscale)
  - localscale.html (merge)
- rejected
  - hassp.html (more about hassp)
  - localscale.html (more on merge)
  - madbst.html (how madbst works)
- rejecting
  - all_keywords.html
  - all_keywords.html (read_reject)
  - heavy.html
  - heavy.html (rejecting data)
- rejects
  - localscale.html (merge)
  - localscale.html (more on merge)
- removal
  - all_keywords.html (veryquick)
- remove
  - all_keywords.html (hassp)
  - formats.html (ccp4 mtz)
- requested
  - new.html
- res_phase
  - all_keywords.html (crystal_info)
  - all_keywords.html (heavy_control)
  - auto_keywords.html
  - different.html
- rescales
  - FAQS.html (good)
  - scale_mad.html

- residual
  - all_keywords.html (heavy_control)
  - heavy.html (statistical output)
  - madbst.html (how madbst works)
- residuals
  - all_keywords.html (heavy_control)
  - heavy.html
  - heavy.html (statistical output)
  - heavy.html (correlated phasing)
  - heavy.html (typical cycle)
- residues
  - all_keywords.html (crystal_info)
  - analyze_mad.html
  - analyze_mad.html (keywords-analyze_mad)
  - auto_keywords.html
  - generate.html
- resolution
  - FAQS.html (good)
  - FAQS.html (input2)
  - FAQS.html (checksolve)
  - FAQS.html (bmin)
  - FAQS.html (cell_dim_error)
- resolution_steps
  - all_keywords.html (veryquick)
  - different.html
  - new.html
- resolutions
  - FAQS.html (compare_z)
  - hassp.html (more about hassp)
- resolve
  - FAQS.html (more)
  - FAQS.html (hand)
  - how_solve_works.html (refine)
  - intro.html (help)
  - new.html
- resolve_autobuild
  - new.html
- resolve_build
  - FAQS.html (hand)
- resources

    - FAQS.html (sbin_loader)
- restores
    - new.html
- resulted
    - new.html
- reverse
    - FAQS.html (hand)
- reversed
    - FAQS.html (hand)
- sad
    - FAQS.html (more)
    - FAQS.html (good)
    - FAQS.html (hand)
    - FAQS.html (bmin)
    - all_keywords.html (veryquick)
- save
    - all_keywords.html (control)
- save_files
    - all_keywords.html (control)
- says
    - FAQS.html (sbin_loader)
- sbin
    - FAQS.html (sbin_loader)
- scale_derivative
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - sample_scripts/ccp4_mir.html
- scale_mad
    - all_keywords.html (read_reject)
    - all_keywords.html (file_names)
    - all_commands.html
    - analyze_mad.html
    - auto_keywords.html
- scale_mir
    - all_keywords.html (read_reject)
    - all_commands.html
    - analyze_mir.html
    - generate.html
    - how_solve_works.html (refine)
- scale_native

    - all_keywords.html (read_reject)
    - all_commands.html
    - auto_keywords.html
    - generate.html
    - how_solve_works.html (refine)
- scaled
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - formats.html
    - formats.html (ccp4 mtz)
    - heavy.html (mad phasing)
- scalednativefile
    - scale_mir.html
    - scale_native.html
    - scale_native.html (script-scale_native)
- scalepack
    - FAQS.html (nomerge)
    - all_keywords.html (read_reject)
    - auto.html
    - auto_keywords.html
    - formats.html
- scaling
    - FAQS.html (input)
    - FAQS.html (nomerge)
    - all_keywords.html
    - all_keywords.html (read_reject)
    - all_keywords.html (atom_input)
- scatter
    - sample_scripts/script_list.html (sad phasing)
- score
    - all_keywords.html (veryquick)
    - auto_keywords.html
    - different.html
    - how_solve_works.html (speed)
    - output.html (solve_prt)
- scoring
    - all_keywords.html (solve_control)
    - all_keywords.html (veryquick)
    - auto_keywords.html

    - different.html
    - next.html (addsolve)
- scoring_table
    - all_keywords.html (solve_control)
    - auto_keywords.html
    - next.html (addsolve)
- scratch
    - all_keywords.html (control)
    - new.html
- script
    - FAQS.html (more)
    - FAQS.html (hand)
    - all_keywords.html (file_names)
    - all_keywords.html (solve_control)
    - all_keywords.html (veryquick)
- scriptfile
    - analyze_mad.html (keywords-analyze_mad)
    - analyze_mir.html (script-analyze_mir)
- sdiff
    - fdiff.html (keywords for fdiff)
- sdub
    - all_keywords.html (hassp)
- searches
    - all_keywords.html (crystal_info)
    - all_keywords.html (hassp)
    - auto_keywords.html
    - hassp.html
    - hassp.html (more about hassp)
- searching
    - hassp.html
    - hassp.html (more about hassp)
- searchregion
    - all_keywords.html (hassp)
    - hassp.html (analyzing a solution)
- section
    - binary.html (map)
    - intro.html
    - intro.html (data_files)
    - references.html
- seeds

    - all_keywords.html (solve_control)
    - auto_keywords.html
    - different.html
    - how_solve_works.html (speed)
    - intro.html (commands)
- sent
    - FAQS.html (more)
- separateano
    - misc.html (math)
- separates
    - scale_mad.html
- sept
    - FAQS.html (bmin)
- serious
    - new.html
- set_program_attributes
    - FAQS.html (sbin_loader)
- settings
    - FAQS.html (scatt_factors)
- severely
    - FAQS.html (more)
- sigbar
    - all_keywords.html (heavy_control)
- sigdelhg
    - formats.html (ccp4 mtz)
- sigderscale
    - all_keywords.html (atom_input)
- sigdph1
    - formats.html (ccp4 mtz)
    - sample_scripts/ccp4_mir.html
    - sample_scripts/ccp4_mad.html
- sigdph2
    - sample_scripts/ccp4_mir.html
    - sample_scripts/ccp4_mad.html
- sigdph3
    - sample_scripts/ccp4_mad.html
- sigfp
    - formats.html (ccp4 mtz)
    - output.html (solve_mtz)
    - sample_scripts/ccp4_mir.html

    - sample_scripts/ccp4_mad.html
- sigfph1
    - formats.html (ccp4 mtz)
    - sample_scripts/ccp4_mir.html
    - sample_scripts/ccp4_mad.html
- sigfph2
    - sample_scripts/ccp4_mir.html
    - sample_scripts/ccp4_mad.html
- sigfph3
    - sample_scripts/ccp4_mad.html
- sighg
    - formats.html (ccp4 mtz)
- sigi
    - all_keywords.html (read_reject)
    - formats.html (ccp4_unmerged)
- sigma
    - FAQS.html (good)
    - all_keywords.html (read_reject)
    - all_keywords.html (file_names)
    - all_keywords.html (column_numbers)
    - all_keywords.html (veryquick)
- sigma_i_ratio
    - all_keywords.html (read_reject)
- sigma_remove
    - all_keywords.html (hassp)
- sigma_scale
    - misc.html (math)
- sigmas
    - FAQS.html (good)
    - FAQS.html (all_data)
    - all_keywords.html (read_reject)
    - all_keywords.html (atom_input)
    - all_keywords.html (heavy_control)
- signal
    - FAQS.html (good)
    - FAQS.html (all_data)
    - all_keywords.html (crystal_info)
    - all_keywords.html (heavy_control)
    - auto_keywords.html
- signatscale

    - all_keywords.html (heavy_control)
- signif
    - hassp.html (more about hassp)
- significance
    - all_keywords.html (hassp)
    - hassp.html (more about hassp)
- signify
    - filemerge.html
- signs
    - FAQS.html (hand)
- sim
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - new.html
- sinc
    - all_keywords.html (scatter)
- siras
    - FAQS.html (fom)
    - all_keywords.html (file_names)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - binary.html (dorgbn files)
- sized
    - FAQS.html (bigger)
- skip
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - scale_mir.html
    - scale_native.html (script-scale_native)
- skipped
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - scale_mir.html
    - scale_native.html (script-scale_native)
- slash
    - intro.html (symmetry)
- slashes
    - intro.html (symmetry)
- sn_min
    - all_keywords.html (crystal_info)

    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - different.html
- sn_ratio_min
    - all_keywords.html (crystal_info)
    - all_keywords.html (heavy_control)
    - auto_keywords.html
    - different.html
- soln
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (correlation)
    - notes/solve_brief_examples.html (gmcsf)
- solve
    - FAQS.html
    - FAQS.html (more)
    - FAQS.html (good)
    - FAQS.html (input)
    - FAQS.html (input2)
- solve2
    - FAQS.html (new_access)
    - intro.html (license)
    - intro.html (solvedir)
- solve_combine
    - combine.html
- solve_fast_sad
    - all_keywords.html (veryquick)
- solve_giant
    - FAQS.html (bigger)
- solve_huge
    - FAQS.html (bigger)
- solve_inverse
    - FAQS.html (hand)
    - new.html
- solve_mad
    - all_keywords.html (file_names)
    - all_keywords.html (veryquick)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - next.html (addsolve)
- solve_mir

    - all_keywords.html (file_names)
    - all_keywords.html (veryquick)
    - analyze_mir.html
    - analyze_mir.html (script-analyze_mir)
    - next.html (addsolve)
- solvedatafile
    - all_keywords.html (file_names)
    - analyze_mad.html (keywords-analyze_mad)
    - binary.html (dorgbn files)
- solvefile
    - generate.html
- solvetmpdir
    - intro.html (started)
    - intro.html (solvedir)
    - new.html
    - table-of-contents.html
    - notes/solve_brief_examples.html (gene 5 protein)
- sometimes
    - different.html
- somewhat
    - sample_scripts/script_list.html
- sorry
    - FAQS.html (combine)
- source
    - FAQS.html (bigger)
    - auto.html
- specification
    - all_keywords.html (read_reject)
    - auto_keywords.html
- speeding
    - new.html
- ssft
    - all_keywords.html (hassp)
- ssin
    - all_keywords.html (hassp)
- staff
    - FAQS.html (scatt_factors)
    - auto.html
- stage
    - all_keywords.html (solve_control)

- statistics
  - FAQS.html (all_data)
  - all_keywords.html (heavy_control)
  - heavy.html (statistical output)
  - heavy.html (correlated phasing)
  - heavy.html (typical cycle)
- strongly
  - all_keywords.html (scatter)
- strp
  - all_keywords.html (hassp)
- substituting
  - new.html
- superposition
  - hassp.html
- superquick_build
  - new.html
- swap
  - all_keywords.html (read_reject)
  - auto_keywords.html
  - import_export.html (import)
  - import_export.html (scripts for import)
- swap_ano
  - FAQS.html (hand)
  - all_keywords.html (read_reject)
  - auto_keywords.html
- swapped
  - all_keywords.html (read_reject)
  - auto_keywords.html
  - new.html
- symfile
  - all_keywords.html (crystal_info)
  - all_keywords.html (file_names)
  - auto_keywords.html
  - generate.html
  - intro.html (scripts)
- symmetry
  - all_keywords.html (crystal_info)
  - all_keywords.html (file_names)
  - all_keywords.html (hassp)
  - all_commands.html

    - auto_keywords.html
- symop
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (beta-catenin)
    - notes/solve_brief_examples.html (gmcsf)
- terminated
    - new.html
- thorough
    - all_keywords.html (veryquick)
    - auto_keywords.html
    - different.html
- threshold
    - all_keywords.html (heavy_control)
- tmp
    - intro.html (solvedir)
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (beta-catenin)
    - notes/solve_brief_examples.html (gmcsf)
- toss
    - all_keywords.html (read_reject)
    - auto_keywords.html
    - localscale.html (keywords for localscale)
    - localscale.html (keywords for merge)
    - misc.html (getanom)
- tossbad
    - all_keywords.html (read_reject)
    - heavy.html (rejecting data)
    - localscale.html (keywords for localscale)
    - localscale.html (keywords for merge)
- tossed
    - madbst.html (output)
    - madbst.html (keywords)
    - misc.html (getanom)
- tossing
    - different.html
- transformation
    - FAQS.html (scatt_factors)
- translation
    - import_export.html (import)
- trialsite

    - all_keywords.html (hassp)
- trouble
    - FAQS.html (close sites)
- unclear
    - all_keywords.html (scatter)
- uncorrelated
    - heavy.html (correlated phasing)
- underscore
    - intro.html (symmetry)
- underscores
    - intro.html (symmetry)
- unformatted
    - binary.html (dorgbn files)
    - import_export.html (dorgbn files)
- unique
    - hassp.html (more about hassp)
    - hassp.html (analyzing a solution)
    - heavy.html (correlated phasing)
    - intro.html (commands)
- unit cell
    - FAQS.html (cell_dim_error)
    - auto.html
    - hassp.html (more about hassp)
    - hassp.html (analyzing a solution)
    - import_export.html (ffttoboss)
- units
    - FAQS.html (close sites)
    - all_keywords.html (veryquick)
    - auto_keywords.html
    - hassp.html (more about hassp)
    - heavy.html (correlated phasing)
- unknown
    - auto.html (run)
    - intro.html (solvedir)
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (beta-catenin)
    - notes/solve_brief_examples.html (gmcsf)
- unlimit
    - FAQS.html (sbin_loader)
- unmerged

    - all_keywords.html (read_reject)
    - auto_keywords.html
    - formats.html
    - formats.html (denzo/scalepack)
    - formats.html (free-format i)
- unscaled
    - how_solve_works.html
- uppercase
    - formats.html (ccp4 mtz)
- use_f_bins
    - all_keywords.html (heavy_control)
- use_input_phases
    - all_keywords.html (veryquick)
- uvw
    - all_keywords.html (hassp)
- uvw_remove
    - all_keywords.html (hassp)
- var
    - intro.html (solvedir)
    - notes/solve_brief_examples.html (gene 5 protein)
    - notes/solve_brief_examples.html (beta-catenin)
    - notes/solve_brief_examples.html (gmcsf)
- vectorsum
    - misc.html (math)
- verbose
    - all_keywords.html (control)
    - auto_keywords.html
- versa
    - different.html
- version
    - FAQS.html (good)
    - FAQS.html (bigger)
    - FAQS.html (new_access)
    - FAQS.html (bmin)
    - FAQS.html (exec_error)
- versions
    - FAQS.html (new_access)
    - FAQS.html (bmin)
    - copyright.html
    - new.html

    - scoring.html
- veryquick
    - all_keywords.html (veryquick)
    - how_solve_works.html (speed)
- viewing
    - output.html (solve_ccp4)
- volume
    - FAQS.html (scatt_2)
- wavelength
    - FAQS.html (more)
    - FAQS.html (input)
    - FAQS.html (scatt_factors)
    - all_keywords.html (read_reject)
    - all_keywords.html (file_names)
- wavelength?
    - formats.html
- wavelengths
    - FAQS.html (more)
    - analyze_mad.html
    - analyze_mad.html (keywords-analyze_mad)
    - generate.html
    - heavy.html (mad phasing)
- wclu
    - all_keywords.html (scatter)
- weighing
    - all_keywords.html (heavy_control)
    - auto_keywords.html
- weight
    - fdiff.html
    - fdiff.html (script for fdiff)
    - fdiff.html (keywords for fdiff)
    - madmrg.html (notes)
    - misc.html (weights)
- weighted
    - binary.html (dorgbn files)
    - heavy.html (statistical output)
    - madbst.html (how madbst works)
    - madbst.html (output)
    - madmrg.html (notes)
- weighting

    - all_keywords.html (veryquick)
    - all_commands.html
    - auto_keywords.html
    - fdiff.html
    - heavy.html (refinement)
- weights
    - all_commands.html
    - fdiff.html
    - misc.html
    - misc.html (weights)
    - misc.html (script for weights)
- whenever
    - new.html
- why
    - FAQS.html (bmin)
    - FAQS.html (fom)
    - FAQS.html (nomerge)
    - FAQS.html (exec_error)
    - FAQS.html (cell_dim_error)
- wilson
    - all_keywords.html (read_reject)
    - localscale.html
    - localscale.html (keywords for localscale)
    - madbst.html (how madbst works)
    - misc.html (math)
- won
    - FAQS.html (combine)
    - FAQS.html (close sites)
    - auto.html
    - sample_scripts/script_list.html
- xplor
    - all_keywords.html (file_names)
    - import_export.html
    - import_export.html (dorgbn-to-xplor)
    - misc.html (script for weights)
- yield
    - FAQS.html (good)
- ~mok
    - next.html

electronic reprint

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Maximum-likelihood density modification using pattern recognition of structural motifs

Thomas C. Terwilliger

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Maximum-likelihood density modification using pattern recognition of structural motifs

Thomas C. Terwilliger

Bioscience Division, Mail Stop M888, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence e-mail: terwilliger@lanl.gov

Received 17 April 2001
Accepted 17 August 2001

The likelihood-based approach to density modification [Terwilliger (2000), *Acta Cryst.* D**56**, 965–972] is extended to include the recognition of patterns of electron density. Once a region of electron density in a map is recognized as corresponding to a known structural element, the likelihood of the map is reformulated to include a term that reflects how closely the map agrees with the expected density for that structural element. This likelihood is combined with other aspects of the likelihood of the map, including the presence of a flat solvent region and the electron-density distribution in the protein region. This likelihood-based pattern-recognition approach was tested using the recognition of helical segments in a largely helical protein. The pattern-recognition method yields a substantial phase improvement over both conventional and likelihood-based solvent-flattening and histogram-matching methods. The method can potentially be used to recognize any common structural motif and incorporate prior knowledge about that motif into density modification.

1. Density modification by likelihood optimization

Although very powerful experimental methods exist for determining crystallographic phases in macromolecular crystallography, it is frequently necessary to improve or extend these phases before an atomic model of the macromolecule can be built. A variety of tools for density modification have been developed for this purpose, including solvent flattening, non-crystallographic symmetry averaging, histogram matching, phase extension, molecular replacement, entropy maximization and iterative model building (Abrahams & Leslie, 1996; Abrahams, 1997; Béran & Szöke, 1995; Bricogne, 1984, 1988; Cowtan & Main, 1993, 1996; Giacovazzo & Siliqi, 1997; Goldstein & Zhang, 1998; Gu *et al.*, 1997; Lunin, 1993; Perrakis *et al.*, 1997; Podjarny *et al.*, 1987; Prince *et al.*, 1988; Refaat *et al.*, 1996; Roberts & Brünger, 1995; Rossmann & Arnold, 1993; Shneerson *et al.*, 2001; Szöke, 1993; Szöke *et al.*, 1997; Terwilliger, 2000; Vellieux *et al.*, 1995; Wilson & Agard, 1993; Xiang *et al.*, 1993; Zhang & Main, 1990; Zhang, 1993; Zhang *et al.*, 1997). The basis of density modification is that there are many possible sets of phases that are reasonably consistent with the experimental data and the most likely of these sets of phases are those that lead to electron-density maps that are most consistent with expectations for a macromolecule. The most common way to carry out density modification has been to calculate an electron-density map, modify it to meet expectations, calculate modified phases and combine the modified phases with experimental phases to yield new estimates of the crystallographic phases. This method has the disadvantages that optimal weighting of

© 2001 International Union of Crystallography
Printed in Denmark – all rights reserved

electronic reprint

modified and experimental phases is difficult and that it is not clear when to stop iterating. The difficulty in weighting in particular is well known and a number of approaches have been designed to circumvent it, including the use of maximum-entropy methods and the use of weighting optimized using cross-validation (Xiang *et al.*, 1993; Roberts & Brünger, 1995; Cowtan & Main, 1996) and 'solvent flipping' (Abrahams & Leslie, 1996).

We have recently developed a method for carrying out density modification that consists of directly maximizing the likelihood of the structure factors, including both experimental information and the characteristics of the electron density resulting from the structure factors (Terwilliger, 1999, 2000). The general idea is very simple. We express the total likelihood of a set of structure factors $\{F_{\mathbf{h}}\}$ in terms of three quantities: (i) any prior knowledge we have from other sources about these structure factors, (ii) the likelihood that we would have measured the observed set of structure factors $\{F_{\mathbf{h}}^{\rm OBS}\}$ if this set of structure factors were correct and (iii) the likelihood that the map resulting from this set of structure factors $\{F_{\mathbf{h}}\}$ is consistent with our prior knowledge about this and other macromolecular structures. This can be written as

$$\mathrm{LL}(\{F_{\mathbf{h}}\}) = \mathrm{LL}^{o}(\{F_{\mathbf{h}}\}) + \mathrm{LL}^{\rm OBS}(\{F_{\mathbf{h}}\}) + \mathrm{LL}^{\rm MAP}(\{F_{\mathbf{h}}\}), \quad (1)$$

where $\mathrm{LL}(\{F_{\mathbf{h}}\})$ is the log-likelihood of a possible set of crystallographic structure factors $F_{\mathbf{h}}$, $\mathrm{LL}^{o}(\{F_{\mathbf{h}}\})$ is the log-likelihood of these structure factors based on any information that is known in advance, such as the distribution of intensities of structure factors (Wilson, 1949), $\mathrm{LL}^{\rm OBS}(\{F_{\mathbf{h}}\})$ is the log-likelihood of these phases given the experimental data alone and $\mathrm{LL}^{\rm MAP}(\{F_{\mathbf{h}}\})$ is the log-likelihood of the electron-density map resulting from these phases. In this formulation, density modification consists of maximizing the total likelihood given by (1). To maximize this likelihood, it is necessary both to define a map-likelihood function and to have a practical way of finding structure factors that maximize it.

We recently developed a formulation of the map-likelihood function that often allows a straightforward and rapid optimization of the total likelihood in (1). The log-likelihood for the electron-density map $\mathrm{LL}^{\rm MAP}(\{F_{\mathbf{h}}\})$ is written as the integral over the map of a local log-likelihood of electron density, $\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]$,

$$\mathrm{LL}^{\rm MAP}(\{F_{\mathbf{h}}\}) \simeq \frac{N_{\rm REF}}{V} \int_V \mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\, \mathrm{d}^3\mathbf{x}. \quad (2)$$

This formulation neglects contributions to the log-likelihood of the map that involve more than one point at a time, but is nevertheless very useful in describing the overall likelihood of the map (Terwilliger, 1999, 2000).

As long as the first and second derivatives of the local log-likelihood of electron density with respect to electron density can be calculated, a steepest-ascent method can be used to optimize the total likelihood in (1) (Terwilliger, 1999, 2000). In this broad class of situations, an FFT-based method can be used to approximate derivatives of the total map log-likelihood function with respect to each structure factor (Terwilliger, 1999, 2000). These derivatives can then in turn be used in a Taylor's series expansion to approximate the total map log-likelihood function as a function of each structure factor. This makes it practical to optimize the total likelihood in (1) because the other terms (*a priori* knowledge of phases, and experimental phase information) are also normally expressed separately for each structure factor. In each cycle of optimization, a new probability distribution for each structure factor (or phase) is obtained by calculating the relative likelihood of each possible value of that structure factor using (1) with this approximation for the map log-likelihood function.

The local map log-likelihood function in (2) is a critical element in our maximum-likelihood density-modification approach. This likelihoood function could include any type of expectations about the electron-density value at a particular point in the map. In particular, we have shown that expectations about electron-density values at points both in the solvent region and in the protein region of a protein crystal can be included in maximum-likelihood density modification and that this approach can be very powerful for improving crystallographic phases (Terwilliger, 1999, 2000). We show here that the same approach can be used to incorporate detailed information about patterns of electron density in a map such as those corresponding to secondary-structural elements in a protein structure.

2. Local log-likelihood function for a map

The local map log-likelihood function is essentially a statement of the plausibility of each possible value of electron density at a point in the electron-density map. It is important to recognize that for the present purpose this probability of electron density is in the context of all the errors in the map caused by uncertainty in structure factors (Terwilliger, 2000). This distinction is necessary because in any one cycle of our approach each phase is optimized independently of all others. Consequently, as one phase (or structure factor) is being optimized it is in the context of the errors remaining in all other phases. This means that even in an idealized case in which the value of the true electron density was known exactly at a particular point in the map, the correct value of a particular phase would not ordinarily lead to exactly this value of electron density. Instead, the probability distribution of plausible electron densities at this point would have a finite breadth corresponding to the overall error in the map.

Following this approach, the probability distribution $p(\rho)$ for electron density at the point $\mathbf{x}$ in a map with substantial phase errors can be written as

$$p(\rho) = \int_{\rho_T} p(\rho_T) \exp\left[-\frac{(\rho - \beta\rho_T)^2}{2\sigma^2_{\rm MAP}}\right] \mathrm{d}\rho_T, \quad (3)$$

where $p(\rho_T)$ is the probability distribution for electron density in a model (perfect) case, $\sigma^2_{\rm MAP}$ is the variance in the map and β is a scale factor (Terwilliger, 2000).

As it is generally not known for certain whether a particular point $\mathbf{x}$ is in the protein or solvent region, it is useful to write the local map-likelihood function as the sum of conditional

electronic reprint

probabilities dependent on which environment the point is located in,

$$\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})] = \ln\{p[\rho(\mathbf{x})|\mathrm{PROT}]p_{\mathrm{PROT}}(\mathbf{x}) + p[\rho(\mathbf{x})|\mathrm{SOLV}]p_{\mathrm{SOLV}}(\mathbf{x})\}, \quad (4)$$

where $p_{\mathrm{PROT}}(\mathbf{x})$ is the probability that $\mathbf{x}$ is in the protein region, $p[\rho(\mathbf{x})|\mathrm{PROT}]$ is the conditional probability for $\rho(\mathbf{x})$ given that $\mathbf{x}$ is in the protein region and $p_{\mathrm{SOLV}}(\mathbf{x})$ and $p[\rho(\mathbf{x})|\mathrm{SOLV}]$ are the corresponding quantities for the solvent region. The probability that $\mathbf{x}$ is in the protein or solvent regions can estimated by a modification of the methods of Wang (1985) and Leslie (1987) as described earlier (Terwilliger, 1999).

3. Incorporating information obtained from image reconstruction

The local log-likelihood function for the map in (4) is based simply on probability distributions for the protein and solvent regions of the map. The same approach can be applied to information on the likely values of electron density at a particular point derived from any other source. In particular, suppose that it were known that there is the probability p_H that there is a helix in a particular orientation located at a particular place in the unit cell. Then our prior knowledge about the electron-density distribution in a helix could be used in just the same way as our knowledge about the electron density in the solvent region of the unit cell. At each point within and in the immediate vicinity of this helix, a probability distribution for plausible values of electron density could be constructed using model values of electron density for a helix along with (3). These probability distributions could then be used in a local log-likelihood function that is an extension of (4):

$$\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})] = \ln\{p[\rho(\mathbf{x})|\mathrm{PROT}]p_{\mathrm{PROT}}(\mathbf{x}) + p[\rho(\mathbf{x})|\mathrm{SOLV}]p_{\mathrm{SOLV}}(\mathbf{x}) + p[\rho(\mathbf{x})|H]p_H(\mathbf{x})\}, \quad (5)$$

where $p_H(\mathbf{x})$ refers to the probability that there is a helix at a known location, with a known orientation, somewhere near the point $\mathbf{x}$; $p[\rho(\mathbf{x})|H]$ is the probability distribution for electron density at this point given that this helix actually is present. As there is nothing special about helices (other than their relative regularity), (5) could equally well be used to include any other type of structural motif or indeed any other pattern of electron density that can be recognized. The significance of (5) is that it provides a way to incorporate pattern recognition (the probability that there is a helix with this orientation at this point) into density modification. If the pattern to be detected involves a large part of the map, then it might be identifiable even when errors in the map are very large. Then if the pattern is well defined the last term in (5) could potentially contribute very substantially to the local log-likelihood function and therefore to density modification. This approach can be thought of as a likelihood-based extension of the iterative skeletonization procedure for phase improvement (Baker *et al.*, 1993; Wilson & Agard, 1993) and of the iterative model-building procedures incorporated into *ARP* and *wARP* (Perrakis *et al.*, 1999).

The formulation in (5) essentially segments the map into points within protein, within solvent and within another pattern (helix) of electron density. Strictly speaking, these categories are clearly not mutually exclusive, as a point can be both within protein and within a helix. Furthermore, a particular point could be within more than one helix pattern (as the template used to identify a helix might be shorter than the actual helix and several overlapping patterns of helix might be recognized). It is convenient, however, to use the most informative piece of information when there is either type of overlap. If a point is both within the protein region and within a helix, for example, the fact that it is within a helix is far more informative because it defines the electron density very precisely, while the fact that the point is within the protein only gives a very broad range for possible values of electron density. In practice, if more than one pattern has information about the electron density at a particular point, then the pattern that has the highest probability is used. Then the probabilities that the point is in protein or solvent are modified from our earlier expressions (Terwilliger, 1999, 2000) by normalizing their total to simply be whatever the probability is that the point is not in this pattern.

4. Image reconstruction by template matching

Template matching has been used as an aid to map interpretation for some time in X-ray crystallography (Kleywegt & Jones, 1997, 1998; Cowtan, 1998). Many structural elements in proteins are quite uniform and can sometimes be recognized in even a noisy electron-density map. In the context of image reconstruction, once an element such as a helical region is recognized, the electron density in the neighborhood of the main-chain atoms can often be estimated more accurately from the model of a helix than from the map itself.

To make optimal use of (5), a method is needed for estimating the probability that a particular pattern of electron density (*e.g.* one corresponding to a helix) is located at each possible position and with each possible orientation in the unit cell. To make this practical, it is convenient to separate it into three steps. First, a template is constructed that is an average of the patterns of electron density found in many instances where it occurs. Next, locations and orientations of a template (such as the electron density for a helix) that match the electron density in the map to some degree are identified. Then the probabilities of these possibilities are estimated.

4.1. Construction of a template for a helix

Although helices are relatively regular secondary structures, there is some variation from one to another in the precise locations of atoms and in their thermal factors. Even more importantly, the side chains in one helix may be completely different to those in another. Consequently, construction of a template that has average features is useful for the purpose of pattern matching. Additionally, it is helpful

electronic reprint

to have a point-by-point estimate of the standard deviation of this density that can be used to identify regions within the template that have more or less variation. We used a simple method to generate a template and standard deviation of the template for helices. Residues 133–138 of myoglobin (PDB entry 1a6m) were chosen as a model helical segment. Then 326 segments of six amino acids from the largely helical protein phycoerythrin (Chang *et al.*, 1996; PDB code1lia) for which the N, C, C^{α} and O atoms could be superimposed on the corresponding atoms in the myoglobin helix with an r.m.s. deviation of 0.5 Å or less were used to generate an average template for α-helices.

The template was constructed by superimposing each six-amino-acid helical segment of phycoerythrin on the myoglobin helix and calculating an electron-density map at a resolution of 3 Å based on all atoms of the phycoerythrin structure that fell inside a 20 Å cube with the helix at the center. The resulting electron density within 2.5 Å of an atom in the myoglobin helix was averaged to yield our helical template. The average density in the template region was adjusted to a value of zero and all points outside the template region were set to values of zero. At the same time, the standard deviation of electron density at the same set of points was determined.

Fig. 1 shows the resulting helical template. The positions of C^{β} atoms are visible, but all further side-chain atoms are sufficiently different at different positions that no density is visible.

4.2. Matching a helix template to an electron-density map

We used an FFT-based convolution method to identify rotations and translations of our helix template that match the electron density in a map to some degree (Kleywegt & Jones, 1997; Cowtan, 1998). The helix template was rotated in real space and placed at the origin of a unit cell with dimensions identical to the map to be searched. Structure factors for the rotated template were calculated in space group $P1$ and the convolution of the template and the electron-density map was calculated using an FFT. Each point in this convolution corresponds to a translation of the rotated template. The value of the convolution at each point is essentially the integral over the template region of the density in the rotated translated template, multiplied by the density in the map. This product is expected to be high if the rotated translated template has a high correspondence to the map and low otherwise.

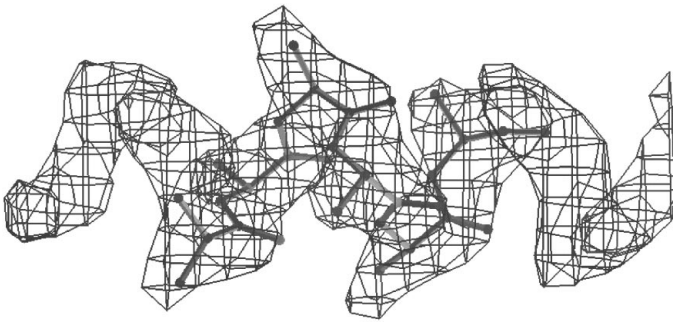


Figure 1
Averaged helical template. The template was calculated at a resolution of 3 Å as described in the text.

In our implementation of a helix search, the template is rotated in increments of 10° over three rotation axes. As our α-helix template is essentially symmetric when rotated 100° about its axis, the search only included 100° of rotation about the helix axis.

To identify peaks in this search that are reasonably likely to correspond to actual helical segments in the electron-density map, a height cutoff was calculated such that in a random map only about one peak would be chosen every other rotation. The cutoff was estimated from the number of reflections (an estimate of the number of degrees of freedom in the map), the mean and standard deviation of the convolution function. Typically, the cutoff was in the range of 3σ to 4σ and typically about 200–2000 peaks were saved. In cases where there are templates with center-to-center distances of less than 2 Å, the one with the higher peak height was chosen.

Once matches of template to map are identified in this fashion, the rotation and translation parameters are refined by minimizing the residual error in the fit between the map and the template. This residual error $\sigma_{\rm RESID}$ is estimated from the r.m.s. difference $\sigma_{\rm FIT}$ between the map and the template (after multiplying the template by a scale factor α and adding an adjustable offset) and the uncertainty in the template itself σ_H (based on the variability in electron densities for model helices),

$$\sigma^2_{\rm RESID} = \sigma^2_{\rm FIT} - (\alpha\sigma_H)^2. \qquad (6)$$

4.3. Estimating probabilities of matches of a template to a map

In the scheme described above (5) for incorporating information about patterns of electron density in a map, it is essential to have an estimate of the probability p_H that a template is actually located at a particular position and with a particular orientation. The convolution-based search we use to identify plausible matches is not entirely suitable for this purpose because the peak heights are just a measure of how good the match is, not how likely it is that this pattern really is located there. To see the difference, consider a case where it is known somehow that there are no helices of six amino acids in length in a particular protein, but where there is a stretch of three amino acids in an α-helical conformation. A convolution-based search might show a large peak corresponding to overlap of the template and these three amino acids, yet only part of the template pattern is really present. In this example, it might be reasonable to say that there is a 50% chance that any given point in the template is a good description of the true electron density in the map, but not to say that this chance is 100%.

We use a combination of prior knowledge of the helical content of the protein in the crystal and the correlation coefficient of each match of template to map to estimate the

electronic reprint

probability that each match correctly identifies a region of the map with this pattern of electron density. First, the mean $\overline{\mathrm{CC}}$ and standard deviation σ_{CC} of correlation coefficients were determined for randomly chosen template orientations and translations. This allows an estimate for each match of template to map of the probability $p(\mathrm{CC}_{\mathrm{OBS}}|\mathrm{not}\ H)$ that this match with a correlation coefficient of $\mathrm{CC}_{\mathrm{OBS}}$ would have occurred entirely by chance (that is, if there were no helical pattern at this location),

$$p(CC_{\mathrm{OBS}}|\mathrm{not}\ H) \propto \exp\left[-\frac{(\mathrm{CC}_{\mathrm{OBS}}-\overline{\mathrm{CC}})^2}{2\sigma_{\mathrm{CC}}^2}\right]. \qquad (7)$$

Next, we estimate the number of templates that are likely to be needed to describe all the helical regions in the unit cell. This is necessarily rather approximate both because the number of residues in helical conformation is not ordinarily known very accurately and because in our method the templates describing a helix can overlap. Using the prior knowledge of the fraction f_H of the macromolecule that is in a helical conformation and of the fraction f_{PROT} of the unit cell that is occupied by macromolecule, the cell volume V and the template volume V_{template}, and using the empirical observations that about 70% (f_{template}) of the volume in a model helical protein is within a corresponding helical template and that only about 35% (f_{unique}) of each template does not overlap with another template, we can write that

$$N_{\mathrm{template}} \simeq \frac{f_H f_{\mathrm{PROT}} f_{\mathrm{template}} V}{f_{\mathrm{unique}} V_{\mathrm{template}}}. \qquad (8)$$

Now we can estimate the relative probability $p(H|\mathrm{CC}_{\mathrm{OBS}})$ that each template match, with correlation coefficient $\mathrm{CC}_{\mathrm{OBS}}$, is at least partially correct (that is, it does not arise by chance),

$$p(H|\mathrm{CC}_{\mathrm{OBS}}) = \frac{p_o(H)\ p(\mathrm{CC}_{\mathrm{OBS}}|H)}{p_o(H)\ p(\mathrm{CC}_{\mathrm{OBS}}|H) + p_o(\mathrm{not}\ H)\ p(\mathrm{CC}_{\mathrm{OBS}}|\mathrm{not}\ H)}, \qquad (9)$$

where $p_o(H)$ and $p_o(\mathrm{not}\ H)$ are the *a priori* probabilities that there is or is not a helix located at this position and orientation and $p(\mathrm{CC}_{\mathrm{OBS}}|H)$ and $p(\mathrm{CC}_{\mathrm{OBS}}|\mathrm{not}\ H)$ are the probabilities that this correlation coefficient would be found for correct and incorrect matches, respectively. As the vast majority of locations and orientations do not correspond to a correct match, we can reasonably assume that $p_o(\mathrm{not}\ H) \simeq 1$. Additionally, as we are only considering the highest peaks in the convolution, it is reasonable to assume that correct matches could have led to any of the peak heights observed, so that $p(\mathrm{CC}_{\mathrm{OBS}}|H) \simeq 1$. As we have an expression for $p(\mathrm{CC}_{\mathrm{OBS}}|\mathrm{not}\ H)$ (7), the only unknown term in (9) is $p_o(H)$, the *a priori* probability that there is a helix in this position and orientation. We estimate $p_o(H)$ by adjusting it so that the total number of templates is equal to N_{template} (7–9):

$$N_{\mathrm{template}} = \sum_{\mathrm{templates}} p(H|\mathrm{CC}_{\mathrm{OBS}}), \qquad (10)$$

where the probability that each template match is at least partially correct is

$$p(H|\mathrm{CC}_{\mathrm{OBS}}) = \frac{p_o(H)}{p_o(H) + \exp-[(\mathrm{CC}_{\mathrm{OBS}}-\overline{\mathrm{CC}})^2/(2\sigma_{\mathrm{CC}}^2)]}. \qquad (11)$$

Although all possible matches with all levels of probability might ideally be included in the image-reconstruction process, we find that in practice only the most probable ones contribute in a useful way. Consequently, only template matches with a value of $p(H|\mathrm{CC}_{\mathrm{OBS}}) > 0.8$ are included.

Finally, as discussed above there may be many cases where part of the template matches a pattern in the map but another part does not. We estimate this fraction that matches the pattern (f_{match}) based on the ratio of the correlation coefficient for each match ($\mathrm{CC}_{\mathrm{OBS}}$) to the highest correlation coefficient for any match in the map ($\mathrm{CC}_{\mathrm{MAX}}$),

$$f_{\mathrm{match}} \simeq \frac{\mathrm{CC}_{\mathrm{OBS}}}{\mathrm{CC}_{\mathrm{MAX}}}. \qquad (12)$$

Using (11) along with the average helix template and its standard deviation, we are now in a position to evaluate the new terms in (5). The probability $p_H(\mathbf{x})$ that there is a helix at a particular location and orientation that contributes some information about the electron density at point $\mathbf{x}$ is given by

$$p_H(\mathbf{x}) \simeq f_{\mathrm{match}} p(H|\mathrm{CC}_{\mathrm{OBS}}), \qquad (13)$$

where the probability that this template match is at least partially correct is $p(H|\mathrm{CC}_{\mathrm{OBS}})$ (11), where the estimated fraction of the template that is involved in the match is f_{match} and where H refers to a template match that overlaps the point $\mathbf{x}$. The probability distribution for electron density at $\mathbf{x}$ is given by (3), where the ideal electron-density distribution $p(\rho_T)$ is based on the mean ρ_{template} and standard deviation $\sigma_{\mathrm{template}}$ of the rotated translated template at the point $\mathbf{x}$,

$$p(\rho_T) \simeq \exp-\left[\frac{(\rho_T-\beta\rho_{\mathrm{template}})^2}{2\sigma_{\mathrm{template}}^2}\right]. \qquad (14)$$

5. Application to density modification of a map of an α-helical protein

We tested our pattern-matching approach to density modification using the armadillo repeat region of β-catenin, which is largely α-helical (Huber *et al.*, 1997). This structure was solved using MAD phasing on 15 Se atoms incorporated into methionine residues in the protein. To make the test suitably difficult, we used only three of the 15 Se atoms in calculating initial phases. As expected, this led to a very noisy map; the correlation coefficient of this map with a map calculated using phases from the refined model was only 0.29 (Fig. 2*a*). We carried out real-space density modification using *DM* (Cowtan & Main, 1996), resulting in some improvement of the map and a correlation coefficient of only 0.42 (not shown). The maximum-likelihood density-modification approach we described earlier (without any pattern recognition) resulted in a substantial improvement in the map, with a correlation coefficient of 0.62 (Fig. 2*b*). The pattern recognition of helices is illustrated in Fig. 2(*c*). In order to visualize the templates,

electronic reprint

the map shows the electron density in the rotated translated templates (from Fig. 1), multiplied by the probability that the template is a correct match (13). The density in the templates is a fairly good but not perfect match to the refined atomic model. The maximum-likelihood density modification with pattern recognition of helices improved the map even more substantially, with an overall correlation coefficient of 0.67 (Fig. 2*d*).

An even more difficult map to interpret is illustrated in Fig. 3. This map was created in the same way as the one in Fig. 2, except that only one selenium was used in phasing the 700 amino-acid residue protein. The starting correlation coefficient of the map with the model map was just 0.24; maximum-likelihood density modification increased this to 0.32 and density modification with pattern recognition to 0.51.

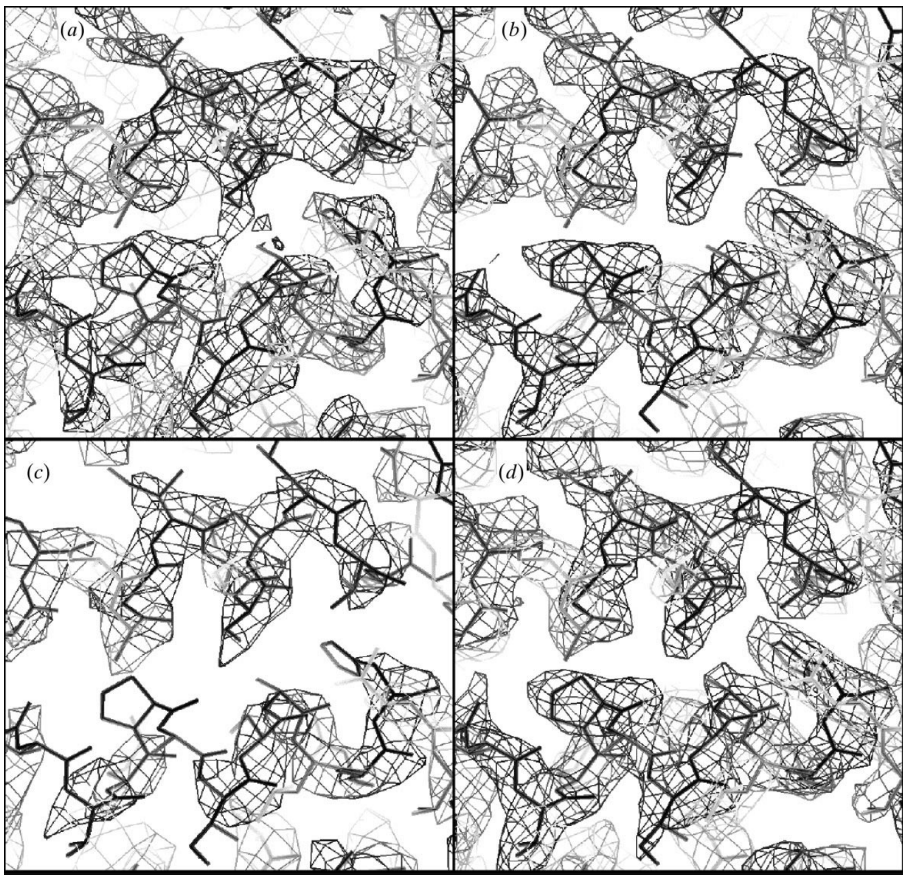


Figure 2
Experimental, real-space density-modified, maximum-likelihood density-modified and maximum-likelihood with pattern-recognition modified maps of an α-helical protein. The armadillo repeat region of β-catenin crystallizes in space group $C222_1$, with unit-cell parameters $a = 64$, $b = 102$, $c = 187$ Å and a solvent content of about 50% (Huber *et al.*, 1997). Phases were calculated with *SOLVE* (Terwilliger & Berendzen, 1999) using three selenium sites at a resolution of 3 Å. A section of this map is shown in (*a*). Real-space density modification was carried out with *DM* (Cowtan & Main, 1996) using solvent flattening with a solvent content of 50% and histogram matching (not shown). Maximum-likelihood density modification without image reconstruction was carried out as described earlier (Terwilliger, 2000) using a solvent content of 50% (*b*). Templates found in the experimental electron-density map are illustrated in (*c*). Maximum-likelihood density modification with pattern recognition was carried out as described in the text, using a solvent content of 50% and a fraction helical secondary structure of 80% (*d*). Template matches with a probability less than 0.8 were not included.

6. Discussion

The density-modification procedures developed here and in our recent work (Terwilliger, 1999, 2000) contain two substantial changes from existing methods. One is the use of optimization of a likelihood function rather than phase recombination between experimental and modified maps. The second is the use of a log-likelihood function for a map.

The optimization of a likelihood function (more precisely a posterior probability function in this case, *e.g.* equation 1) is important, as discussed in depth by others (Bricogne, 1984, 1988; Lunin, 1993), because it places density modification on a sound statistical foundation. In the present case, it also eliminates difficulties in weighting of experimental and modified phases. This optimization is made practical by the approaches we have developed involving reciprocal-space calculations of derivatives of the likelihood function with respect to structure factors.

A more far-reaching change from existing methods is in the development of a likelihood function for a map. This likelihood function is a statement of the plausibility of an electron-density map calculated from some set of structure factors. The plausibility can include any information about patterns of electron density that are expected and not expected. Our implementation of the likelihood function for a map (2) is a simplified version in which each point in the map is treated independently. The overall log-likelihood of the map is the integral over the unit cell of the local map log-likelihood function.

The use of a map-likelihood function is related to the methods of Szöke (1993; Szöke *et al.*, 1997) and Béran & Szöke (1995) in which crystallographic phases are obtained by matching the electron density in a part of the unit cell to a target value. The maximum-likelihood approach described here differs from these methods in that probabilistic descriptions of the expected electron density are used, allowing a calculation of phase probability distributions, rather than searching for a set of phases that is consistent with constraints.

The local log-likelihood function for a map can readily incorporate information about solvent and protein regions in the map if they are identified by some means (Terwilliger, 2000). After taking into consideration the noise in the map (3), the electron density at a point known to be in the solvent region is plausible only

electronic reprint

if it has values within a narrow range expected in the solvent. Similary, the density at a point in the protein region is plausible only if it has a value in the somewhat greater range expected in the protein region.

The patterns of electron density that are included in the local log-likelihood function need not be as simple as the probability distribution for electron density in solvent or protein regions. They can also include detailed information about the electron density in a region. (5) shows how to incorporate information on a pattern of density corresponding to a structural motif such as a fragment of α-helix. Any other fragment density information could be incorporated in a similar fashion.

It is important to recognize that the use of partial structure information in a likelihood function for a map is fundamentally different than using what may appear to be the same partial structure information in a σ_A or related model phase calculation (Read, 1986). The difference is that in the σ_A model phase calculation, the errors in the partial structure information are assumed to be the same everywhere in the unit cell, while in the map-likelihood approach, the errors can be explicitly specified for each point in the map. The method of Szöke (1993) also has this property.

The difference can be best appreciated in an idealized case where a only small fragment of structure is missing from an otherwise perfect model and a difference Fourier or similar calculation is carried out to identify the missing fragment. In the σ_A-weighted map, the difference density can be located anywhere in the map (though much will be in the correct region). In the map-likelihood approach, the fact that the density is known exactly everywhere except in the region of the missing fragment is explicitly taken into account. Consequently, in this approach all the difference density would be located in the region where the missing fragment is located.

In a more accessible case the same principle applies as well. In the examples described in this work, α-helices are identified in a map and used to improve phases. In the model phase calculation approach, the rotated translated templates (or coordinates of atoms in a model helix) would be used to calculate model phases and a σ_A-weighted combined phase map would be calculated. As in the more extreme example above, the uncertainties in electron density based on the model alone would be assumed to be distributed over the entire unit cell. In the map-likelihood method, uncertainties in electron density are relatively low in the entire region of each helical template (where the model electron density is relatively well known) and higher elsewhere in the protein region (where it is poorly known) and once again lower in the solvent region (where it is very precisely known). This point-by-point specification of uncertainty in the map allows a much more complete use of the available information about the partial model than the model phase method.

The key to the use of the local log-likelihood function for a map is the specification of a probability distribution for the electron density for some subset of points in the map. It does not matter if this specification says that all the points in a region have the same electron density or whether the points in this region have a particular pattern of electron density such as a part of a helix. Much the same amount of information is conveyed in either case and essentially the same amount of improvement in phases or structure factors can potentially be obtained in either case.

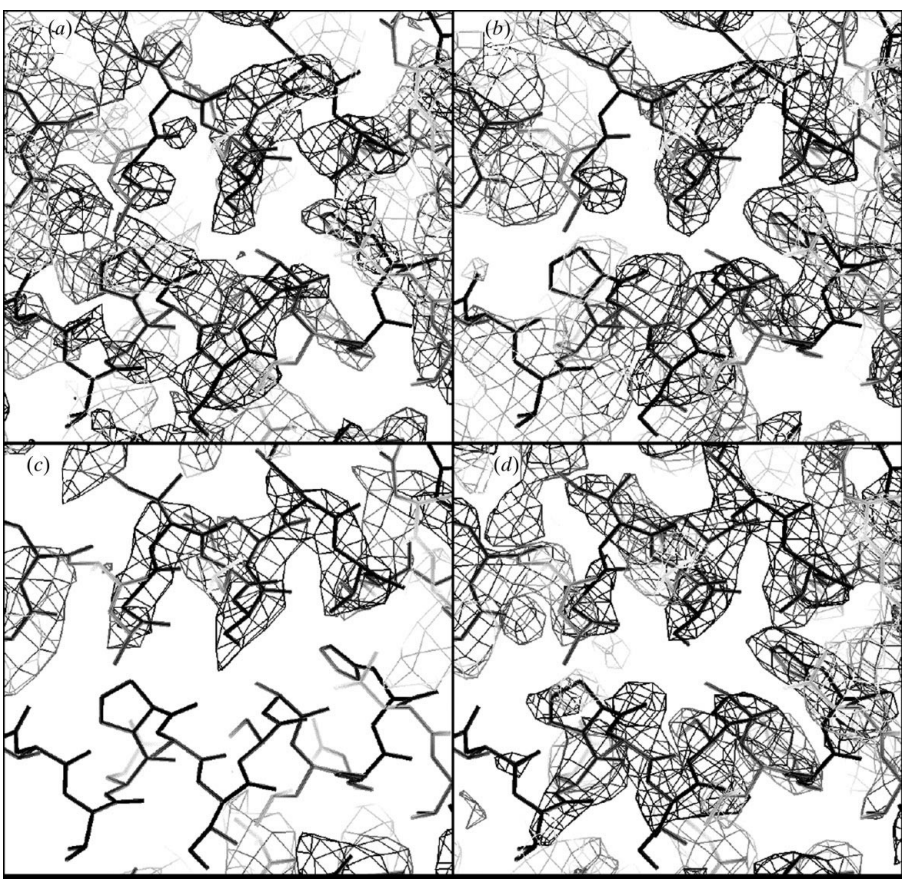


Figure 3
Template matching with a very noisy map. Analyses were carried out as in Fig. 2, starting with a map calculated using one selenium for phasing.

7. Conclusions

The methods we have developed here and in recent work (Terwilliger, 1999, 2000) provide a simple and practical way to incorporate prior knowledge of the electron density in a crystal structure into probability distributions for structure factors. The prior knowledge can range from the locations of solvent and protein regions to detailed infor-

electronic reprint

mation on a local pattern of electron density corresponding to a fragment of structure.

There are a number of important extensions of our approaches that can be readily envisioned. One is the incorporation of non-crystallographic symmetry information. Electron-density information from one copy of a macromolecule in the asymmetric unit can be used in our approach in the same way as other partial structure information. The ability to specify separate probability distributions for electron density at each point in the map will make it possible to take into account the different amounts of error in different parts of the partial model. In that way, the parts that are most similar can effectively be weighted more strongly and the parts that are more different be weighted less strongly, a property that is more difficult to achieve with current methods.

A second is in the area of molecular replacement. The calculation of phases from a partial model is currently problematic owing to model bias. The ability to specify, on a point-by-point basis, the uncertainties in a model could substantially improve the quality of phasing that can be obtained. A third is in automated model building. The approach described here for identification of α-helices and incorporation of model information into density modification is essentially the first step in automated model building. The iterative approaches incorporated into *ARP* and *wARP* (Perrakis *et al.*, 1999) could be modified to incorporate the likelihood functions we have described here.

The author would like to thank Joel Berendzen for helpful discussions and the NIH and the US Department of Energy for generous support.

References

Abrahams, J. P. (1997). *Acta Cryst.* D**53**, 371–376.
Abrahams, J. P. & Leslie, A. G. W. (1996). *Acta Cryst.* D**52**, 30–32.
Baker, D., Bystroff, C., Fletterick, R. J. & Agard, D. A. (1993). *Acta Cryst.* D**49**, 429–439.
Béran, P. & Szöke, A. (1995). *Acta Cryst.* A**51**, 20–27.
Bricogne, G. (1984). *Acta Cryst.* A**40**, 410–445.
Bricogne, G. (1988). *Acta Cryst.* A**44**, 517–545.
Chang, W. R., Jiang, T., Wan, Z. L., Zhang, J. P., Yang, Z. X. & Liang, D. C. (1996). *J. Mol. Biol.* **262**, 721–731.
Cowtan, K. D. (1998). *Acta Cryst.* D**54**, 750–756.
Cowtan, K. D. & Main, P. (1993). *Acta Cryst.* D**49**, 148–157.
Cowtan, K. D. & Main, P. (1996). *Acta Cryst.* D**52**, 43–48.
Giacovazzo, C. & Siliqi, D. (1997). *Acta Cryst.* A**53**, 789–798.
Goldstein, A. & Zhang, K. Y. J. (1998). *Acta Cryst.* D**54**, 1230–1244.
Gu, Y., Zheng, C., Zhao, Y., Ke, H. & Fan, H. (1997). *Acta Cryst.* D**53**, 792–794.
Huber, A. H., Nelson, W. J. & Weis, W. I. (1997). *Cell*, **90**, 871–882.
Kleywegt, G. J. & Jones, T.A. (1997). *Acta Cryst.* D**53**, 179–185.
Kleywegt, G. J. & Jones, T.A. (1998). *Acta Cryst.* D**54**, 1119–1131.
Leslie, A. G. W. (1987). *Proceedings of the CCP4 Study Weekend*, pp. 25–31. Warrington: Daresbury Laboratory.
Lunin, V. Y. (1993). *Acta Cryst.* D**49**, 90–99.
Perrakis, A., Morris, R. & Lamzin, V. S. (1999). *Nature Struct. Biol.* **6**, 458–463.
Perrakis, A., Sixma, T. K., Wilson, K. S. & Lamzin, V. S. (1997). *Acta Cryst.* D**53**, 448–455.
Podjarny, A. D., Bhat, T. N. & Zwick, M. (1987). *Annu. Rev. Biophys. Biophys. Chem.* **16**, 351–373.
Prince, E., Sjolin, L. & Alenljung, R. (1988). *Acta Cryst.* A**44**, 216–222.
Read, R. J. (1986). *Acta Cryst.* A**42**, 140–149.
Refaat, L. S., Tate, C. & Woolfson, M. M. (1996). *Acta Cryst.* D**52**, 252–256.
Roberts, A. L. U. & Brünger, A. T. (1995). *Acta Cryst.* D**51**, 990–1002.
Rossmann, M. G. & Arnold, E. (1993). *International Tables for Crystallography*, Vol. B, edited by U. Shmueli, pp. 230–258. Dordrecht: Kluwer Academic Publishers.
Shneerson, V. L., Wild, D. L. & Saldin, D. K. (2001). *Acta Cryst.* A**57**, 163–175.
Szöke, A. (1993). *Acta Cryst.* A**49**, 853–866.
Szöke, A., Szöke, H. & Somoza, J. R. (1997). *Acta Cryst.* A**53,** 291–313.
Terwilliger, T. C. (1999). *Acta Cryst.* D**55**, 1863–1871.
Terwilliger, T. C. (2000). *Acta Cryst.* D**56**, 965–972.
Terwilliger, T. C. & Berendzen, J. (1996). *Acta Cryst.* D**51**, 609–618.
Vellieux, F. M. D. A. P., Hunt, J. F., Roy, S. & Read, R. J. (1995). *J. Appl. Cryst.* **28**, 347–351.
Wang, B.-C. (1985). *Methods Enzymol.* **115**, 90–112.
Wilson, A. J. C. (1949). *Acta Cryst.* **2**, 318–321.
Wilson, C. & Agard, D. A. (1993). *Acta Cryst.* A**49**, 97–104.
Xiang, S., Carter, C. W. Jr, Bricogne, G. & Gilmore, C. J. (1993). *Acta Cryst.* D**49**, 193–212.
Zhang, K. Y. J. (1993). *Acta Cryst.* D**49**, 213–222.
Zhang, K. Y. J., Cowtan, K. D. & Main, P. (1997). *Methods Enzymol.* **277**, 53–64.
Zhang, K. Y. J. & Main, P. (1990). *Acta Cryst.* A**46**, 41–46.

electronic reprint

electronic reprint

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Map-likelihood phasing

Thomas C. Terwilliger

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Map-likelihood phasing

Thomas C. Terwilliger

Bioscience Division, Mail Stop M888, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence e-mail: terwilliger@lanl.gov

Received 17 April 2001
Accepted 17 August 2001

The recently developed technique of maximum-likelihood density modification [Terwilliger (2000), *Acta Cryst.* D**56**, 965–972] allows a calculation of phase probabilities based on the likelihood of the electron-density map to be carried out separately from the calculation of any prior phase probabilities. Here, it is shown that phase-probability distributions calculated from the map-likelihood function alone can be highly accurate and that they show minimal bias towards the phases used to initiate the calculation. Map-likelihood phase probabilities depend upon expected characteristics of the electron-density map, such as a defined solvent region and expected electron-density distributions within the solvent region and the region occupied by a macromolecule. In the simplest case, map-likelihood phase-probability distributions are largely based on the flatness of the solvent region. Though map-likelihood phases can be calculated without prior phase information, they are greatly enhanced by high-quality starting phases. This leads to the technique of prime-and-switch phasing for removing model bias. In prime-and-switch phasing, biased phases such as those from a model are used to prime or initiate map-likelihood phasing, then final phases are obtained from map-likelihood phasing alone. Map-likelihood phasing can be applied in cases with solvent content as low as 30%. Potential applications of map-likelihood phasing include unbiased phase calculation from molecular-replacement models, iterative model building, unbiased electron-density maps for cases where $2F_o - F_c$ or σ_A-weighted maps would currently be used, structure validation and *ab initio* phase determination from solvent masks, non-crystallographic symmetry or other knowledge about expected electron density.

1. Introduction

Density-modification techniques are a firmly established and important tool for macromolecular structure determination. These methods include such powerful approaches as solvent flattening, non-crystallographic symmetry averaging, histogram matching, phase extension, molecular replacement, entropy maximization and iterative model building (Abrahams, 1996; Béran & Szöke, 1995; Bricogne, 1984, 1988; Cowtan & Main, 1993, 1996; Giacovazzo & Siliqi, 1997; Goldstein & Zhang, 1998; Gu *et al.*, 1997; Lunin, 1993; Perrakis *et al.*, 1997; Podjarny *et al.*, 1987; Prince *et al.*, 1988; Refaat *et al.*, 1996; Roberts & Brünger, 1995; Rossmann & Arnold, 1993; Szöke, 1993; Szöke *et al.*, 1997; Terwilliger, 2000; van der Plas & Millane, 2000; Vellieux *et al.*, 1995; Wang, 1985; Wilson & Agard, 1993; Xiang *et al.*, 1993; Zhang & Main, 1990; Zhang, 1993; Zhang *et al.*, 1997). The central basis of these

© 2001 International Union of Crystallography
Printed in Denmark – all rights reserved

electronic reprint

approaches is that prior knowledge about expected values of the electron density in part or all of the unit cell can be a very strong constraint on the crystallographic structure factors. For example, prior knowledge about electron density often consists of the identification of a region where the electron density is flat owing to the presence of disordered solvent (Wang, 1985). Real-space information of this kind has generally been used to improve the quality of crystallographic phases obtained by other means, such as multiple isomorphous replacement or multiwavelength experiments, but phase information from such real-space constraints can sometimes be so powerful as to be useful in *ab initio* phase determination (Béran & Szöke, 1995; van der Plas & Millane, 2000; Wang *et al.*, 1998; Roversi *et al.*, 2000).

2. Maximum-likelihood density modification

We recently developed maximum-likelihood density modification, a method for carrying out density modification in which the phasing information coming from various sources is explicitly kept separate (Terwilliger, 1999, 2000). This separation of phasing information allowed a statistical formulation for density modification that was very straightforward and avoided major existing difficulties with density modification. In maximum-likelihood density modification, the total likelihood of a set of structure factors $\{F_{\mathbf{h}}\}$ is defined in terms of three quantities: (i) any prior knowledge from other sources about these structure factors, (ii) the likelihood of measuring the observed set of structure factors $\{F_{\mathbf{h}}^{\rm OBS}\}$ if this set of structure factors were correct and (iii) the likelihood that the map resulting from this set of structure factors $\{F_{\mathbf{h}}\}$ is consistent with our prior knowledge about this and other macromolecular structures. This can be written as

$$\mathrm{LL}(\{F_{\mathbf{h}}\}) = \mathrm{LL}^{o}(\{F_{\mathbf{h}}\}) + \mathrm{LL}^{\rm OBS}(\{F_{\mathbf{h}}\}) + \mathrm{LL}^{\rm MAP}(\{F_{\mathbf{h}}\}), \quad (1)$$

where $\mathrm{LL}(\{F_{\mathbf{h}}\})$ is the log-likelihood of a possible set of crystallographic structure factors $F_{\mathbf{h}}$, $\mathrm{LL}^{o}(\{F_{\mathbf{h}}\})$ is the log-likelihood of these structure factors based on any information that is known in advance, such as the distribution of intensities of structure factors (Wilson, 1949), $\mathrm{LL}^{\rm OBS}(\{F_{\mathbf{h}}\})$ is the log-likelihood of these phases given the experimental data alone and $\mathrm{LL}^{\rm MAP}(\{F_{\mathbf{h}}\})$ is the log-likelihood of the electron-density map resulting from these phases. In this formulation, density modification consists of maximizing the total likelihood given by (1).

We showed previously that the total likelihood in (1) could be maximized efficiently by an iterative procedure in which a probability distribution for each phase is calculated independently of those for all other phases in each cycle (Terwilliger, 1999, 2000). In one cycle of optimization, an electron density map is calculated using current estimates of the structure factors. Each structure factor is then considered separately from the others and a phase-probability distribution for that structure factor is calculated from the variation of the total likelihood in (1) with the phase (or phase and amplitude) of that structure factor.

3. Map-likelihood phasing

In previous applications of the maximum-likelihood density-modification approach, phase information was derived from a combination of experimental probabilities and from the characteristics of the map (Terwilliger, 1999, 2000). In principle, however, experimentally derived or other prior phase information need not necessarily be included in the maximum-likelihood density-modification procedure. Instead, phase information could be derived from the agreement of the map with expectations alone.

The overall procedure for one cycle of map-likelihood phasing has five basic steps which are based on the methods used for maximum-likelihood density modification (Terwilliger, 2000). First, a starting set of phases is used to calculate a figure-of-merit-weighted electron-density map. This map is important because a comparison of this map with expected electron-density distributions in the unit cell will form the basis for the determination of phase probabilities. Next, the expectations about the electron-density distributions in this map are evaluated. As described in more detail below, this can consist of probability distributions for electron density in the protein and solvent regions along with probability estimates of whether each point in the map is within the protein or solvent region, for example. These probability distributions are crucial for defining the prior expectations about the electron-density map and therefore the log-likelihood of the map. Third, the log-likelihood of this map and the first and second derivatives of this log-likelihood with respect to electron density at each point in the map are calculated. These derivatives will be used to predict how the log-likelihood of the map will change as the electron density in the map is changed. Fourth, using the chain rule and an FFT-based algorithm, the first and second derivatives of the log-likelihood of the map with respect to structure factors are calculated. Fifth, for each reflection k the variation of the log-likelihood of the map with the phase (or phase and amplitude) of the reflection is estimated from these derivatives. This is the key step in map-likelihood phasing. Through the use of the derivatives of the log-likelihood of the map with respect to the structure factor k, map-likelihood phasing allows relative probabilities to be assigned for each possible value of the phase of reflection k. These phase probabilities are used to estimate a new weighted mean estimate of the structure factor k.

In this calculation of the phase-probability distribution for reflection k, ordinarily the measured amplitude is kept fixed and the allowed phases for this reflection are sampled at regular intervals (typically increments of 5° for acentric reflections). The log-likelihood of the map is approximated in terms of a Taylor's series based on the derivatives with respect to structure factors as described previously (Terwilliger, 2000), with the addition of a cross-term in the Taylor's series as suggested by Cowtan (2000). To the extent that this approximation is accurate (that is, that higher order terms do not contribute substantially), this phase-probability calculation estimates how the log-likelihood of the map will vary with the phase of reflection k without regard to the value of the phase

electronic reprint

that was used to calculate the original electron-density map. Once all five steps in map-likelihood phasing are carried out, it is possible to calculate a new figure-of-merit-weighted electron-density map using the newly estimated phase-probability distributions. These phases can then be used to initiate a new cycle of map-likelihood phasing. As the phases are modified in this fashion, it is useful to update the analysis of the probability estimates for whether each point in the map is in the protein or solvent region and any other analyses based on the map.

The effect of each cycle in this procedure is to obtain a probability distribution for each phase independently of all the others, based on the agreement of the electron-density map with expectations. In the phase-probability calculations, all possible values of the phases are considered without any preference for the values used in the previous cycle.

The iteration of phasing and analysis of the map is ordinarily repeated until phase changes are minimal. As described below, however, in some cases where there is relatively little phase information available from the map-likelihood function it is useful to end the iterative process before complete convergence. Also, in some cases iterations of this procedure lead to some oscillations in which the changes in the weighted mean estimate of structure factor k are strongly anticorrelated from one cycle to the next. In such cases a damping factor (typically 0.5) can be applied to the changes from one cycle to the next to reduce the oscillations.

Map-likelihood phasing is related to the methods of Béran & Szöke (1995) and to the application of NCS phase refinement starting from a solvent mask (Braik *et al.*, 1994; Kleywegt & Read, 1997; van der Plas & Millane, 2000; Wang *et al.*, 1998) in which crystallographic phases are obtained by matching the electron density in a part of the unit cell to a target value. The method of Béran & Szöke (1995), which employs simulated annealing to find a set of phases consistent with constraints on electron density, was shown to be capable of *ab initio* phase determination using a solvent mask. Similarly, high-order non-crystallographic symmetry has been shown to be sufficient to determine crystallographic phases starting just from a solvent mask (Braik *et al.*, 1994; Kleywegt & Read, 1997; van der Plas & Millane, 2000; Wang *et al.*, 1998). The maximum-likelihood approach described here and in Terwilliger (2000) differs from the methods of Béran & Szöke (1995), van der Plas & Millane (2000) and Wang *et al.* (1998) in that probabilistic descriptions of the expected electron density are used, allowing a calculation of phase-probability distributions, rather than searching for a set of phases that is consistent with constraints.

The phase information coming from the map-likelihood function $\mathrm{LL}^{\mathrm{MAP}}(\{F_{\mathbf{h}}\})$ comes from the agreement of the electron-density map with prior expectations about that map. This agreement depends on the phase of each reflection, in the context of the phases of all other reflections. In the implementation used in maximum-likelihood density modification, the probability (based on the map likelihood) for a particular structure factor that the phase has a value φ is given by the relative likelihood of the map obtained with this value of the phase. For example, a simple map-likelihood function might be based on defined regions containing the macromolecule and containing disordered solvent. A value of the phase for a particular reflection k that leads to a map with a relatively flat solvent is more likely to be correct than a phase that does not.

In a more general case, a map-likelihood function can be defined that describes solvent and 'protein' regions of the electron-density map and probability distributions for electron density in each such region. The probability of a particular phase for a particular reflection can then be estimated from how well the resulting map matches these expected characteristics. The concept can also be extended further to include non-crystallographic symmetry. A map-likelihood function could be constructed that reflects the extent to which symmetry-related density in the map is indeed similar, for example.

A formulation of the map log-likelihood function $\mathrm{LL}^{\mathrm{MAP}}(\{F_{\mathbf{h}}\})$ that follows this approach (Terwilliger, 2000) can be written as the integral over the map of a local log-likelihood of electron density, $\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]$,

$$\mathrm{LL}^{\mathrm{MAP}}(\{F_{\mathbf{h}}\}) \simeq \frac{N_{\mathrm{REF}}}{V} \int_V \mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})]\, \mathrm{d}^3\mathbf{x}. \qquad (2)$$

where this local log-likelihood of electron density describes the plausibility of the map at each point.

The local log-likelihood function, in turn, can be expressed in terms of whether the point is in the solvent or protein regions and the expected electron-density distributions in each case. As it is often uncertain whether a particular point $\mathbf{x}$ is in the protein or solvent region, it is useful to write the local map-likelihood function as the sum of conditional probabilities dependent on which environment the point is located in,

$$\mathrm{LL}[\rho(\mathbf{x}, \{F_{\mathbf{h}}\})] = \ln\{p[\rho(\mathbf{x})|\mathrm{PROT}]p_{\mathrm{PROT}}(\mathbf{x}) + p[\rho(\mathbf{x})|\mathrm{SOLV}]p_{\mathrm{SOLV}}(\mathbf{x})\}, \qquad (3)$$

where $p_{\mathrm{PROT}}(\mathbf{x})$ is the probability that $\mathbf{x}$ is in the protein region, $p[\rho(\mathbf{x})|\mathrm{PROT}]$ is the conditional probability for $\rho(\mathbf{x})$ given that $\mathbf{x}$ is in the protein region and $p_{\mathrm{SOLV}}(\mathbf{x})$ and $p[\rho(\mathbf{x})|\mathrm{SOLV}]$ are the corresponding quantities for the solvent region. The probability that $\mathbf{x}$ is in the protein or solvent regions can estimated by a modification of the methods of Wang (1985) and Leslie (1987) as described earlier (Terwilliger, 1999) or by other probability-based methods (Roversi *et al.*, 2000).

The probability distributions for electron density given that a point is in the protein or solvent regions are central to map-based phasing. They define the expectations about electron density in the map. These expectations about electron-density distributions in the map are not derived from 'perfect' maps, but rather from the current electron-density map. There are several reasons for doing this (Terwilliger, 2000). The key reason is that it is unreasonable to expect any value of the phase for a particular reflection to lead to a map matching expectations of a perfect map because the map has large errors from all the other reflections. In particular, the correct value of the phase for reflection k can only be expected to

electronic reprint

reduce the variation in the solvent region only slightly, not to make it perfectly flat. The amount by which the solvent can be expected to be flattened by adjusting just one reflection is dependent on the overall noise in the map. In effect, the expectations about the electron-density map include not just the features of a perfect map, but also effects of the errors in all of the structure factors other than the one under consideration. Consequently, for a starting phase set with large phase errors, the target probability distribution of electron density in the solvent region is very broad, while for a starting phase set that is very accurate this distribution can be very narrow.

Because the targeted features of the electron-density map are only weakly defined for poor starting phase sets but are more precisely defined for accurate ones, the phase information coming from the map-likelihood function becomes stronger as the phases improve. In essence, the more accurate the starting phases and the less noise in the map, the more precisely the phase of a particular reflection can be expected to lead to a map that matches the characteristics of a perfect map, and the more precisely the values of each phase can be determined.

4. Bias and map-likelihood phases

Somewhat paradoxically, although the quality of the starting phase set is an important factor in determining the phase information that comes from the map, the phase probability for a reflection obtained from map-likelihood phasing is completely unbiased with respect to the prior probabilities for that phase. On the other hand, the map-likelihood phase probability for a reflection can be biased by a model used to calculate all starting phases.

To see how the map-likelihood phase for a reflection can be unbiased with respect to prior probabilities for that phase, consider using map-likelihood phasing to obtain a probability distribution for the phase of reflection k. In order to make the situation clear, the procedure described will be a little simpler than the one used in practice. First, calculate an electron-density map using all reflections other than k. This map clearly has no bias towards the prior value for reflection k, as reflection k was not even used to obtain the map. Now examine all possible phases of the reflection k in question. For each phase, add to the map the electron density that would result from reflection k with this phase. Then compare the characteristics of the resulting electron-density map with the ones that we expect, given the location of solvent and macromolecule and given the expected distributions of electron density in solvent and protein regions. Some values of the phase of reflection k will generally lead to more plausible maps than others. This defines our probability distribution for the phase of reflection k and the process has made no use whatsoever of any prior information about this reflection. Consequently, the resulting phases are completely unbiased with respect to any prior information about reflection k. In practice, this cross-validation procedure is carried out with all the reflections at once employing an approximation and an FFT-based method (Terwilliger, 2000). The resulting phase-probability distributions are essentially the same as the ones described above, however.

Each individual phase-probability distribution obtained with map-likelihood phasing is independent of the prior phase-probability distribution for that reflection. Nevertheless, there are kinds of bias that can affect map-likelihood phasing. If the set of phases used to initiate map-likelihood phasing has been adjusted as a whole in a way that leads to a relatively flat solvent region, for example, then the first few cycles of map-likelihood phasing will tend to find these starting phases to be probable ones (because they lead to a flat solvent when combined with all the other starting phases) even if these starting phases are incorrect. This situation can occur for example if a model has been used to calculate the starting phases, as the solvent region will tend to be relatively flat even if the model is not entirely correct. It can also occur if the phases have been refined in order to flatten the solvent region. Fortunately, as described below, this type of model bias is generally removed by iterative application of map-likelihood phasing.

As described above, other approaches to using expectations about electron-density distributions in a map for determining crystallographic phases without including phase-probability distributions from other sources have been demonstrated (Wang *et al.*, 1998; van der Plas & Millane, 2000; Béran & Szöke, 1995). Each of these approaches begins with no prior phase information and is designed to result in an *ab initio* phase determination. These approaches could be modified to begin with a starting phase set as described here for map-likelihood phasing; however, the probability-based approach described here is more general and can include a variety of expectations about the map. Additionally, map-likelihood phasing leads to phase-probability distributions rather than phases consistent with expectations, so that optimally weighted maps can be calculated.

5. Prime-and-switch phasing to remove model bias

Model bias is a very serious problem in macromolecular crystallography (Adams *et al.*, 1999; Hodel *et al.*, 1992; Kleywegt, 2000). A bias in phases that leads to electron-density patterns that are incorrect, yet look like features of a macromolecule, is very difficult to detect. Such a bias is much more serious than an equivalent amount of noise in a map that is distributed in a random fashion in the unit cell. Bias of this kind commonly occurs when crystallographic phases are calculated based on a model that contains atoms that are incorrectly placed. Maps that are based on these phases tend to show peaks at the positions of these atoms even if the correct electron density would not.

Model bias in an electron-density map does not necessarily mean that the phases are very inaccurate. Relatively accurate phases, calculated from a largely correct model with some atoms in incorrect locations, can still lead to peaks at the coordinates of the incorrectly placed atoms. In this sense, the phases are biased. In the sense that the phases are close to the true phases, the phases are still relatively accurate. As in many

electronic reprint

situations, there is an important trade-off between accuracy and bias in the calculation of electron-density maps. In the crystallographic case, this trade-off is fundamentally related to the difference between errors in electron density that are randomly distributed about the unit cell and those that are focused on certain locations in the cell. In many situations, the most accurate map (the one with the minimum expected mean-square error, for example) will be one that is based on all available information. Unfortunately, in the crystallographic situation the errors in such a map can be highly non-random. As mentioned above, high peaks can be obtained at the specific positions of atoms in a model used to calculate phases, even if those atoms are incorrectly placed. Such a map, though accurate, can be highly misleading. A map that is less accurate, but that does not suffer from this bias, could in many cases be far more informative.

Many methods for reducing model bias in electron density maps have been developed. One of the most widely used approaches is the σ_A method of Read (1986), in which the weighting and amplitudes of structure factors (but not the phases) are optimized for minimizing effects of model bias. As the phases remain based on the model, σ_A weighting retains some model bias (Hodel *et al.*, 1992). Another important method is the use of omit maps, in which all atoms in a region of the unit cell in the model are removed before using the model to calculate phases. This method reduces model bias, but leads to electron-density maps that are intrinsically much noisier than those calculated with all atoms present. Omit maps can still contain some model bias despite the omission of atoms in a region of space, as refinement can adjust the parameters describing all the other atoms in such a way as to leave a 'memory' of the coordinates of the omitted atoms (Hodel *et al.*, 1992). This memory in omit maps corresponds to the model bias described above that can occur in the first few cycles of map-likelihood phasing. The residual bias in omit maps can be reduced by simulated annealing of the partial model (Hodel *et al.*, 1992), if the resolution of the data and the accuracy of the starting model allows atomic refinement. In general terms, this corresponds to the iterative application of map-likelihood phasing to remove residual bias. Maximum-likelihood refinement of the model structure can also be used to reduce model bias even in cases where σ_A-weighted electron-density maps are not interpretable (Adams *et al.*, 1999).

The technique of prime-and-switch phasing takes advantage of the lack of bias in map-likelihood phasing and the strong dependence of the accuracy of map-likelihood phases on the quality of the phases used to initiate the process discussed above. In this technique, all available phase information, including any coming from a model, is used to initiate map-likelihood phasing. The model phases are then set aside and not used further. As discussed above, model-based phases can be relatively accurate and biased at the same time. Owing to their accuracy, they can be useful in initiating map-likelihood phasing. Owing to their bias, setting them aside during map-likelihood phasing can reduce the bias in the final phases. As prime-and-switch phasing does not use all the phase information available, the final phases could be less accurate than a set that could be obtained using all this information. As shown below, in most cases any loss of accuracy is compensated for by a corresponding decrease in bias.

Map-likelihood phasing has the potential for producing electron-density maps that have little or no bias, as the phase probabilities for each reflection are independent of the prior phases for that reflection. However, as described above, it is possible for map-likelihood phasing to be biased by a starting phase set that has a systematic bias, for example by a starting set of incorrect phases that has a relatively flat solvent region. The iteration of cycles of map-likelihood phasing is a useful tool in reducing or eliminating this bias. The reason for expecting that an iterative application of map-likelihood phasing would remove the bias present in a single cycle is that the bias for an individual reflection comes from the set of starting phases as a whole. Once many of the phases in the set are substantially changed, the bias might be greatly reduced.

6. Convergence of map-likelihood phasing

There are two general cases that could arise in carrying out iterative cycles of map-likelihood phasing. If the solvent content or non-crystallographic symmetry are high, then the phases are likely to be well determined and simple iterative map-likelihood phasing would be effective. If the solvent content is low and non-crystallographic symmetry is lacking, however, the phases might not be entirely determined by the map-likelihood function. In this case, it might be necessary to trade off a small bias towards the starting phase set in order to obtain a well defined set of phases. One way to carry out such a trade-off is simply to end the iteration of map-likelihood phasing before complete convergence. This is generally the preferable alternative in practice because, as shown below, map-likelihood phases often change very rapidly during early cycles then only very gradually after that.

An alternate procedure is to introduce a small weighting towards the model-based prior phases. This introduction of some model-based phase information has several effects. One is to reintroduce some bias into the final phases. A second is to stabilize the phasing process. A third is (at least potentially) to increase the overall accuracy of the phases. The degree of bias towards the starting phase set in map-likelihood phasing can be adjusted using a weight on the prior phase probabilities. In cases where the phase information in the map is insufficient to fully define the phases (such as substantially less than 50% solvent content with no non-crystallographic symmetry), it is sometimes useful to trade off a small amount of bias in order to increase the stability of the iterative phasing process. This can typically be accomplished with a weighting of a few percent on the prior phase-probability distribution.

7. The bias ratio α_B

A useful measure of the degree of bias towards model-based or other prior phases used to initiate prime-and-switch

electronic reprint

phasing or used as a source of combined phase information is the bias ratio α_B, defined as

$$\alpha_B = \frac{\langle \cos(\varphi_{\rm PRIOR} - \varphi_{\rm MAP}) \rangle}{\langle m_{\rm PRIOR} \rangle \langle m_{\rm MAP} \rangle}, \qquad (4)$$

where $\varphi_{\rm PRIOR}$ is the centroid phase based on prior phase information, $\varphi_{\rm MAP}$ is that based on the map-likelihood function, $\langle m_{\rm PRIOR} \rangle$ and $\langle m_{\rm MAP} \rangle$ are estimates of $\langle \cos(\varphi_{\rm PRIOR} - \varphi_{\rm TRUE}) \rangle$ and $\langle \cos(_{\rm MAP} - \varphi_{\rm TRUE}) \rangle$, the mean cosines of the phase differences between the true phase $\varphi_{\rm TRUE}$ and $\varphi_{\rm PRIOR}$ or $\varphi_{\rm MAP}$, respectively, and the averages are taken over all reflections.

If the estimates of $\langle m_{\rm PRIOR} \rangle$ and $\langle m_{\rm MAP} \rangle$ are reasonably accurate, the bias ratio α_B can be a useful measure of the extent of correlation between the prior phases $\varphi_{\rm PRIOR}$ and the map-likelihood phases $\varphi_{\rm MAP}$, compared with the correlation expected for two independent sources of phasing. The numerator $\langle \cos(\varphi_{\rm PRIOR} - \varphi_{\rm MAP}) \rangle$ is a measure of the actual correlation between prior and map-likelihood phases. If these sources of phase information are independent, then the errors in each are independent and we can write that

$$\langle \cos(\varphi_{\rm PRIOR} - \varphi_{\rm MAP}) \rangle \simeq \langle \cos(\varphi_{\rm PRIOR} - \varphi_{\rm TRUE}) \rangle \times \langle \cos(\varphi_{\rm MAP} - \varphi_{\rm TRUE}) \rangle \qquad (5)$$

or

$$\langle \cos(\varphi_{\rm PRIOR} - \varphi_{\rm MAP}) \rangle \simeq \langle m_{\rm PRIOR} \rangle \langle m_{\rm MAP} \rangle, \qquad (6)$$

leading to a bias ratio α_B of about unity. In contrast, if the sources of phase information are not independent, such as might occur if the bias in the model-based phases was not completely removed by iterative map-likelihood phasing, then the correlation between prior and map-likelihood phases will typically be greater than when they are independent,

$$\langle \cos(\varphi_{\rm PRIOR} - \varphi_{\rm MAP}) \rangle > \langle \cos(\varphi_{\rm PRIOR} - \varphi_{\rm TRUE}) \rangle \times \langle \cos(\varphi_{\rm MAP} - \varphi_{\rm TRUE}) \rangle, \qquad (7)$$

and the bias ratio α_B will generally be greater than unity.

The utility of the bias ratio is dependent on having reasonable estimates of the figures of merit of phasing $\langle m_{\rm PRIOR} \rangle$ and $\langle m_{\rm MAP} \rangle$ for the prior and map-likelihood phases. If these are overestimated, for example, then the bias ratio will be underestimated. In an extreme case, a bias ratio of substantially less than unity can be used as a diagnostic for overestimated values of the figures of merit of one or both of these sources of phasing. The bias ratio can potentially be used in a third approach for handling situations where the map-likelihood phase information is insufficient in itself to fully define the phases. In this approach, iterations of map-likelihood phasing are carried out until the bias ratio reaches a value of approximately unity, indicating that much of the bias from the prior model-based phases has been removed.

8. Examples of map-likelihood phasing

8.1. Separation of experimental and map-likelihood phase information

Fig. 1 illustrates how the phase information in maximum-likelihood density modification can be separated into experimentally derived phase information and map-likelihood phase information. Fig. 1(*a*) shows an experimental electron-density map based on MAD phasing of initiation factor 5a (Peat *et al.*, 1998). Fig. 1(*b*) shows an electron-density map calculated from the map-likelihood phase probabilities obtained on the first cycle of maximum-likelihood density modification using the experimental map in Fig. 1(*a*) as a starting point. The crystals of initiation factor 5a contain about 60% solvent, so the phasing information that can be obtained from the map likelihood is very substantial. The map-likelihood phased map in Fig. 1(*b*) is clearly of equivalent or higher quality than the experimental map in Fig. 1(*a*). This is quite remarkable when it is recognized that the phase probabilities for the map in Fig. 1(*b*) are obtained simply by matching calculated and expected electron-density distributions in the solvent and protein regions. Fig. 1(*c*) illustrates that the accuracy of the starting phase set used in map-likelihood phasing has a substantial effect on the final phase-probability distributions. In this panel, the starting phases were those obtained after maximum-likelihood density modification with the program *RESOLVE* (Terwilliger, 2000) was applied to the data used in Fig. 1(*a*). This electron-density map is of even higher quality than those in Fig. 1(*a*) or Fig. 1(*b*).

8.2. Convergence and phase improvement in map-likelihood phasing

In order to evaluate the range of applicability of map-likelihood phasing and the utility of iterative phase improvement with this technique, several tests were carried out with model data, where the quality of phasing could readily be assessed. Figs. 2 and 3 illustrate the convergence properties of map-likelihood phasing as a function of percentage of the asymmetric unit that is occupied by disordered solvent. Model data sets were constructed based on the refined structure of dehalogenase enzyme from *Rhodococcus* (Newman *et al.*, 1999) to a resolution of 3 Å as described in Terwilliger (2000). To simulate varying amounts of solvent, varying numbers of water molecules and C-terminal residues were left out of the phase calculations. This led to models with disordered solvent content ranging from 31% (as in the actual crystals) to 73%. Starting phase sets with simulated errors were constructed and used along with the model amplitudes in map-likelihood phasing. In these simulations, a mask defining the solvent and protein regions was calculated from the atomic coordinates in the model, defining all points within 2.5 Å of an atom as being within the protein region. In each test, 20 cycles of phase calculation followed by figure-of-merit weighted map calculation were carried out. For each cycle, the mean true figure of merit, given by the cosine of the phase error $\langle \cos\Delta\varphi \rangle$ is plotted.

electronic reprint

Fig. 2(*a*) shows the effect of the percentage of the cell occupied by the macromolecule and by 'solvent' (actually simply absence of atoms in these simulations) on the phases obtained from map-likelihood phasing starting with very poor initial phases. The starting mean true figure of merit in each case was 0.32 and the data extended to a resolution of 3 Å. For simulations with about 50% solvent or greater, each cycle of map-likelihood phasing resulted in phases that were at least as accurate as those in the previous cycle, with convergence essentially complete within 20 cycles. For those with 39% solvent, the phases became slightly worse with map-likelihood phasing compared with the starting phases; for the case with 31% solvent they were considerably worse. Fig. 2(*b*) illustrates the effect of high-resolution and low-resolution cutoffs on the quality of the phasing for the simulation with 53% solvent shown in Fig. 2(*a*). When all the data from 2.8 to 20 Å are included, the final mean true figure of merit was 0.56. When data from only 2.8 to 6 Å are included, the resulting true figure of merit decreases to only 0.44 and when data from only 2.8 to 5 Å are included, to only 0.34. Conversely, when data from only 5 to 20 Å are included, the resulting true figure of merit is only 0.28 and, as high-resolution data to 2.8 Å are included, this increases to 0.56.

Fig. 3 expands on the simulation shown in Fig. 2, illustrating the stability and convergence of phasing beginning with phases with varying errors, for solvent contents of 31, 47 and 73%. In the case of 31% solvent content, for all starting phase sets the quality of phases generally decreased with each cycle of map-likelihood phasing, although when the starting true figure of merit was about 0.6 or greater, the overall phasing process was relatively stable. In contrast, for the simulation with 47% solvent the quality of phases increased slightly with each cycle. Starting from phase sets with a true figure of merit of about 0.45 or greater, all of the test simulations converged to phase sets with similar true figures of merit of about 0.6. For 73% solvent, the quality of the phases reached the same very high true figure of merit of about 0.8, regardless of the true figure of merit of the starting set of phases in the range 0.3–0.8.

Fig. 4 illustrates the effect of errors in the definition of solvent and protein regions on phasing. The simulations in this figure were carried out in the same way as those in Fig. 2, except that the mask used was based on a model that was missing about 10% of the atoms, so that about 10% of the 'protein' region was classified as 'solvent'. The quality of the

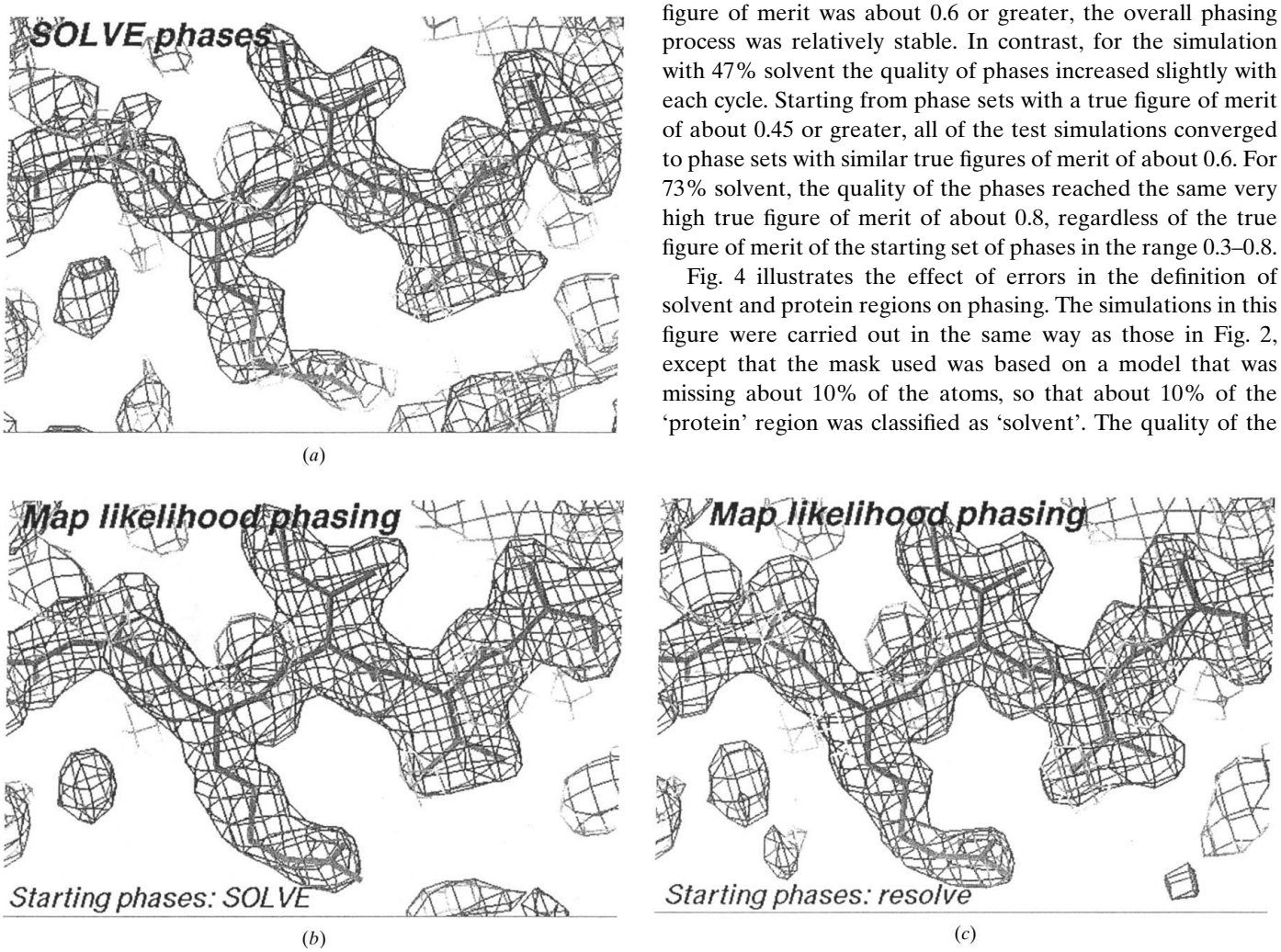


Figure 1
Map-likelihood phasing of initiation factor 5a. (*a*) Experimental (MAD) map. (*b*) Map-likelihood phased map, using experimental phases as a starting phase set. (*c*) Map-likelihood phased map, using maximum-likelihood density-modified phases as starting phase set. Experimental (MAD) phases for initiation factor 5a (Peat *et al.*, 1998) had an overall figure of merit of 0.61 to 2.2 Å. The refined model of initiation factor 5a (PDB entry 1bkb; Berman *et al.*, 2000) is overlaid on the experimental map. The map is contoured at 1.5σ. In (*b*), one cycle of map-likelihood phasing was carried out and phase probabilities were calculated from the map likelihood only. The figure of merit of map-likelihood phases was 0.37. The calculation of the solvent mask and the map-likelihood phasing was carried out as described for maximum-likelihood density modification (Terwilliger, 2000), except that experimental phases were not used in the phase-probability calculation. Experimental phases were used in determining the mask, as in maximum-likelihood density modification. In (*c*), a full set of five mask cycles of maximum-likelihood density modification, each with ten minor cycles of phase improvement, were carried out. These phases were then used to calculate a mask and to initiate one cycle of map-likelihood phasing as in (*b*). The figure of merit of map-likelihood phasing was 0.50.

electronic reprint

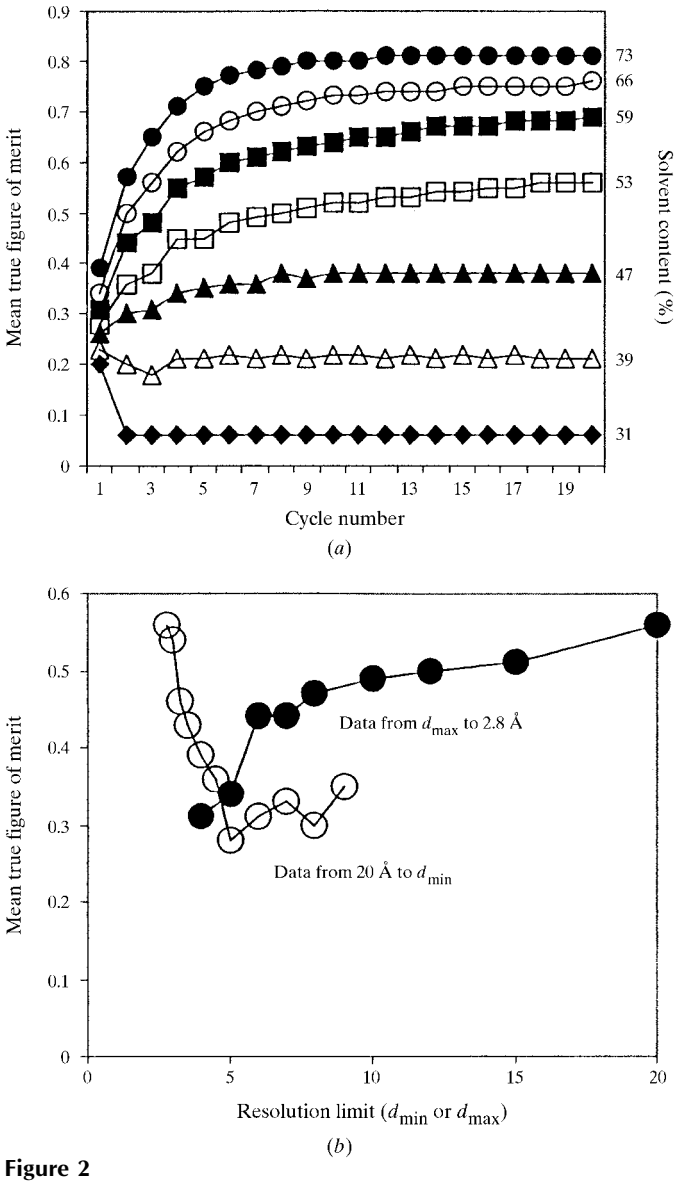


Figure 2
Effect of solvent content on map-likelihood phasing. Mean true figure of merit ($\langle\cos\Delta\varphi\rangle$) is plotted as a function of cycle number for simulations phased based on map likelihood with model cases having solvent contents of 31 (closed diamonds), 39 (open triangles), 47 (closed triangles), 53 (open squares), 59 (closed squares), 66 (open circles) and 73% (closed circles). Model phases were calculated from a model based on the refined structure of dehalogenase from *Rhodococcus* (Newman *et al.*, 1999; PDB entry 1bn7), except that varying numbers of atoms were omitted from the calculation to simulate varying solvent content. The space group is $P2_12_12$, with unit-cell parameters 93.8, 79.8 and 43.1 Å, and the resolution limits used were from 20 to 2.8 Å. The protein atoms in the simulations in Fig. 2(*a*) occupy from 27 to 69% of the unit cell, based on an algorithm in which all points within 2.5 Å of an atom in the model are considered occupied by protein. In each simulation, a starting set of phases was generated by adding Gaussian errors to the model phases so as to achieve a mean true figure of merit ($\langle\cos\Delta\varphi\rangle$) of 0.32. This starting set of phases, along with model amplitudes, was used to initiate map-likelihood phasing. Solvent masks were based on the atomic models in this simulation. In (*a*), the mean true figure of merit ($\langle\cos\Delta\varphi\rangle$) is plotted as a function of cycle number for simulations with solvent content of 31 (closed diamonds), 43 (open triangles), 47 (closed triangles), 53 (open squares), 59 (closed squares), 66 (open circles) and 73% (closed circles). In (*b*), the simulation with 53% solvent was repeated using varying high-resolution cutoffs (open circles) or low-resolution cutoff (closed circles) and the final mean true figure of merit after 20 cycles is plotted.

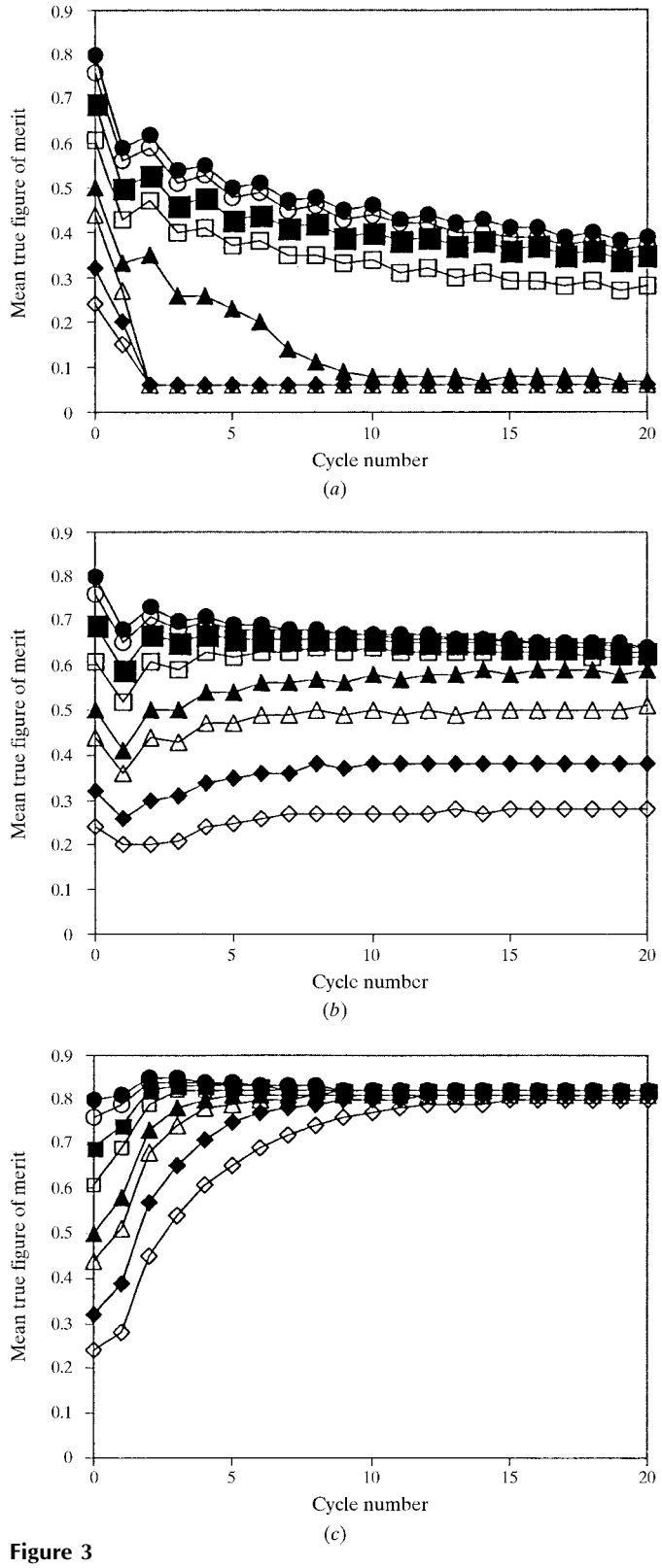


Figure 3
Effect of starting phase accuracy on map-likelihood phasing. The mean true figure of merit ($\langle\cos\Delta\varphi\rangle$) is plotted as a function of cycle number for simulations with starting mean true figure of merit of 0.24 (open diamonds), 0.32 (closed diamonds), 0.44 (open triangles), 0.50 (closed triangles), 0.61 (open squares), 0.69 (closed squares), 0.76 (open circles) and 0.80 (closed circles) and with solvent contents of (*a*) 31, (*b*) 47 and (*c*) 73%. All simulations were carried out as in Fig. 2 except for the starting phase sets.

electronic reprint

map-likelihood phases obtained was less than that obtained with the correct mask; even so, in the cases with about 50% or greater solvent content the phase quality with map-likelihood phasing improves over the starting phase set.

8.3. *Ab initio* phase determination with map-likelihood phasing

Fig. 3(*c*) showed that in cases with very high solvent content (73%), map-likelihood phasing yielded very substantial phase improvements and converged to essentially the same point regardless of the starting phase set used. Fig. 5 explores this further by illustrating the phase quality obtained by map-likelihood phasing as a function of solvent content, beginning with zero phase information (a blank map), but with a perfect solvent mask calculated from the atomic model. Fig. 5 shows that in cases with 66 and 73% solvent, map-likelihood phasing is sufficient in itself to determine crystallographic phases with high accuracy. In the model cases with 59 and 53% solvent, modest phase quality was obtained. These results are similar to those obtained by Béran & Szöke (1995) using a very different approach (simulated annealing) to find phase sets for model data that are compatible with defined solvent and protein regions.

It should be noted that although the map-likelihood approach was successful in *ab initio* phasing when using model data, tests carried out so far with experimental data have not resulted in substantial phase improvement. Presumably, this is because of complications from measurement errors and from the smaller differentiation between solvent and protein regions in real crystals compared with the model data sets examined here.

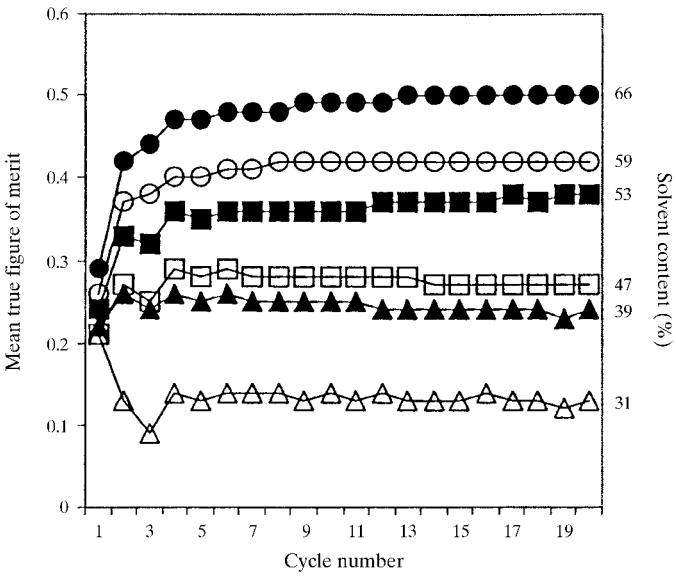


Figure 4
Effect of solvent content on map-likelihood phasing with a partially incorrect mask. Mean true figure of merit ($\langle\cos\Delta\varphi\rangle$) is plotted as a function of cycle number for simulations with solvent contents of 31 (open triangles), 39 (closed triangles), 47 (open squares), 53 (closed squares), 59 (open circles) and 66% (closed circles). In each case the solvent mask used was based on the correct atomic model, except that the last approximately 10% of atoms in the structure were omitted in order to create an incorrect mask. All simulations were carried out as in Fig. 2 except for the starting phase sets.

8.4. Reduction of model bias with prime-and-switch phasing

A very important feature of map-likelihood phasing is the potential for reducing or eliminating model bias in electron-density map calculations through the technique of prime-and-switch phasing. Test cases with model data were set up in order to examine how thoroughly model bias could be removed using prime-and-switch phasing and how this depended on the solvent content of the crystal. Additionally, the effect of including some prior phase information on bias and map quality for various solvent contents was examined.

Model data sets were constructed using the refined structure of dehalogenase enzyme from *Rhodococcus* (Newman *et al.*, 1999) and leaving out varying numbers of water molecules and atoms from the C-terminus to simulate varying amounts of solvent content as in Fig. 2. These models were considered the 'correct' structures in the tests. Then, from each correct model, a 'molecular replacement' model was constructed by varying the coordinates of atoms in the correct model by an r.m.s.d. of 1.4 Å, using a function that varied sinusoidally in space so that the connectivity of the molecule remained intact. Next, all the atoms in the molecular-replacement model that were placed incorrectly were identified by noting the value of

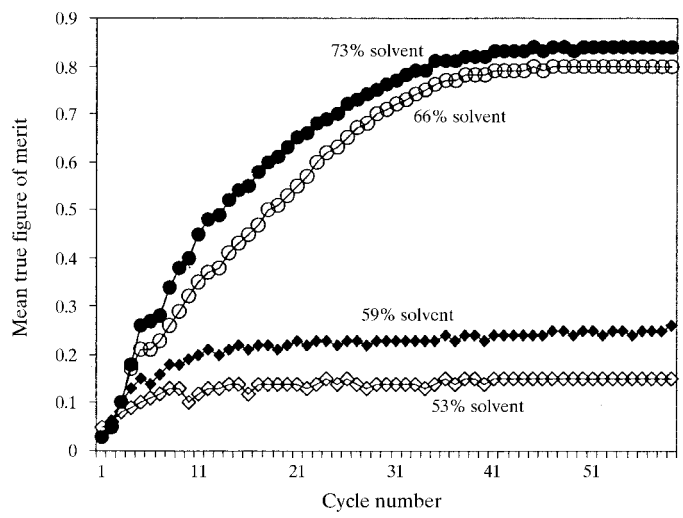


Figure 5
Effect of solvent content on map-likelihood phasing with no prior phase information. The mean true figure of merit ($\langle\cos\Delta\varphi\rangle$) is plotted as a function of cycle number for simulations with solvent contents of 53, 59, 66 and 73%. Simulations were started with zero phase information (a flat map). Starting probability distributions for electron density in the protein and solvent regions were taken from the first cycle of the simulation shown in Fig. 2(*a*) (for the simulation with 73% solvent). For all subsequent cycles, probability distributions were estimated by cross-validation as follows. The general procedure was to obtain an 'omit' map in which each point was derived from a density-modification cycle in which that point had not been included in the solvent mask. An 'omit' region was uniformly defined as 'protein', regardless of its actual location. Three cycles of density modification were carried out and the 'omit' region was saved. Omit regions covering the entire asymmetric unit were calculated and combined to make a complete 'omit' map of the asymmetric unit. This map was used to estimate probability distributions for density in the protein and solvent region for the next overall cycle.

electronic reprint

the electron density in a 'perfect' map calculated with structure factors based on the correct model. All those atoms in the molecular-replacement model that were in density from -0.5σ to 0.5σ were considered to be incorrectly placed. From 20 to 30% of the atoms in the molecular-replacement models were incorrectly placed according to this criterion. The mean density at coordinates of these incorrectly placed atoms in the perfect electron-density maps for the simulations with various solvent content ranged from 0.03σ to 0.06σ and the mean density at the coordinates of atoms in the correct model in the perfect electron-density map ranged from 1.7σ to 2.9σ, with the higher values corresponding to higher solvent contents (in which most of the cell is solvent, so the ratio of peak height to the r.m.s. density of the map is higher even with perfect data).

In the tests of model bias, the overall accuracy of electron-density maps in these tests was assessed from the normalized mean value of electron density at the coordinates of atoms in the correct model. The model bias was assessed from the normalized mean value of electron density at coordinates of incorrectly placed atoms in the molecular-replacement model used in phasing. Fig. 6(*a*) shows the overall accuracy and model bias obtained by prime-and-switch phasing (with no prior phase information included in probability calculations) as a function of the solvent content in the model crystals. For comparison, the accuracy and model bias for σ_A-weighted maps based on the same data are shown. The overall accuracy of both the σ_A-weighted and prime-and-switch phased maps was quite high in all cases, with the prime-and-switch phased maps showing greater accuracy in all cases except at very low solvent content. The σ_A-weighted maps had mean values of electron density at coordinates of atoms in the correct model ranging from 0.9σ (31% solvent) to 1.8σ (73% solvent), while the prime-and-switch phased maps had mean values of electron density at coordinates of atoms in the correct model ranging from 0.9σ (31% solvent) to 2.6σ (73% solvent).

The level of bias was very different in the two methods. The σ_A-weighted maps had mean values of electron density at coordinates of incorrectly placed atoms in the molecular-replacment model ranging from 0.5σ (31% solvent) to 1.1σ (73% solvent). In contrast, the map-likelihood phased maps had values ranging from just 0.01σ (31% solvent) to 0.13σ (73% solvent), only slightly higher than the values of 0.03σ to 0.06σ found for a perfect map. Overall, the fractional bias, the ratio of the mean values of electron density at incorrectly placed to correctly placed atoms, for σ_A-weighted maps was in the range 0.5–0.6 for all values of solvent content (Fig. 6*b*). The fractional bias using prime-and-switch phasing was in the range 0.03–0.09 for all values of the solvent content, indicating that bias was nearly eliminated in all cases.

Fig. 7 illustrates the relationship between including model-based phase information and the resulting bias in the electron-density map. The overall quality of maps and fractional bias (as in Fig. 6) for map-likelihood phasing with 31, 47 and 73% solvent and including varying amounts of prior phase information, ranging from zero weight on prior phases to equal weighting of prior phases and map-likelihood phases, are shown. For the simulations with solvent content of 31 and 47%, the overall quality of the maps generally increases as expected with inclusion of prior phase information and then slowly decreases, with mean electron density at coordinates of atoms in the perfect model with 31% solvent increasing from 0.89 (zero prior phase information) to 1.09 (10% weight on

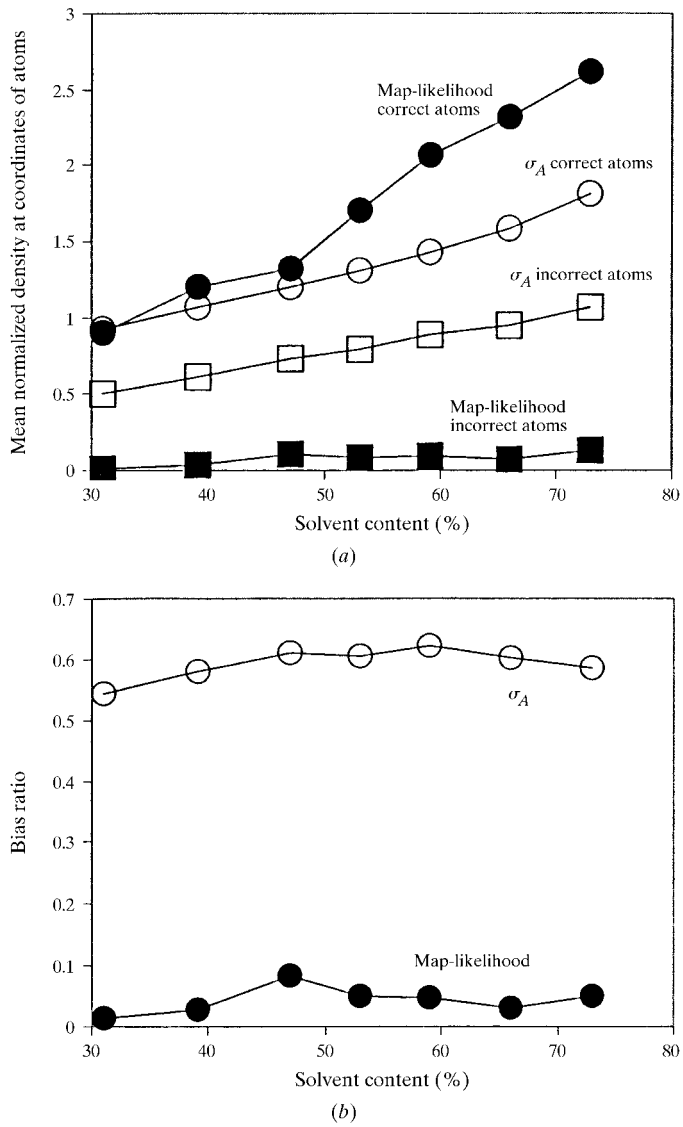


Figure 6
Bias in prime-and-switch phasing. Model data sets with varying solvent contents and 'molecular-replacement' models with atomic coordinates differing from the perfect models by an r.m.s.d. of 1.4 Å were constructed as described in the text. Phases were calculated with σ_A weighting and with prime-and-switch phasing. The prime-and-switch phasing was carried out beginning with the σ_A-weighted phases; ten cycles of solvent-mask identification each with 40 iterations of phasing were carried out. In all cases, essentially complete convergence was achieved within this number of cycles. In (*a*), the normalized electron density in the σ_A-weighted map (open circles) or prime-and-switch phased map (closed circles) at coordinates of atoms in the perfect model are shown. Additionally, the normalized electron density in the σ_A-weighted map (open squares) or prime-and-switch phased map (closed squares) at coordinates of incorrectly placed atoms in the molecular-replacement model used for phasing are shown. In (*b*), the ratio of the electron density at incorrectly placed atoms to correctly placed atoms is shown for the σ_A-weighted map (open circles) or prime-and-switch phased map (closed circles).

electronic reprint

prior information). When equal weight is placed on the prior information, overall quality decreases slightly, indicating that the prior phase-probability distributions may not be quite optimal. For the simulation with 73% solvent, inclusion of prior phase information had only a small and generally negative effect on the overall accuracy of phasing. This is presumably owing to the very high amount of unbiased phase information in the map-likelihood function in this case of high solvent content.

Fig. 8 illustrates the convergence of the map-likelihood phasing procedure as a function of the solvent content of the unit cell. In an ordinary application of map-likelihood phasing about 50 cycles of iteration would be carried out. In order to examine the convergence properties in more detail, 400 cycles were carried out for each simulation, with weights on the prior phase information ranging from zero to unity. The procedure converges rapidly for the cases with 73% solvent, requiring fewer than 50 cycles for essentially complete convergence. In the cases with 53% and with 31% solvent, convergence was not fully achieved even after 400 cycles. This illustrates cases where one of the simple procedures discussed above for stopping the iterative phasing procedure before full convergence or for the introduction of a limited amount of prior phase information to stabilize the phasing process would be applicable.

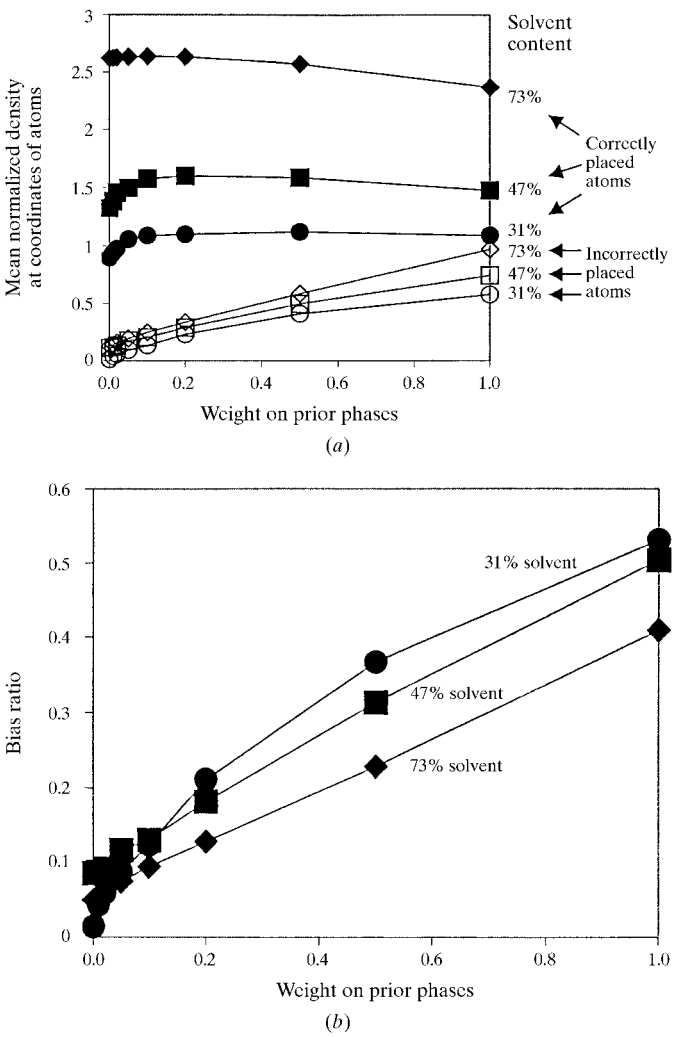


Figure 7
Effect of including prior phase information on map quality and on bias. Iterative map-likelihood phasing was carried out on the model data with 31 (circles), 47 (squares) and 73% (diamonds) solvent as in Fig. 6, except that prior σ_A-based phase probabilities were included with varying weights. The mean electron density at coordinates of atoms in the perfect model (filled symbols) and the mean density at coordinates of incorrectly placed atoms (open symbols) are shown in (*a*) and the ratio of electron density at incorrectly placed atoms to density at correctly placed atoms is shown in (*b*) (filled symbols).

8.5. Structure validation

An important application of map-likelihood phasing is likely to be structure validation (Wilson *et al.*, 1998; Kleywegt, 2000). An unbiased method of comparing a model with amplitudes of structure factors that can identify specific places in the structure that are not fully compatible with the data would be of great help in structure validation. The map-likelihood phasing method is well suited to this task as it produces phase probabilities that are essentially unbiased by the starting phase set. Fig. 9 illustrates an example of this. The structure of gene 5 protein has been determined several times, and one of the earlier structures, refined at the moderate resolution of 2.3 Å (PDB entry 2gn5; Berman *et al.*, 2000; Brayer & McPherson, 1983), differed in the loops and consequently in the register of the β-strands from structures determined at the higher resolution of 1.8 Å (PDB entry 1vqb; Skinner *et al.*, 1994) and by NMR (Folkers *et al.*, 1994).

Structure validation of PDB entry 2gn5 was carried out in two steps. The data used consisted of the atomic model 2gn5 and measured structure factors from 20 to 2.6 Å. First, the atomic model 2gn5 was used to calculate model phases and the σ_A approach of Read (1986) was used to estimate phase probability distributions for all of the structure factors. A

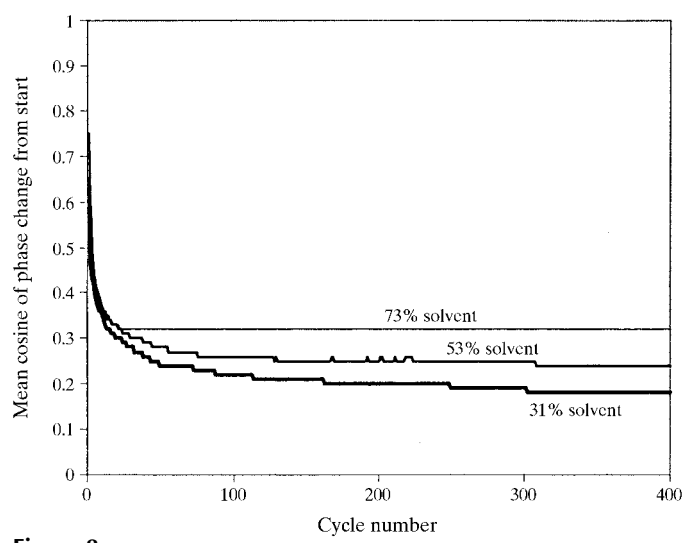


Figure 8
Convergence of prime-and-switch phasing as a function of solvent content. The mean cosine of the phase angle difference between the starting model phases and the iterative prime-and-switch phases obtained using the same data as in Fig. 6 is plotted as a function of cycle number for solvent contents of 31, 53 and 73%.

electronic reprint

region of the σ_A-weighted map containing the loop at residues 64–67 of gene 5 protein is shown in Figs. 9(*a*) and 9(*b*). In Fig. 9(*a*), the atomic model from PDB entry 2gn5 is overlaid on the map and in Fig. 9(*b*) the atomic model from the higher resolution model 1vqb is overlaid on the map. Somewhat surprisingly, considering the difference in register between the models, in general this map agrees quite well with both structures. In the region from residues 64–67, neither model fits it perfectly and neither is entirely incompatible with the map. Next, the σ_A-weighted phases were used to initiate map-likelihood phasing and five cycles of solvent-mask identification, each with ten minor cycles of phase optimization, were carried out. In this map-likelihood phasing process, the σ_A-weighted starting phases were only used to initiate the first cycle of phasing and were not used in phase-probability calculations or in subsequent cycles of phasing. The crystals of gene 5 protein contain about 40% solvent. Figs. 6(*c*) and 6(*d*) illustrate the same region shown in Figs. 6(*a*) and 6(*b*), this time with the prime-and-switch based phasing. Once again, overall the map agrees relatively well with both structures, but the prime-and-switch based phasing results in a map that is clearly more consistent with the higher resolution structure 1vqb. Figs. 6(*c*) and 6(*d*) illustrate, for example, that in the region of residues 64–67, this map shows connectivity that is in excellent agreement with the higher-resolution atomic model 1vqb, even though it is derived from the model 2gn5.

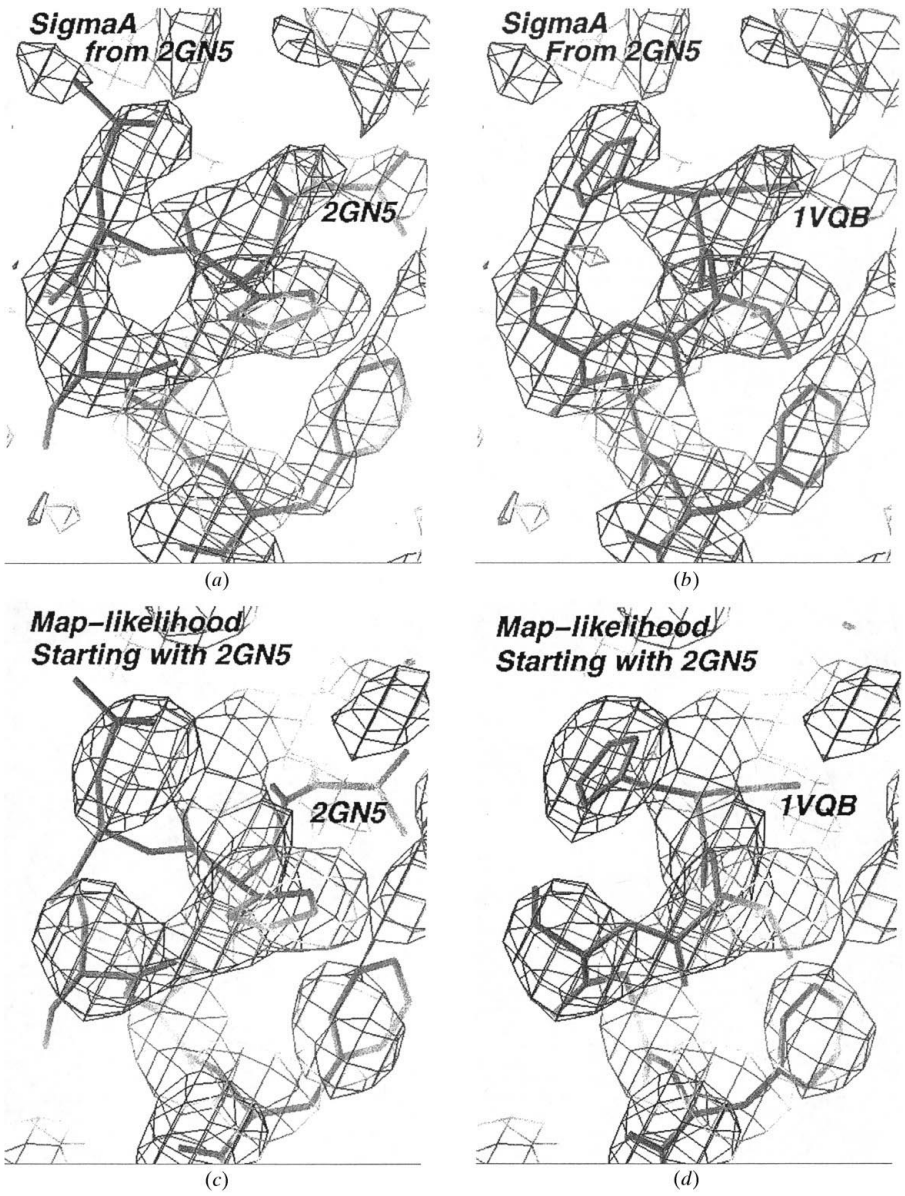


Figure 9
Structure validation with map-likelihood phasing. The atomic model in PDB entry 2gn5 was used with amplitudes of structure factors from 20 to 2.6 Å. Structure-factor amplitudes were from selenomethionine-containing gene 5 protein at $\lambda = 0.9794$; they consisted of the FP values output by *SOLVE* (Terwilliger & Berendzen, 1999) when run on the gene 5 protein MAD data (Skinner *et al.*, 1994). (*a*) and (*b*) are σ_A-weighted maps. (*c*) and (*d*) are from map-likelihood phasing as described in the text. The atomic coordinates of residues 60–70 of PDB entry 2gn5 are overlaid on the map in (*a*) and (*c*); those of the same residues from 1vqb are overlaid in (*b*) and (*d*). The maps are all contoured at 1σ.

9. Applications of map-likelihood and prime-and-switch phasing

The technique of map-likelihood phasing has potential applications in many situations in X-ray crystallography. The critical characteristics of map-likelihood phasing are (i) that it derives phase information from the agreement of features of the electron-density map with expectation and (ii) that it produces phase (or amplitude and phase) probability information that is minimally biased by the starting phase set. The phases it produces are complementary to those obtained by experimental (*e.g.* MIR, MAD) approaches because the source of phase information is completely separate (*e.g.* solvent flatness *versus* MAD measurements). For the same reason, phases are also complementary to phases calculated from a model or partial model by σ_A-based (Read, 1986) or related approaches. Prime-and-switch phasing is a special case of map-likelihood phasing in which an accurate but potentially biased source of prior phase information such as might be obtained from an atomic model is used to initiate map-likelihood phasing but then is not used further in the phasing process.

The characteristics of map-likelihood phasing make it suited for a diverse set of applications, including minimally biased phase calculations from search models in the method of molecular replacement (Rossmann, 1990, 1995),

electronic reprint

iterative model-building (Perrakis *et al.*, 1999), structure validation (Wilson *et al.*, 1998) and *ab initio* phase determination from solvent masks or non-crystallographic symmetry (Béran & Szöke, 1995; Rossmann, 1995; van der Plas & Millane, 2000; Wang *et al.*, 1998).

The approach is applicable to any situation in which phase probabilities unbiased by a starting phase set are desirable and in which some characteristics of the electron-density map can be anticipated in advance. It is most readily applied to cases where a starting set of phases exists although, as shown above, this is not required.

The accuracy of the phases obtained using map-likelihood phasing can be expected to depend largely on two factors. One is the extent of constraints that are known in advance about the electron-density map. If the structure contains a very large amount of solvent, for example, then much phase information can be obtained because electron density in the solvent region is very highly constrained. The other is the quality of the starting phase information. In an extreme case, if the phases of all reflections with significant intensities except one were known perfectly, then the phase of the final reflection could be determined perfectly because only the perfect phase would lead to a perfectly flat solvent region. In general, the higher the quality of starting phase information, the better defined the resulting probability distributions.

The author would like to thank Joel Berendzen for discussion and the NIH and the US Department of Energy for generous support. Map-likelihood phasing is available in version 2.0 of the program *RESOLVE*, available from http://resolve.lanl.gov.

References

Abrahams, J. P. & Leslie, A. G. W. (1996). *Acta Cryst.* D**52**, 30–32.
Adams, P. D., Pannu, N. S., Read, R. J. & Brunger, A. T. (1999). *Acta Cryst.* D**55**, 181–190.
Béran, P. & Szöke, A. (1995). *Acta Cryst.* A**51**, 20–27.
Berman, H. M., Westbrook, J., Feng., Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). *Nucleic Acids Res.* **28**, 235–242.
Braik, K., Otwinowski, Z., Hegde, R., Boisvert, D. D., Joachimiak, A., Horwich, A. L. & Sigler, P. B. (1994). *Nature (London)*, **371**, 578–586.
Brayer, G. D. & McPherson, A. (1983). *J. Mol. Biol.* **169**, 565–596.
Bricogne, G. (1984). *Acta Cryst.* A**40**, 410–445.
Bricogne, G. (1988). *Acta Cryst.* A**44**, 517–545.
Cowtan, K. D. (2000). *Acta Cryst.* D**56**, 1612–1621.
Cowtan, K. D. & Main, P. (1993). *Acta Cryst.* D**49**, 148–157.
Cowtan, K. D. & Main, P. (1996). *Acta Cryst.* D**52**, 43–48.
Folkers, P. J. M., Nilges, M., Folmer, R. H. A., Konings, R. N. H. & Hilbers, C. W. (1994). *J. Mol. Biol.* **236**, 229–246.
Giacovazzo, C. & Siliqi, D. (1997). *Acta Cryst.* A**53**, 789–798.
Goldstein, A. & Zhang, K. Y. J. (1998). *Acta Cryst.* D**54**, 1230–1244.
Gu, Y., Zheng, C., Zhao, Y., Ke, H. & Fan, H. (1997). *Acta Cryst.* D**53**, 792–794.
Hodel, A., Kim, S.-H. & Brünger, A. T. (1992). *Acta Cryst.* A**48** 851–858.
Kleywegt, G. J. (2000). *Acta Cryst.* D**56**, 249–265.
Kleywegt, G. J. & Read, R. J. (1997). *Structure*, **5**, 1557–1569.
Leslie, A. G. W. (1987). *Proceedings of the CCP4 Study Weekend*, pp. 25–31. Warrington: Daresbury Laboratory.
Lunin, V. Y. (1993). *Acta Cryst.* D**49**, 90–99.
Newman, J., Peat, T. S., Richard, R., Kan, L., Swanson, P. E., Affholter, J. A., Holmes, I. H., Schindler, J. F., Unkefer, C. J. & Terwilliger, T. C. (1999). *Biochemistry*, **38**, 16105–16114.
Peat, T. S., Newman, J., Waldo, G. S. Berendzen, J. & Terwilliger, T. C. (1998). *Structure*, **15**, 1207–1214.
Perrakis, A., Morris, R. & Lamzin, V. S. (1999). *Nature Struct. Biol.* **6**, 458–463.
Perrakis, A., Sixma, T. K., Wilson, K. S. & Lamzin, V. S. (1997). *Acta Cryst.* D**53**, 448–455.
Plas, J. L. van der & Millane, R. P. (2000). *Proc. SPIE*, **4123**, 249–260.
Podjarny, A. D., Bhat, T. N. & Zwick, M (1987). *Annu. Rev. Biophys. Biophys. Chem.* **16**, 351–373.
Prince, E., Sjolin, L. & Alenljung, R. (1988). *Acta Cryst.* A**44**, 216–222.
Read, R. J. (1986). *Acta Cryst.* A**42**, 140–149.
Refaat, L. S., Tate, C. & Woolfson, M. M. (1996). *Acta Cryst.* D**52**, 252–256.
Roberts, A. L. U. & Brünger, A. T. (1995). *Acta Cryst.* D**51**, 990–1002.
Rossmann, M. G. (1990). *Acta Cryst.* A**46**, 73–82.
Rossmann, M. G. (1995). *Curr. Opin. Struct. Biol.* **5**, 650–655.
Rossmann, M. G. & Arnold, E. (1993). *International Tables for Crystallography*, Vol. B. edited by U. Shmueli, pp. 230–258. Dordrecht: Kluwer Academic Publishers.
Roversi, P., Blanc, E., Vonrhein, C., Evans, G. & Bricogne, G. (2000). *Acta Cryst.* D**56**, 1316–1323.
Skinner, M. M., Zhang, H., Leschnitzer, D. H., Guan, Y., Bellamy, H., Sweet, R. M., Gray, C. W., Konings, R. N. H., Wang, A. H.-J. & Terwilliger, T. C. (1994). *Proc. Natl Acad. Sci. USA*, **91**, 2071–2075.
Szöke, A. (1993). *Acta Cryst.* A**49**, 853–866.
Szöke, A., Szöke, H. & Somoza, J. R. (1997). *Acta Cryst.* A**53**, 291–313.
Terwilliger, T. C. (1999). *Acta Cryst.* D**55**, 1863–1871.
Terwilliger, T. C. (2000). *Acta Cryst.* D**56**, 965–972.
Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**, 849–861.
Vellieux, F. M. D. A. P., Hunt, J. F., Roy, S. & Read, R. J. (1995). *J. Appl. Cryst.* **28**, 347–351.
Wang, B.-C. (1985). *Methods Enzymol.* **115**, 90–112.
Wang, J. M., Hartling, J. A. & Flanagan, J. M. (1998). *J. Struct. Biol.* **124**, 151–163.
Wilson, A. J. C. (1949). *Acta Cryst.* **2**, 318–321.
Wilson, C. & Agard, D. A. (1993). *Acta Cryst.* A**49**, 97–104.
Wilson, K. S., Butterworth, S., Dauter, Z., Lamzin, V. S., Walsh, M., Wodak, S., Pontius, J., Richelle, J., Vaguine, A., Sander, C., Hooft, R. W. W. & Vriend, G. (1998). *J. Mol. Biol.* **276**, 417–436.
Xiang, S., Carter, C. W. Jr, Bricogne, G. & Gilmore, C. J. (1993). *Acta Cryst.* D**49**, 193–212.
Zhang, K. Y. J. (1993). *Acta Cryst.* D**49**, 213–222.
Zhang, K. Y. J., Cowtan, K. D. & Main, P. (1997). *Methods Enzymol.* **277**, 53–64.
Zhang, K. Y. J. & Main, P. (1990). *Acta Cryst.* A**46**, 41–46.

electronic reprint

electronic reprint

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Statistical density modification with non-crystallographic symmetry

Thomas C. Terwilliger

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Statistical density modification with non-crystallographic symmetry

Thomas C. Terwilliger

Mail Stop M888, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence e-mail: terwilliger@lanl.gov

Received 22 April 2002
Accepted 10 September 2002

Statistical density modification is a technique for phase improvement through a calculation of the posterior probability of the phases, given experimental phase information and expectations about features of the electron-density map. The technique can take advantage of both estimates of electron density in the map and uncertainties or probability distributions for those estimates. For crystals with non-crystallographic symmetry (NCS), this allows the use of the expected similarity of electron density at NCS-related points without requiring an implicit assumption that these regions are identical.

1. Introduction

Non-crystallographic symmetry (NCS) can be a powerful aid in improving the quality of macromolecular electron-density maps (Bricogne, 1974; Rossmann, 1972; Kleywegt & Read, 1997). When present, NCS is often used along with solvent flattening (Wang, 1985) as a constraint on the electron density in a map, resulting indirectly in an improvement of the phases. Largely because it is difficult to do otherwise, in this process the NCS is generally treated as if it were exact, even if it might not be or if NCS-related density might be more similar in one region than in another. In the holographic method of Szöke *et al.* (1997), the possibility of imperfect NCS was introduced into the density-modification process by using a cost function to describe the expected similarity of NCS copies. In the method of Abrahams & Leslie (1996), imperfect NCS was also considered in weighting the various NCS copies during averaging.

Recently, we developed a method for improving crystallographic phases through the use of expectations about the electron density in a map that can take advantage of both the estimates of electron density in the map and uncertainties or probability distributions for those estimates (Terwilliger, 1999, 2000). This 'statistical density-modification' technique (previously known as 'maximum-likelihood density modification') combines experimental phase probabilities with phase probabilities derived from the expectations about the electron-density map to yield posterior (combined) phase probabilities. The key elements in this method are the use of a map-probability function to describe the plausibility of an electron-density map and the calculation of derivatives of the probability function to describe how this plausibility would change if an individual phase were changed.

The map-probability function in statistical density modification consists of the integral over all points in the map of a local log probability of the map. In turn, the local log probability of the map is the logarithm of the *a priori* probability of the value of the electron-density map at that point.

© 2002 International Union of Crystallography
Printed in Denmark – all rights reserved

electronic reprint

The *a priori* probability distributions for plausible values of the electron density at each point in the map can come from any source, including the flatness of the solvent, the expected range of electron densities in the region of a macromolecule and, in the present case, non-crystallographic symmetry. As the map-probability function uses probability distributions rather than expected electron densities, it takes into account both the expected electron density (the mean of the probability distribution) and uncertainties in this expectation (through the distribution itself). In the case of NCS, this means that statistical density modification can take NCS into account without requiring an implicit assumption that the NCS is exact or even that the deviations from perfect NCS are the same everywhere in the region where NCS applies.

2. Methods

2.1. Identification of NCS operators.

Non-crystallographic symmetry operators were identified using the NCS in heavy-atom sites (Terwilliger, 2002). Additionally, approximate centers of regions where they would apply were estimated from the centroids of the coordinates of each of the $n_{\rm NCS}$ sets of heavy-atom sites that could be related to another set of heavy-atom sites through NCS. These NCS operations were then checked by determining the covariance of density in regions related by the NCS operators as a function of distance d from the (approximate) centers of the regions where NCS applies, fitting this covariance to a Gaussian centered at $d = 0$, extrapolating the covariance to $d = 0$ and only including NCS if the extrapolated covariance $\langle\rho_i\rho_j\rangle$ was at least 0.1 times the mean-square value of the electron density in the map (it was typically 0.5 to 2 times the mean-square value of the map).

2.2. Identification of 'NCS asymmetric unit'

The region over which NCS applies and which is repeated $n_{\rm NCS}$ times in the asymmetric unit of the crystal was identified using the covariance in a fashion similar to that used for checking the NCS operators and similar to the automatic method described by Cowtan (1998). A local mean covariance of density among the $n_{\rm NCS}$ regions of NCS was used to identify this region. For each point on a grid centered at the center of one of the presumed regions where NCS applies, the mean value of the covariance of density $\langle\rho_i\rho_j\rangle$ for a sphere with radius of r around all pairs of points related by NCS to this one was calculated. The radius r was typically taken to be the same as the radius used for smoothing the squared electron density in mask calculation for solvent flattening (Wang, 1985). The NCS asymmetric unit was then defined by sequentially testing all points on the grid, starting with those close to the origin and then moving further away. If the point had a mean covariance of density greater than a cutoff $c_{\rm MIN}$ and was not related by crystallographic symmetry or NCS to any other point already in the NCS asymmetric unit, it was included. The cutoff $c_{\rm MIN}$ was chosen by testing a range of values and picking the one that yielded approximately the same fraction of the unit cell within the $n_{\rm NCS}$ regions of NCS as was expected to be within the macromolecule (*i.e.* not solvent) region of the unit cell.

2.3. Estimation of expected electron-density probability distribution from NCS

The electron density ρ_i at a point i related by NCS symmetry to $n_{\rm NCS} - 1$ other points in the asymmetric unit was estimated from the weighted mean of the density at the $n_{\rm NCS} - 1$ NCS-related points. The weights and the uncertainty in this estimate were estimated from the local covariance of density $\langle\rho_i\rho_j\rangle$ mentioned above as follows. A simple error model for the relationship between the density ρ_i and ρ_j at two NCS-related points i and j was used,

$$\rho_i = x + z_i, \quad (1)$$

$$\rho_j = x + z_j, \quad (2)$$

where x corresponds to the contribution to electron density that is shared by NCS-related points and z_i and z_j correspond the contributions that are unique to NCS copies i and j (including both errors in the map and true differences). The expected value of x^2 can then be estimated from the covariance of ρ_i and ρ_j,

$$\langle x^2\rangle = \langle\rho_i\rho_j\rangle, \quad (3)$$

where the average is taken over all points in the NCS asymmetric unit and all pairs of NCS copies i and j. The expected value of z_i^2 can then be estimated using (1), again averaged over the NCS asymmetric unit,

$$\sigma_i^2 = \langle z_i^2\rangle = \langle\rho_i^2\rangle - \langle x^2\rangle. \quad (4)$$

Using this error model, if ρ_i is used as an estimate of the electron density shared by all NCS copies for this point (x), then the expected variance in this estimate of x is given by σ_i^2.

This variance was used as the weighting term for averaging the $n_{\rm NCS} - 1$ values of electron density used to estimate the value of x for each point in the NCS asymmetric unit and for estimating the variance in this estimate. This estimate and variance x then formed the basis for a Gaussian probability distribution of the expected electron density at this point derived from NCS.

Several modifications to the simple model resulting in (4) were made to take into account local variations in similarity among NCS-related molecules and to reduce bias in the estimation of $\langle x^2\rangle$ arising from the iterative nature of density modification. The estimates of $\langle x^2\rangle$, the mean-square electron density shared by all NCS copies, are calculated in (3) as an average over the NCS asymmetric unit. In this formulation, all NCS copies and all points within the NCS asymmetric unit have the same value of $\langle x^2\rangle$. To take into account variation in the overall similarity between pairs of NCS-related molecules, an estimate $c_{ij} = \langle x^2\rangle_{ij}$ was estimated separately for each pair. Then, to take into account local variations in similarity among NCS-related molecules, a local estimate of $\langle x^2\rangle$ corresponding to the local mean-square density in common between all NCS-related copies was obtained by using a locally averaged value

electronic reprint

of $c(x) = \langle \rho_i \rho_j \rangle$ in (3), where the local average was taken over a sphere with the same radius r used above in the identification of the NCS asymmetric unit. Then, the overall estimate of $\langle x^2 \rangle$ for a particular pair of NCS-related points in molecules i and j was the product of c_{ij} and $c(x)$.

In this density-modification procedure, the electron density at points in $n_{\rm NCS} - 1$ NCS-related copies of the NCS asymmetric unit are use to estimate the expected density at points in the remaining copy. After one or more cycles of density modification, the density at all $n_{\rm NCS}$ copies of the NCS asymmetric unit tend to become increasingly similar. This is desirable of course, as the point of using NCS in density modification is to take advantage of the fact that the density in the various NCS copies really is more similar than is found in the initial map. However, in practice the density-modification procedure can sometimes make the NCS copies even more similar than they really are, leading to an overestimate of $\langle x^2 \rangle$ and an underestimate of the variance σ_i^2. To obtain a less biased estimate of the variance σ_i^2, a cross-validation approach was used. At the start of the density-modification procedure several cycles were carried out without including NCS information in the phasing process, but including the calculation of $\langle x^2 \rangle$. This overall covariance estimate was used as a 'free' or unbiased estimate throughout the density-modification procedure. Then, to account for local variation in the covariance of density $\langle x^2 \rangle$, the overall values were multiplied for each cycle of density modification by the current ratio of the locally averaged value of $\langle \rho_i \rho_j \rangle$ (as described above) to the overall mean value of $\langle \rho_i \rho_j \rangle$.

2.4. Combination of information from NCS with *a priori* probability distribution of electron density in the macromolecule region

Even in the absence of NCS, a substantial amount of information exists on the expected distribution of electron density at points in the region of the macromolecule. As discussed earlier (Terwilliger, 2000), the *a priori* probability distribution of electron density in this region can be expressed in terms of the distribution found for model electron density, ρ_M,

$$p(\rho_M) = \sum_k w_k \exp\left[-\frac{(\rho_M - d_k)^2}{2\sigma_k^2}\right], \qquad (5)$$

where the coefficients w_k, d_k and σ_k^2 are estimated by fitting $p(\rho_M)$ to the model electron density. A similar description can be obtained for the solvent-containing region of the model map. The *a priori* probability distribution for electron density in the macromolecule-containing region of an experimental map with errors can then be described by

$$p(\rho) = \sum_k w_k \exp\left[-\frac{(\rho - \beta d_k)^2}{2(\beta^2\sigma_k^2 + \sigma_{\rm MAP}^2)}\right], \qquad (6)$$

where the coefficients β and $\sigma_{\rm MAP}^2$ are estimated by fitting (6) and the corresponding equation for the solvent region (with the same values of β and $\sigma_{\rm MAP}^2$ and different values of w_k, c_k and σ_k^2) to the electron density in the experimental map. For simplicity we rewrite (6) as

$$p(\rho) = \sum_k a_k \exp[-b_k(\rho - c_k)^2], \qquad (7)$$

where

$$a_k = w_k, \qquad (8a)$$

$$b_k = \frac{1}{2(\beta^2\sigma_k^2 + \sigma_{\rm MAP}^2)}, \qquad (8b)$$

$$c_k = \beta d_k. \qquad (8c)$$

The NCS-based information consists of an estimate, ρ', of the density at each point in the map, with an associated variance σ^2,

$$p(\rho) \simeq \frac{1}{\sigma(2\pi)^{1/2}} \exp\left[-\frac{(\rho - \rho')^2}{2\sigma^2}\right] \qquad (9)$$

or

$$p(\rho) \simeq A \exp\left[-B(\rho - C)^2\right], \qquad (10)$$

where,

$$A = \frac{1}{\sigma(2\pi)^{1/2}}, \qquad (11a)$$

$$B = \frac{1}{2\sigma^2}, \qquad (11b)$$

$$C = \rho'. \qquad (11c)$$

Combining this with (7) yields the expression

$$p(\rho) = \sum_k a'_k \exp[-b'_k(\rho - c'_k)^2], \qquad (12)$$

where the coefficients a'_k, b'_k and c'_k are given by

$$a'_k = a_k A \exp\left[-\frac{b_k B}{(b_k + B)}(c_k - C)^2\right], \qquad (13a)$$

$$b'_k = b_k + B, \qquad (13b)$$

$$c'_k = \frac{b_k c_k + BC}{b_k + B}. \qquad (13c)$$

3. Results and discussion

3.1. Automatic identification and verification of NCS

The procedure for identifying (Terwilliger, 2002) and testing for NCS described here was tested by applying it to phases obtained from four MAD experiments and one SAD experiment on crystals with twofold, threefold, fourfold and sixfold NCS. In each case, the *SOLVE* software (Terwilliger & Berendzen, 1999) was used to identify selenium sites and calculate a starting electron-density map. The MAD data sets included a nucleotide diphosphate kinase with nine selenium sites from *Pyrobaculum aerophilum* (Pédelacq *et al.*, 2002), a hypothetical protein with 16 selenium sites from *P. aerophilum* (J. D. Pedelacq, E. Liong & T. C. Terwilliger, unpublished work), a red fluorescent protein with 26 selenium sites (Yarbrough *et al.*, 2001) and a formate dehydrogenase with 12

electronic reprint

Table 1
Automatic identification and evaluation of NCS from heavy-atom sites.

Structure	No. of sites found by *SOLVE*	NCS (actual)	NCS (found from heavy-atom sites)	NCS (found in electron-density map)
NDP kinase	9	Threefold	Threefold	Threefold
Hypothetical	16	Twofold	Twofold	Twofold
Red fluorescent protein	26	4 copies	4 copies	4 copies
2-Aminoethylphosphonate (AEP) transaminase	66	6 copies	6 copies	6 copies
Formate dehydrogenase	12	Twofold	Twofold†	Twofold
Gene 5 protein	2	None	None	None
Armadillo repeat from β-catenin	15	None	2 copies	None
Dehalogenase	13 (unique sites in 5 derivatives, plus 5 duplicate sites not included)	None	3 copies	None
Initiation factor 5A	4	None	None	None

† Only identified automatically when tolerance for NCS in heavy-atom sites was increased from 1.4 to 3 Å.

Table 2
Use of NCS in statistical density modification.

	Map correlation			$\langle\cos(\Delta\varphi)\rangle$		
		Density-modified			Density-modified	
Structure	*SOLVE*	No NCS	NCS	*SOLVE*	No NCS	NCS
NDP kinase	0.40	0.65	0.77	0.27	0.42	0.54
Hypothetical	0.42	0.58	0.62	0.50	0.61	0.68
Red fluorescent protein	0.76	0.90	0.91	0.70	0.79	0.80
Red fluorescent protein (using only 4 of 26 Se sites from *SOLVE*)	0.29	0.33	0.85	0.23	0.26	0.66
2-Aminoethylphosphonate (AEP) transaminase	0.64	0.79	0.81	0.52	0.63	0.66
Formate dehydrogenase	0.48	0.77	0.77	0.29	0.50	0.52

selenium sites from *P. aerophilum* (T. S. Peat, J. M. Newman, G. S. Waldo & T. C. Terwilliger, unpublished work; PDB entry 1qp8). The SAD data set was 2-aminoethylphosphonate transaminase with 66 selenium sites (Chen *et al.*, 2000).

Table 1 lists these crystals, with the number of NCS copies in the asymmetric unit, the number of NCS operators found from the selenium sites by the automatic procedure we developed recently (Terwilliger, 2002) and the number of NCS operators found after comparing the electron density at the potentially NCS-related positions. In all five cases, the NCS could be identified correctly from the heavy-atom sites and confirmed using the electron density in the map. In one of the cases (the dehydrogenase from *P. aerophilum)*, the NCS in the selenium sites could only be detected when the allowed deviation of the sites from perfect NCS was increased from the default value of 1.4 Å (half the resolution) to 3.0 Å.

As a further test, the procedure for identifying and verifying NCS was applied to three MAD data sets and one MIR data set in which the crystals had no NCS (Table 1). The MAD data sets were gene 5 protein (Skinner *et al.*, 1994), the armadillo repeat region from β-catenin (Huber *et al.*, 1997) and initiation factor 5A from *P. aerophilum* (Peat *et al.*, 1998), and the MIR data set was dehalogenase from *Rhodococcus* (Newman *et al.*, 1999). In each case, the two-step procedure of searching for NCS in the heavy-atom sites and verifying any NCS that was present in the heavy-atom sites resulted correctly in the conclusion that no NCS was present. In two cases (β-catenin and dehalogenase), the correct conclusion was drawn after NCS was found in the heavy-atom sites but not in the electron-density map.

3.2. Incorporation of NCS in statistical density modification

Table 2 illustrates the utility of NCS information in the context of statistical density modification, comparing the quality of the final electron-density maps with and without the inclusion of NCS information. The quality of the maps was assessed using two measures. One was the mean effective figure of merit of the phases, calculated from the mean cosine of the phase difference between the phases from this procedure and the phases from a refined model. The other was the correlation coefficient of the map calculated from this procedure with the map calculated using phases from the refined model.

The extent of phase improvement resulting from the use of the NCS information varied considerably from case to case (Table 2). The starting correlation of the map obtained from MAD or SAD phasing with the map calculated from the refined models and the starting mean cosine of the phase error is listed for each. Additionally, the corresponding values for the maps obtained after statistical density modification, with and without NCS, are shown. In both of the cases with twofold symmetry, the NCS information improved the phasing very slightly. This was a little surprising considering that the twofolds were not parallel to crystallographic symmetry axes in either case and the weights on NCS were sufficient to bring the correlation coefficients between NCS copies to 0.81 (formate dehydrogenase) and 0.93 (the hypothetical protein).

In the case of threefold symmetry, the inclusion of NCS made a very large difference, increasing the correlation of the resulting map with the map calculated from the refined model from 0.65 to 0.77. Somewhat surprisingly, in the cases with four and six NCS-related molecules the inclusion of NCS had a relatively small effect. On the other hand, the phases were very accurate even without NCS (correlations with the model map of 0.90 and 0.79, respectively), so in these cases there might simply not be much additional information available from the inclusion of NCS. This conclusion was tested by creating a test set of phases from the red fluorescent protein data set using just four of the 26 sites for phasing. As anticipated, beginning with a map correlation of 0.29, the inclusion of NCS had a very large effect, raising the final map correla-

electronic reprint

tion from 0.33 to 0.85, nearly as high as that obtained using all the selenium sites for phasing.

3.3. Estimation of variances in estimates of electron density from NCS

A key element of the statistical method for density modification is the ability to specify a probability distribution for the expected electron density in a map. In the case of NCS, this means that the method can, in principle, take into account the similarity of NCS-related molecules. It can also take into account the differences between NCS-related copies in a map that arise from errors in phases. In the present implementation of NCS in statistical density modification, (4) is used to estimate the uncertainty in target values of electron density used as prior information based on NCS. We tested whether these estimates are optimal by carrying out a systematic investigation of the phase improvement obtained when these variance estimates are multiplied by each of a range of values from 0.1 to 150.

Fig. 1 shows the result of this test applied to the NDP-kinase data, with the threefold NCS applied. When NCS variance estimates are multiplied by a large scale factor (150) so that NCS is effectively not used in density modification, the final correlation of the NCS-related regions of the map after density modification was just 0.66 and the correlation between the density-modified map and the map based on the refined structure of NDP-kinase was 0.66 (Fig. 1). At the other extreme, when variance estimates are multiplied by a small scale factor (0.1) so that NCS is strongly emphasized in density modification, the final correlation of the NCS-related regions of the map was 0.97, but a map correlation between the density-modified map and the map based on the refined structure of NDP-kinase is still only 0.68. At intermediate values of the scale factor (*e.g.* 0.8–1.2), corresponding closely to the estimates of the variance obtained with (4), NCS is included in density modification, but in a balanced way. The resulting final correlation of the NCS-related regions of the map is 0.94–0.95 and the final map correlation with the model map is 0.76–0.77. Overall, Fig. 1 indicates that the estimates of variances obtained using (4) are very close to optimal in carrying out statistical density modification with NCS.

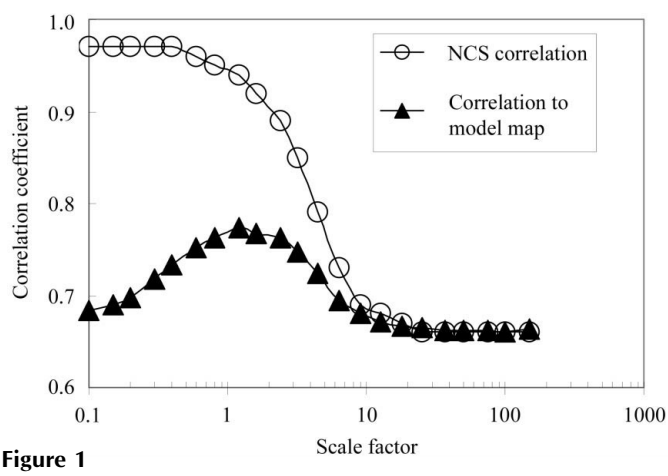


Figure 1
Effect of scale factor multiplying NCS variance estimates on correlation to model map and on NCS correlation.

4. Conclusions

The map-probability function in statistical density modification provides a way to incorporate information using a different probability distribution of expected values of electron density for every point in a map. This flexibility means that it is not necessary to assume that all NCS-related copies of a molecule in a crystal are identical, or even to assume that all parts of a single molecule are equally similar to the NCS-related parts of another molecule. The extent of local similarity among NCS-related molecules can be assessed using the local correlation of density and bias in these estimates owing to the iterative nature of density modification can be reduced by estimating the overall correlation by cross-validation.

The methods described here are implemented in the software *RESOLVE* (Terwilliger, 2000) available from http://solve.lanl.gov.

The author is grateful to the NIH and the *PHENIX* software-development consortium for generous support, to Li-Wei Hung for helpful discussions, to Thomas Peat for use of the formate dehydrogenase data prior to publication and to Osnat Herzberg for the use of the aminoethylphosphonate transaminase data prior to publication.

References

Abrahams, J. P. & Leslie, A. G. W. (1996). *Acta Cryst.* D**52**, 30–42.
Bricogne, G. (1974). *Acta Cryst.* A**30**, 395–405.
Chen, C. C. H., Kim, A., Zhang, H., Howard, A. J., Sheldrick, G. M., Dunaway-Mariano, D. & Herzberg, O. (2000). Am. Crystallogr. Assoc. Annu. Meet., Abstract 02.06.03.
Cowtan, K. (1998). *Acta Cryst.* D**54**, 750–756.
Huber, A. H., Nelson, W. J. & Weis, W. I (1997). *Cell*, **90**, 871–882.
Kleywegt, G. J. & Read, R. J. (1997). *Structure*, **5**, 1557–1569.
Newman, J., Peat, T. S., Richard, R., Kan, L., Swanson, P. E., Affholter, J. A., Holmes, I. H., Schindler, J. F., Unkefer, C. J. & Terwilliger, T. C. (1999). *Biochemistry*, **38**, 16105–16114.
Peat, T. S., Newman, J., Waldo, G. S., Berendzen, J. & Terwilliger, T. C. (1998). *Structure*, **6**, 1207–1214.
Pédelacq, J.-D., Piltch, E., Liong, E. E., Berendzen, J., Kim, C.-Y., Rho, B.-S., Park, M. S., Terwilliger, T. C. & Waldo, G. S. (2002). In the press.
Rossmann, M. G. (1972). *The Molecular Replacement Method.* New York: Gordon & Breach.
Skinner, M. M., Zhang, H., Leschnitzer, D. H., Guan, Y., Bellamy, H., Sweet, R. M., Gray, C. W., Konings, R. N. H., Wang, A. H.-J. & Terwilliger, T. C. (1994). *Proc. Natl Acad. Sci. USA*, **91**, 2071–2075.
Szöke, A., Szöke, H. & Somoza, J. R. (1997). *Acta Cryst.* A**53**, 291–313.
Terwilliger, T. C. (1999). *Acta Cryst.* D**55**, 1863–1871.
Terwilliger, T. C. (2000). *Acta Cryst.* D**56**, 965–972.
Terwilliger, T. C. (2002). *Acta Cryst.* D**58**, 2213–2215.
Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**, 849–861.
Wang, B.-C. (1985). *Methods Enzymol.* **115**, 90–112.
Yarbrough, D., Wachter, R. M., Kallio, K., Matz, M. V. & Remington, S. J. (2001). *Proc. Natl Acad. Sci. USA*, **98**, 462–467.

electronic reprint

electronic reprint

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Rapid automatic NCS identification using heavy-atom substructures

Thomas C. Terwilliger

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Rapid automatic NCS identification using heavy-atom substructures

Thomas C. Terwilliger

Mail Stop M888, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence e-mail: terwilliger@lanl.gov

Received 22 April 2002
Accepted 10 September 2002

An important component of a fully automated system for structure solution and phase improvement through density modification is a capability for identification of non-crystallographic symmetry as early in the process as possible. Algorithms exist for finding NCS in heavy-atom sites, but currently require of the order of N^5 comparisons to be made, where N is the number of sites to be examined, including crystallographically related locations. A method described here based on considering only sets of sites that have common interatomic distances reduces the computational time by several orders of magnitude. Additionally, searches for proper symmetry allow the identification of NCS in cases where only one heavy atom is present per NCS copy.

1. Introduction

Non-crystallographic symmetry (NCS) can be a powerful aid in improving the quality of macromolecular electron-density maps (Rossmann, 1972; Kleywegt & Read, 1998). There are many methods for finding NCS (*e.g.* Kleywegt & Read, 1998; Choi *et al.*, 1997; Colman *et al.*, 1976; Bailey *et al.*, 1988; Lu, 1999). One particularly useful method for identifying NCS early in the structure-solution process is to search for symmetries in the heavy-atom sites obtained by MAD, SAD or MIR (Buehner *et al.*, 1974). Recently, Lu (1999) described an automatic method for identifying symmetry in heavy-atom sites. The method consisted of trying all possible combinations of groups of three sites in an effort to find matching triangles and was demonstrated to be highly effective in finding NCS. The method was rather slow, however, requiring approximately N^5 comparisons to be examined, where N is the number of heavy-atom sites in the region considered for NCS, including all crystallographically related sites.

Here, we describe a related approach for the identification of NCS in heavy-atom sites that is very fast because the only comparisons that are considered are those where interatomic distances in one group at least partially match those in another. Consequently, only a fraction of the possible comparisons need to be made. Additionally, a method using searches for proper symmetry allows the identification of NCS in cases where as few as one heavy atom is present in each NCS copy.

1.1. Summary of the method

The basic idea of this method is similar to that of Lu (1999). Imagine a crystal with six heavy-atom sites. A particular subset of three of these heavy-atom sites (A–B–C) might be NCS-related to another set (D–E–F) if all the interatomic distances in the first set ($A-B$, $A-C$, $B-C$) match interatomic distances in the second set ($D-E$, $D-F$, $E-F$). The method of Lu (1999) is to expand the heavy-atom sites using space-group symmetry, then to take all sets of three sites, compare them with all sets of three other sites and find those sets that match in their interatomic distances. These pairs of sets could be related by NCS. If additional sites are present, then they are grouped into existing NCS sets or into new sets with the same interatomic distances if possible. The NCS operators for the crystal are then deduced based on the relationships of these sets. The method works well, but is very slow because of the very large number of comparisons that are required.

The computational requirements of this method can be greatly reduced by noting that a set (A–B–E) cannot possibly be related to a second set (C–D–F) if any one of the three intertomic distances does not match. This means that if $A-E$ does not match $C-F$, we do not have to even consider the distances $A-B$, $B-E$ *etc.* Furthermore, it means that the pairs A–E and C–F, which have different interatomic distances, never have to be considered as corresponding parts of triplets in combination with any other sites. This vastly reduces the number of comparisons that need to be made.

For example, suppose we have six heavy-atom positions in space group $P1$, with interatomic distances as in Table 1, and suppose further that we are expecting two sets of three heavy-atom positions related by NCS. Before examining the distances in Table 1, any

© 2002 International Union of Crystallography
Printed in Denmark – all rights reserved

electronic reprint

pair of atoms (*A*–*E*) might conceivably be NCS-related to any other pair (*e.g.* *C*–*F*). The distances in Table 1 limit these possibilities in a very systematic way. The pair *A*–*B*, for example, might be related to the pair *D*–*E* or *D*–*F* (because the interatomic distances are the same), but not to the pairs *C*–*D* or *C*–*E* or *C*–*F* (because the distances are very different).

It is possible to take advantage of the requirement for pairs of distances to match by sorting all pairs of sites according to their interatomic distances and only performing comparisons using pairs of sites that are close together in sequence in the list. In this way, for examples, the pairs *C*–*D*, *C*–*E* and *C*–*F* need never be compared with the pair *A*–*B* because they will be far apart in sequence in this list. This is the key element of the present method.

Continuing with the example in Table 1, the NCS can be identified in the following steps. Firstly, the pair *A*–*B* is considered as a possible pair of vertices in a triangle representing three sites in one NCS copy. All possible other pairs that could conceivably be the corresponding two vertices in an NCS-related triangle are then listed. These other pairs must share the same interatomic distance. In this case there are only two such possibilities (*D*–*E* and *D*–*F*), both of which have the same interatomic distance as *A*–*B* (20.6 Å). These matching pairs can be obtained without performing comparisons among all pairs because the three pairs *A*–*B*, *D*–*E* and *D*–*F* will all be very close together in the list of pairs sorted by interatomic distances.

At this point, a reasonable possibility (among many) for Table 1 is that *A*–*B* corresponds to *D*–*E*. All remaining sites could be considered as possible third vertices in the two triangles. Once again, however, the fact that the interatomic distances must match reduces the number of comparisons that have to be made. In this simple example, there are just two sites (*C* and *F*) that are not yet used, but in another case there might be hundreds. The approach in this case is to note that if site *C* becomes part of the first triangle with *A*–*B* and *F* becomes part of the triangle with *D*–*E*, then the distance $A-C$ must match the distance $D-F$. Accordingly, the pairs *A*–*C* and *D*–*F* must be close in sequence in the list of pairs sorted by interatomic distances and only such pairs of pairs need to be considered. In this example, *A*–*C* and *D*–*F* are both 20.6 Å and this combination is plausible. In the case of the twofold axis considered in Table 1, the other possibility (*A*–*B*–*F* and *D*–*E*–*C*) is also possible and in fact equally plausible.

Table 1
Mock interatomic distances (Å) for six sites in space group *P*1, where sites *A*, *B* and *C* are related by a twofold rotation to sites *C*, *D* and *E*.

Sites	*A*	*B*	*C*	*D*	*E*	*F*
A	0.0	20.6	20.6	60.0	63.4	63.4
B		0.0	10.0	63.4	60.8	60.0
C			0.0	63.4	60.0	60.8
D				0.0	20.6	20.6
E					0.0	10.0
F						0.0

2. Methods

The core of this method is the sorting of all pairs of sites according to their interatomic distances. The possible pairs of sites that need to be considered can then be limited to those with similar interatomic distances. In general, a set of *m* pairs of sites has the potential for representing *m* NCS copies only if all *m* pairs share (approximately) the same interatomic distance *d*.

The first step is to generate a list of all unique sites crystallographically equivalent to any one of the heavy-atom sites, but as close to the origin as possible. This list is then expanded using crystallographic symmetry to include all sites within a specified distance of the origin, which by default is chosen to be the smallest of the cell translations. This expansion must be over a large enough volume that all the NCS copies are represented at least once.

The second step is to sort all pairs of sites in this list according to their interatomic distances. This is the key step in this procedure; only pairs of sites near to each other in the sequence of this list can be corresponding pairs in different NCS copies.

The third step is to find two or more sets of three sites that have all interatomic distances in common. This step is greatly aided by the sorting of pairs of sites carried out in step 2, because a set of three sites from NCS copy *a* can only be related to three sites from NCS copy *b* if each set of two sites from copy *a* matches a set of two sites from copy *b*. Consequently, it is possible to build up a potential set of three sites in two NCS copies *a* and *b* as follows. Firstly, start with two pairs pair1*a* and pair1*b* of sites that have equal interatomic distances *d*1. Then consider all additional sets of two pairs of sites pair2*a* and pair2*b* with equal interatomic distances *d*2. Finally, consider only the intersection of these two groups where one atom in pair1*a* is the same as one atom in pair2*a* and one atom in pair1*b* is the same as one atom in pair2*b*. In this case, the three atoms in pair1*a* and pair2*a* share all interatomic distances with the three atoms in pair1*b* and pair2*b*. These groups are reasonable candidates for being NCS-related. Additionally, any additional sets of three atoms with the same set of interatomic distances are reasonable candidates for being part of a larger group of NCS-related molecules.

Fourthly, once a group of *m* sets of three atoms is found for which all sets have the same interatomic distances, a set of transformations relating the *m* NCS copies can be identified (provided the interatomic distances are not equal). Any additional atoms that are related to other atoms by these transformations can then be identified and grouped into the corresponding NCS copies.

The fifth step is to refine and score potential NCS solutions. A solution is refined by grouping all the heavy-atom sites into NCS copies (or not including them), then refining the NCS transformations to minimize the r.m.s. deviation among NCS-related sites. The scoring is performed in much the same way as described by Lu (1999). A set of NCS copies is most likely to be correct if (i) most or all heavy-atom sites are part of an NCS copy and (ii) NCS-related sites are very closely predicted by the NCS transformations. The NCS relationship is particularly likely to be correct if proper NCS is found and if two solutions are found, the one with the higher number of copies is generally more likely to be correct.

Based on these guidelines, two solutions *a* and *b* are compared. Let $N_{\rm NCS,a}$ and $N_{\rm NCS,b}$ be the numbers of sites that are part of an NCS copy for solutions *a* and *b* and let $N_{\rm SYM,a}$ and $N_{\rm SYM,b}$ be the number of NCS copies for solutions *a* and *b*. If solution *a* has the same or higher symmetry compared with solution *b* ($N_{\rm SYM,a} \geq N_{\rm SYM,b}$) and solution *a* has more sites as part of an NCS copy ($N_{\rm NCS,a} > N_{\rm NCS,b}$), solution *a* is always considered better. Also, if solution *a* has lower symmetry ($N_{\rm SYM,a} < N_{\rm SYM,b}$), but has many more sites as part of an NCS copy ($N_{\rm NCS,a}N_{\rm SYM,a} > N_{\rm NCS,b}N_{\rm SYM,b}$), then solution *a* is always considered better.

If all these are equal, then three more quantities are calculated for each solution to help identify which solution is more likely. The first quantity is the r.m.s.d. of the NCS-related sites from positions predicted by NCS (r.m.s.d.$_{\rm NCS,a}$ and r.m.s.d.$_{\rm NCS,b}$, for solutions *a* and *b*, respectively). The second is a variable which is 1 if the NCS has point-group symmetry and 0 if not (pg$_{\rm NCS,a}$ and pg$_{\rm NCS,b}$, for solutions *a* and *b*, respectively). The third is the r.m.s. distance among all the sites in each NCS group (r.m.s.$_{\rm NCS,a}$ and r.m.s.$_{\rm NCS,b}$, for solutions *a* and *b*, respectively). Whichever of the two solutions has

electronic reprint

the lower r.m.s.d. of NCS-related sites from positions predicted by NCS (r.m.s.d.$_{NCS,a}$ or r.m.s.d.$_{NCS,b}$) is considered better. If these are equal, then whichever solution has the greater point-group symmetry is considered better. If these are also equal, then whichever solution has the lower r.m.s. distance among all the sites in each NCS group (r.m.s.$_{NCS,a}$ or r.m.s.$_{NCS,b}$) is considered better. If all these are also equal, the solutions are considered to be of equal quality.

For cases where fewer than three heavy-atom sites exist per NCS copy, but where proper NCS exists, an alternative approach can be taken. Two general cases can be imagined where NCS can still be deduced. Firstly, NCS can be deduced if there is a twofold axis of symmetry and two sites exist per NCS copy and secondly, NCS can be deduced if there is a threefold or higher axis of symmetry and one or more sites exist per NCS copy.

For the case with a twofold axis and two sites per NCS copy, the sorting of pairs of sites by interatomic distances can again be used to identify potential sets of two pairs of sites that could be related by a twofold axis. Each of these sets of pairs of sets is tested to see whether the four atoms could be related by a twofold axis. This is straightforward because the twofold must pass through two points defined by the mid-points between each potentially twofold-related atom.

For the case with an *N*-fold axis and one site per NCS copy, the sorting of pairs of sites is once again useful because the *N*-fold axis must be made up of a set of *N* atoms, all of which have the same interatomic distance to two other atoms. Consequently, only a very few sets of sites need to be considered at all as potentially *N*-fold related.

In each of these methods, some criterion must be applied to define whether two distances are approximately equal or whether two sites are approximately the same. In practice, a cutoff of about half the resolution of the data is suitable for each of these criteria.

3. Results

These approaches for finding NCS in heavy-atom sites were tested using the locations of Se atoms in four data sets containing between nine and 66 sites and containing either proper twofold or threefold symmetry or improper NCS containing up to six copies (Table 2). The cases tested were a nucleotide diphosphate kinase with nine selenium sites from *Pyrobaculum aerophilum* (Pédelacq *et al.*, 2002), a hypothetical protein with 16 selenium sites from *P. aerophilum* (J. D. Pedelacq, E. Liong & T. C. Terwilliger, unpublished work), a red fluorescent protein with 26 selenium sites (Yarbrough *et al.*, 2001) and 2-aminoethylphosphonate transaminase with 66 selenium sites (Chen *et al.*, 2000). In each case, the sites were those found by running the software *SOLVE* (Terwilliger & Berendzen, 1999).

Table 2
NCS relationships found.

Structure	Space group	Sites	NCS	Time (*RESOLVE*) (s)	Time (*FINDNCS*) (s)
NDP kinase	*C*2	9	Threefold	5	624
Hypothetical	*C*2	16	Twofold	8	3700
Red fluorescent protein	$P2_1$	26	4 copies (no point-group symmetry)	12	>9000
2-Aminoethylphosphonate (AEP) transaminase	$P2_1$	66	6 copies (no point-group symmetry)	78	>13000

In each case the algorithms described above found the known NCS. The CPU time required for finding, sorting, scoring and coming up with a single solution for each case ranged from 1 to 78 s. This compares with 600 to over 10 000 s using the brute-force methods described by Lu (1999) and implemented in the program *FINDNCS*, using defaults for all parameters or half the cell dimensions as limits for the search region, whichever was successful in the shorter time. In the cases of the 26 sites in red fluorescent protein and the 66 sites of AEP, the *FINDNCS* program was unable to complete the search as a matrix used in calculation was singular.

The approach described here can find NCS relationships in many cases, but does have limitations. For example, some distance cutoff must be used in considering whether two pairs of atoms are likely to be NCS-related, or an infinite number of possibilities would have to be considered. In practice, a cutoff of the smallest of the cell translations works well for this, but in some cases NCS could still be missed. At the other extreme, a cutoff for how similar two distances must be for them to be considered to be NCS-related is also necessary. The cutoff of half the resolution works well in many cases, but might not in cases where heavy-atom sites are not in quite the same places in different molecules. Also, in some cases the scoring system used to choose the NCS may not be optimally weighted. The user has the option to specify the number of NCS copies, however, and this can be used to limit the search to that number.

4. Conclusions

The methods described here for rapid identification of NCS in heavy-atom substructures are well suited to being a part of automated structure-solution procedures because they are robust and very quick. They have already proven very useful in automatic NCS symmetry averaging in *RESOLVE* (Terwilliger, 2000).

The author is grateful to the NIH and the *PHENIX* software-development project for generous support. The methods described here are implemented in the software *RESOLVE* (Terwilliger, 2000) available from http://solve.lanl.gov. Stand-alone software that carries out just these methods '*HA_NCS*' is also freely available from http://solve.lanl.gov.

References

Bailey, S., Dodson, E. & Phillips, S. (1988). Editors. *Proceedings of the CCP4 Study Weekend. Improving Protein Phases.* Warrington: SERC Daresbury Laboratory.
Buehner, M., Ford, G. C., Moras, D., Olsen, K. W. & Rossmann, M. G. (1974). *J. Mol. Biol.* **82**, 563–585.
Chen, C. C. H., Kim, A., Zhang, H., Howard, A. J., Sheldrick, G. M., Dunaway-Mariano, D. & Herzberg, O. (2000). Am. Crystallogr. Assoc. Annu. Meet., Abstract 02.06.03.
Choi, H., Lu, G., Wengler, G. & Rossmann, M. G. (1997). *Proteins Struct. Funct. Genet.* **27**, 345–359.
Colman, P. M., Fehlhammer, H. & Barteis, K. (1976). *Crystallographic Computing Techniques*, edited by F. R. Ahmed, K. Huml & B. Sedlacek, pp. 248–258. Copenhagen: Munksgaard.
Kleywegt, G. & Read, R. (1998). *Structure*, **15**, 1557–1569.
Lu, G. (1999). *J. Appl. Cryst.* **32**, 365–368.
Pédelacq, J.-D., Piltch, E., Liong, E. E., Berendzen, J., Kim, C.-Y., Rho, B.-S., Park, M. S., Terwilliger, T. C. & Waldo, G. S. (2002). In the press.
Rossmann, M. G. (1972). *The Molecular Replacement Method.* New York: Gordon & Breach.
Terwilliger, T. C. (2000). *Acta Cryst.* D**56**, 965–972.
Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**, 849–861.
Yarbrough, D., Wachter, R. M., Kallio, K., Matz, M. V. & Remington, S. J. R. (2001). *Proc. Natl Acad. Sci. USA*, **98**, 462–467.

electronic reprint

electronic reprint

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Automated main-chain model building by template matching and iterative fragment extension

Thomas C. Terwilliger

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Automated main-chain model building by template matching and iterative fragment extension

Thomas C. Terwilliger

Mail Stop M888, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence e-mail: terwilliger@lanl.gov

Received 1 July 2002
Accepted 1 October 2002

An algorithm for the automated macromolecular model building of polypeptide backbones is described. The procedure is hierarchical. In the initial stages, many overlapping polypeptide fragments are built. In subsequent stages, the fragments are extended and then connected. Identification of the locations of helical and β-strand regions is carried out by FFT-based template matching. Fragment libraries of helices and β-strands from refined protein structures are then positioned at the potential locations of helices and strands and the longest segments that fit the electron-density map are chosen. The helices and strands are then extended using fragment libraries consisting of sequences three amino acids long derived from refined protein structures. The resulting segments of polypeptide chain are then connected by choosing those which overlap at two or more C^{α} positions. The fully automated procedure has been implemented in *RESOLVE* and is capable of model building at resolutions as low as 3.5 Å. The algorithm is useful for building a preliminary main-chain model that can serve as a basis for refinement and side-chain addition.

1. Introduction

Model building is a key and often time-consuming step in macromolecular structure determination. This step is important because model building is the initial interpretation of the experimental electron-density map in terms of the locations of atoms in the structure. If the resolution of the X-ray data is high (<2 Å), then atomic refinement of the model is highly effective and errors in the initial interpretation can often be corrected. If the resolution of the X-ray data is low (~3 Å), however, atomic refinement is less effective and it may be very difficult to correct any errors in this initial interpretation (Kleywegt & Jones, 1997). Although manual model building using a very good electron-density map can require less than a day for 100 or more residues, when the electron-density map is less clear the process can be much slower.

It has been recognized for some time that automated procedures for model building would speed up the macromolecular structure determination process considerably and several procedures for doing this have been developed. Most of these procedures are based on the connectivity of the polypeptide chain or on the presence of regular structure (helices and β-strands, common motifs) in the chain. Greer (1985) devised a rapid procedure ('bones') for tracing the path of the polypeptide chain using the connectivity of regions of high electron density in the map. This procedure was extended by Swanson (1994) to allow threshold-independent tracing of

© 2003 International Union of Crystallography
Printed in Denmark – all rights reserved

electronic reprint

connected regions in a map. Feigenbaum *et al.* (1977) used artificial intelligence methods to identify features in electron-density maps. Jones & Thirup (1986) and Jones *et al.* (1991) fitted electron density with fragments from known protein structures. Oldfield (2002) described a method for automated model building that began by identifying helices and strands and then extending these segments one amino acid at a time to trace a polypeptide chain. Cowtan (1998, 2001) and later Terwilliger (2001) used FFT-based approaches to identify the locations of helices, β-strands and other structure in an electron-density map by template matching. Holton *et al.* (2000) used machine-learning techniques to identify side chains in a map. McRee (1999) has described a semi-automated method for building main-chain and side-chain models in a map, beginning with the identification of C^{α} positions and fitting fragments from a main-chain library and continuing with using a rotamer library to fit side chains. Pavelcik *et al.* (2002) described an alternative and very rapid method for template matching of arbitrary fragments of structure to a map. Levitt (2001) uses a stepwise approach to model building, beginning with the 'bones' of Greer (1985) to identify helices and strands and extending them one amino acid at a time using ψ, φ angles from tables of allowed values. The most widely used automated model-building procedure in current use, *ARP*/*wARP*, has been described by Lamzin & Wilson (1993), Perrakis *et al.* (1999) and Morris *et al.* (2002). This procedure is very different from all those described above because it is based on an interpretation of the electron-density map in terms of individual atoms, iteratively followed by atomic refinement and an interpretation of the atomic coordinates in terms of a polypeptide chain. The requirement for atomicity limits the application of the method to electron-density maps at a resolution of about 2.3 Å, but for data at this resolution or better the method is exceptionally powerful for automatic model building and atomic refinement.

Here, we describe a procedure for automated model building that is related to those described by Oldfield (2002), McRee (1999) and Levitt (2001), but which uses alternative approaches to carry out each of the constituent steps. The method of Cowtan (1998) is used as a sensitive method for identifying the locations of helices and β-strands. Correlations of template density and map density rather than density at atomic coordinates are used for refinement of the position and orientation of fragments. A fragment-placement method based on tripeptides from refined protein structure and related to the method of Jones & Thirup (1986) is used to extend segments of structure. Chain connectivity and the correct chain direction are determined by requiring that independently built segments must overlap at two or more consecutive C^{α} positions before they are merged into a single segment.

2. Methods

As in previous methods for main-chain model building at moderate resolution (Oldfield, 2002; Levitt, 2001), our procedure is carried out in hierarchical steps. Firstly, helices and β-strands are located and fitted, with multiple interpretations of each of these secondary structures typically kept. Each helix or strand is then extended in an iterative fashion with libraries of tripeptides from refined protein structures. The collection of (overlapping) partially extended fragments are then assembled into a polypeptide chain by requiring that two or more consecutive C^{α} positions overlap for two segments to be merged, by requiring that there be no atomic overlaps and by beginning with the best-fitting segments. Each of these steps and the generation of templates and fragment libraries is described below. In all steps, space-group symmetry is used to identify positions that are equivalent in the unit cell and the distance between two points is considered to be the smallest distance between one of the points and any point symmetry-related to the other.

2.1. Helical and β-strand templates

An averaged helical template similar to that described in Terwilliger (2001) was used to identify helical segments in a map. This template consists of the average electron density calculated from a collection of α-helical segments six amino acids in length (from phycoerythrin; PDB code 1lia; Chang *et al.*, 1996; Berman *et al.*, 2000), all superimposed on a standard α-helical segment (from myoglobin; PDB code 1a6m; Vojtechovsky *et al.*, 1999; Berman *et al.*, 2000). The template included all points within 4 Å of a main-chain or C^{β} atom in the standard segment. The template was calculated at a resolution of 3 Å. An averaged β-strand template was constructed in the same fashion, except that the segments used in the template were four amino acids long.

2.2. Fragment libraries

Four fragment libraries were constructed. One consisted of 17 α-helical segments from six to 24 residues long in the protein phycoerythrin. Each segment of more than six residues was superimposed on the standard helical segment in three positions: one with the N-terminal six residues of the segment superimposed on the standard segment, one with the C-terminal six residues superimposed and one with the middle six residues superimposed. In this way, a short helical segment that is identified can potentially be extended in either direction. A second library consisted of 17 β-strand segments from four to nine amino acids long from chain *A* of carboxypeptidase A (PDB code 1bav; Massova *et al.*, 1996; Berman *et al.*, 2000), superimposed on the standard β-strand fragment in the same way as for the helical segments.

The third and fourth libraries consisted of segments of protein structure three amino acids in length chosen to represent all three-amino-acid segments in a set of refined protein structures [chosen arbitrarily from non-redundant PDB files (Hobohm *et al.*, 1993) with *R* factors of 20% or lower and resolution 1.8 Å or better]. The two libraries differed in that one contained all main-chain and C^{β} atoms of a tripeptide and the other contained the C^{α}, C and O of one residue plus the following two full residues. The first library was designed for extending a polypeptide chain in the

electronic reprint

N-terminal direction by superimposing the last C^{α}, C and O atoms of the template with the corresponding atoms of the N-terminal residue in a chain. The second library was designed for extending in the C-terminal direction, superimposing the same three atoms. The two libraries were subsets of the set of all tripeptides (or tripeptides minus the N) in a set of refined protein structures. In each case, the library was constructed by picking members that differed from each other by at least 0.5 Å r.m.s. and such that all tripeptides matched a member of the library with an r.m.s. deviation of less than 0.5 Å. The N-terminal library was based on 298 proteins and contained 9232 members, and the C-terminal library was based on 567 protein structures and contained 4869 members.

2.3. Convolution-based identification of the locations of helical and β-strand segments

The approximate locations and orientations of helices and β-strands were identified using the helical and β-strand templates mentioned above and an FFT-based convolution method for identifying locations of molecular fragments in a map (Cowtan, 1998, 2001) as implemented in Terwilliger (2001). The rotation-angle step size (θ) was typically 30°. In order to minimize the number of orientations that needed to be tested, the helical and β-strand templates described above were oriented so that the axis of the helix and the strand direction were both along the x axis. In this way, the step size of the sampling of possible rotations around the x axis could be maximized and the number of rotations minimized. The rotation step size about x is 30° for helical templates and 40° for strands. The number of rotations was reduced for the helical template by only considering 100° of rotation about the helical axis, as any further rotation yields a near-duplicate that differs by translation. The number of rotations was further decreased by skipping all rotations that through space-group symmetry resulted in a convolution that duplicated any other rotation. With these reductions, a typical convolution search in space group $C2$ at a resolution of 2.6 Å requires about 100 rotations for the helical template and 950 rotations for the β-strand template.

2.4. Correlation-based refinement of orientation and location of helices and strands

The convolution search for helical and β-strand segments results in a list of locations and orientations sorted by the overlap integral of the template with the map at those locations and with those orientations. The locations and orientations were refined by maximizing the correlation of the template with the map. After refinement, the lists of helices and strands were shortened by removing all those with low correlation coefficients, typically cutting off at a correlation of $\frac{1}{2}\langle m\rangle$, where $\langle m\rangle$ is the mean figure of merit of the data used to calculate the map.

2.5. Helix- and strand-fragment placement

The refined position and orientation of each β-strand and helical fragment is then used as a potential location of a strand or helix. Each member of the β-strand or helical fragment libraries is then placed in one such position and orientation and tested for a match to the electron density nearby. For each position/orientation of the standard helical fragment, for example, all 43 members of the helical fragment library were superimposed on the standard fragment, each in three different positions as described above. Then, for each placement of a helical fragment, a segment from the fragment is chosen that fits the electron density in the region. The segment included is the longest contiguous segment of the helix in which the mean density for all atoms is above a threshold ($r_{\rm overall}\rho_{\rm c}$, typically $r_{\rm overall} = \frac{3}{4}$, where ρ_c is the mean density at atoms near the center of the fragment) and the atoms on the ends were in density above a second threshold ($r_{\rm end}\rho_{\rm c}$, typically $r_{\rm end} = \frac{1}{2}$). An identical procedure is used for β-strand segments. Each placement of a segment of helix or strand is then scored with a score Q based on the mean electron density at coordinates of atoms in the segment and the number of atoms in the segment: $Q = \langle\rho\rangle N^{1/2}$. For each position/orientation, the top-scoring segment is saved. Once all helix and strand placements have been analyzed in this fashion, the mean and standard deviation of scores for helices and for strands are calculated, and a Z score is obtained for each placement, $Z = (Q - \langle Q\rangle)/\sigma(Q)$. At this point, all placements where the top-scoring segment has a Z score below a threshold (typically 0.5) are discarded.

2.6. Segment extension

Construction of a segment of a polypeptide chain is accomplished by iterative fragment extension. The goal in extending a segment by one or a few residues is to find a configuration of the main chain that is physically reasonable, that matches the electron-density map and that can be further extended into additional density. A look-ahead procedure was used to extend segments in either the N-terminal or C-terminal directions. The essence of the procedure is to extend with a tripeptide that matches the density and which can itself be extended with a second tripeptide that also matches the density. To accomplish this, each tripeptide from the C-terminal library is tested as a possible extension by superimposing the first residue of the tripeptide on the last residue in the current segment and evaluating the mean density in the map at the coordinates of atoms in the next two residues of the fragment. The top-scoring 'first-level' fragment or fragments are then tested for steric overlaps (distance of any atom in the fragment of <3.5 Å from any C^{α} atoms at least two residues away in the segment already built) and any physically implausible fragments are rejected. Then the look-ahead step is carried out. Each of these first-level top-scoring fragments is then used as a starting point for a second extension and the second-level top-scoring addition to each is noted. The overall score for each of the first-level fragments is the mean electron density at the coordinates of atoms in the fragment plus its extension (*i.e.* at the positions of atoms in four amino acids). The top-scoring first-level fragment (two amino acids) is then used to extend the segment.

electronic reprint

In this extension process, all main-chain atoms in the fragment are required to be above a threshold ($r_{\min}\rho_{\rm r.m.s.}$, typically $r_{\min} = 1$, where $\rho_{\rm r.m.s.}$ is the r.m.s. of the map in the region of the macromolecule) or they are rejected. Additionally, each fragment to be considered as an extension is tested to verify that the density is relatively uniform in the fragment. The procedure described above for truncation of helical segments to the region of helical density is followed (identifying the longest contiguous segment of the helix in which the mean density for all atoms is above a threshold and where the atoms at the ends were in density above a second threshold) and any fragments for which either end is removed by this procedure are rejected.

The procedure for extension described above will stop if no fragment can be found to extend the segment. Several backup procedures are used in this case. Firstly, the procedure is repeated testing a larger number of first-level fragments (one was tested on the first try, ten were tested on the second try and 40 on the third). If this also fails, then the procedure is repeated starting one amino acid back in the segment (fragments are added two amino acids at a time, so backing up one is a new starting point). If this fails, no further additions are made to this end of the segment.

When a segment can no longer be extended in either direction, it is scored, with the score equal to the mean density at coordinates of atoms in the segment times the square root of the number of atoms in the segment.

2.7. Chain assembly

The procedures described above generate a set of segments that may correspond to portions of polypeptide chain. As they begin from helices or strands that may have been overlapping, some pairs of segments may be almost identical. Also, as they may have had extensions on either end, the segments may overlap through their extensions. The goal of the chain-assembly step is to identify sets of segments that are likely to correspond to a continuous polypeptide chain. The step is carried out iteratively. In each cycle, the top-scoring segment identified above that is not already used and that does not overlap with a previously built chain is taken as a starting point for building a continuous chain. All segments are then considered, in order of their scores, as a possible extension to this new chain. If a segment matches the current new chain at two C^{α} atoms or more including one or both ends (matching typically defined as within $r_{\rm match} < 1.6$ Å), extends it in either direction and the extension does not result in any implausibly close atoms (distance < 3.5 Å), then the chain is extended using the residues in the segment. This becomes the new current chain and the process is repeated until no further additions can be made to the chain. A new chain is then begun as above and the overall chain-assembly process is repeated until no new chains can be created. This results in a set of continuous polypeptide chains, none of which overlap with any other.

3. Results and discussion

3.1. Optimizing values of parameters

The automated model-building procedure described here has been incorporated into the *RESOLVE* software (Terwilliger, 2000). The model-building procedure described here depends on a number of parameters mentioned above. To test the sensitivity of the model-building procedure to the values of key parameters, the density-modified electron-density map for NDP kinase (Pédelacq *et al.*, 2002) was used as a starting point, parameters were systematically varied and their effects on the number of residues built and the r.m.s. difference from the refined structure were examined. The NDP kinase map was chosen because it was at moderate resolution (2.6 Å), a moderate fraction of residues could be successfully built (78%) and the map was of moderate quality after density modification ($\langle m \rangle = 0.56$).

The parameters in the model-building procedure that seem most likely to affect the overall results of the procedure include θ, the rotation-angle step for the convolution-based search for helices and strands, $r_{\min}$, the minimum normalized electron density allowed at atomic positions, and $r_{\rm match}$, the maximum distance two C^{α} atoms can be from each other to be considered a match for fragment assembly. Each of these was tested for its effects on the NDP kinase model building. In

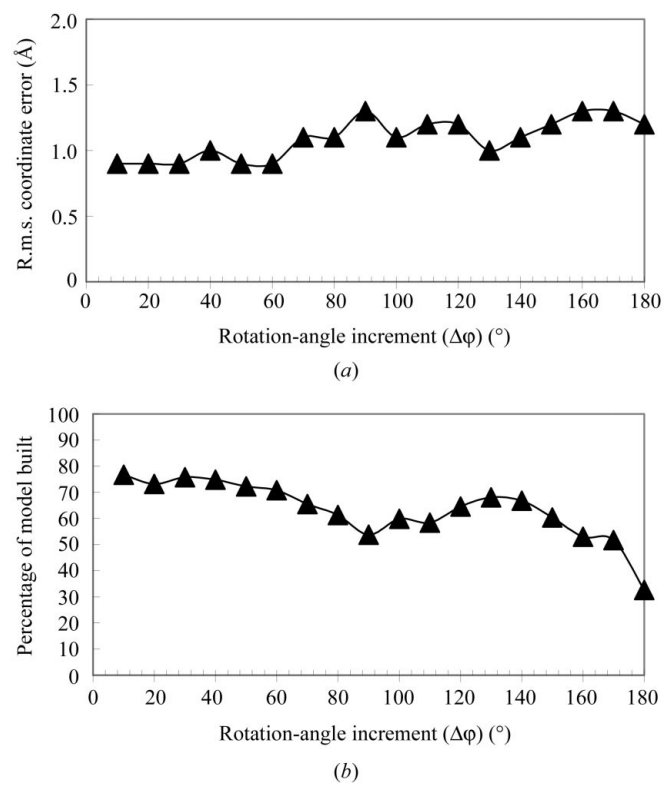


Figure 1
Effect of sampling interval on the FFT-based fragment search. Models were built for NDP kinase (Pédelacq *et al.*, 2002) as described in the text, varying only the values of the angular increment between FFT-based fragement searches. The percentage of the model built and the r.m.s. coordinate difference between the resulting (main chain and C^{β}) model and the refined model of NDP kinase are shown.

electronic reprint

these tests, the values of all the parameters except the one to be varied were fixed at the values of $\theta = 30^\circ$, $r_{\rm min} = 1$ and $r_{\rm match} = 1$ Å. (The value of $r_{\rm match}$ used in this test is not the optimal value of 1.6 Å; however, as noted below, the results are relatively insensitive to this parameter and this test was carried out before the optimum was known.) The quality of each of the models was assessed by comparing it to the refined model of NDP kinase. As the sequence is not assigned in the main-chain models, we assessed this quality as the r.m.s. coordinate difference between each main-chain atom in the models and the nearest atom with the same name in the refined structures, excluding any atoms more than 10 Å from any atoms in the refined structures.

Figs. 1, 2 and 3 shows the results of these tests. For each value of each parameter, the number of residues built and the r.m.s. deviation of the model coordinates from the refined coordinates of NDP kinase were determined. Fig. 1 illustrates the effect of varying the sampling interval in the FFT-based fragment search. As expected, the coordinate error is lowest (0.9 Å) and the completeness of the model is highest (77%) when the fragment search is carried out on a fine grid (10–30° intervals, with a total of 5000 rotations considered for the 10° interval and 138 rotations considered for the 30° interval). Somewhat surprisingly, however, even at the most coarse grid considered (nominally 180°, but actually six rotations considered for the β-strand template) fragments could still be identified and much of the model could still be built. The r.m.s. difference between the coordinates of atoms in the model built automatically and those of the refined model increased slightly (from 0.9 to 1.3 Å) as the grid was made more coarse. Based on this experiment, it appears that a grid search with a nominal interval of about 30° is optimal for this model-building procedure.

Fig. 2 shows the effect of varying the minimum density allowed at the coordinates of main-chain atoms added during fragment extension. For values of $r_{\rm min}$ (the minimum allowed density, normalized to $\rho_{\rm r.m.s.}$, the r.m.s. of the map in the region occupied by the macromolecule) of about 0.5 or less, the coordinate error is quite high (1.5–2 Å), while for values of about 1 or greater, the coordinate error is about 0.9 Å. The fraction of the model built decreases somewhat as this parameter is increased. It does not drop to zero because much of the model is built of fragments obtained in the FFT-based search and that part of model-building is not affected by this parameter.

Fig. 3 illustrates the effect of changing the value of $r_{\rm match}$, the maximum distance between matching C^α atoms to be

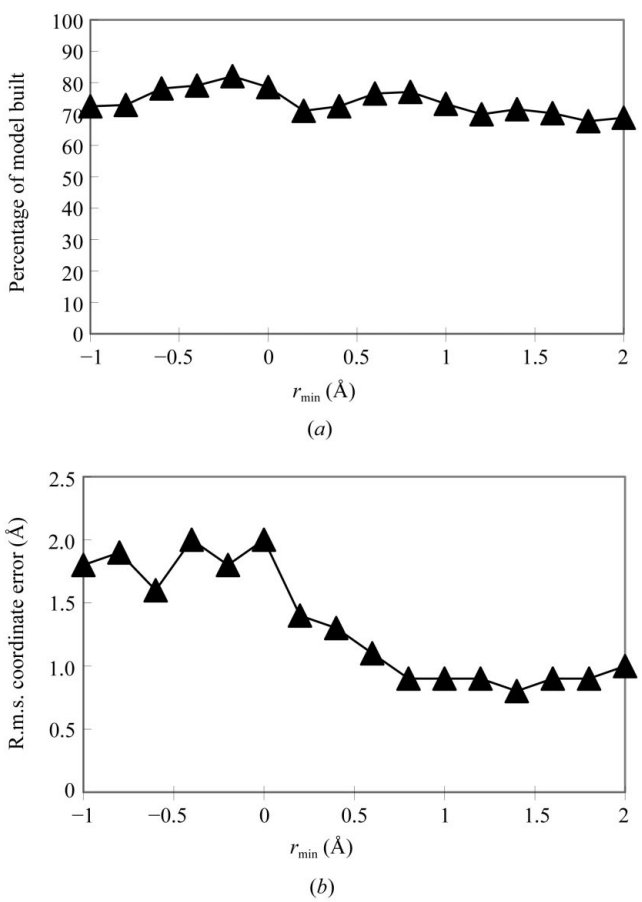


Figure 2
Effect of minimum-density cutoff at atomic positions on model building. Models were built and tested as in Fig. 1, varying only $r_{\rm min}$, the minimum allowed density, normalized to $\rho_{\rm r.m.s.}$, the r.m.s. of the map in the region occupied by the macromolecule.

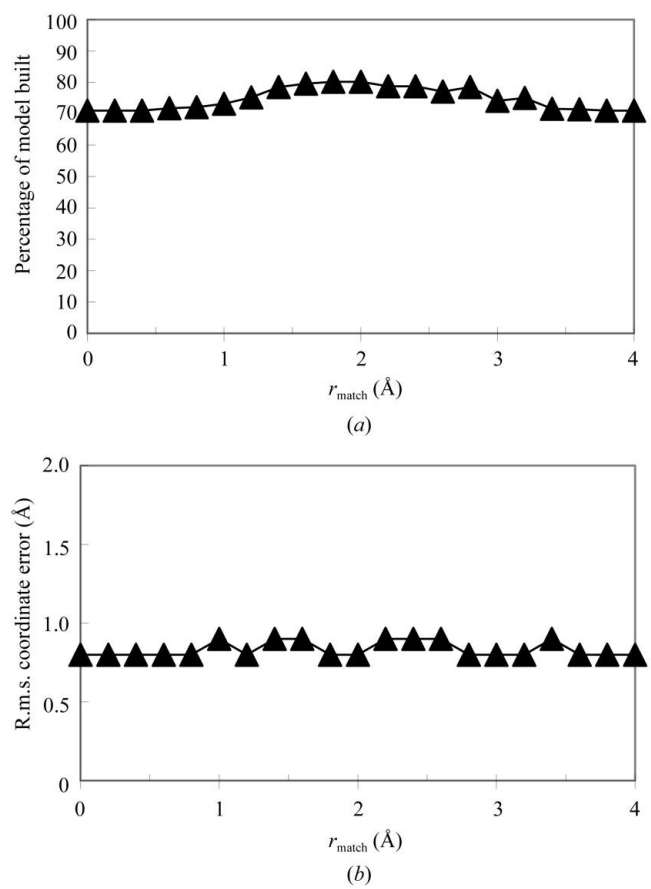


Figure 3
Effect of maximum-distance cutoff for matching atoms in chain assembly. Models were built and tested as in Fig. 1, varying only $r_{\rm match}$, the maximum distance between matching C^α atoms to be considered a match for fragment assembly.

electronic reprint

Table 1
Test structures for which models have been built with *RESOLVE*.

Structure	Resolution (Å)	Figure of merit (after ML density modification) $\langle m \rangle$	No. of residues in refined model	No. of residues built	Percentage of residues built (%)	Main-chain r.m.s. coordinate difference from refined structure (Å)
Gene 5 protein (Skinner *et al.*, 1994)	2.6	0.62	87	57	66	0.9
Granulocyte-stimulating factor (Rozwarski *et al.*, 1996)	3.5	0.70	242	121	50	1.6
Initiation factor 5A (Peat *et al.*, 1998)	2.1	0.85	136	121	89	0.6
β-Catenin (Huber *et al.*, 1997)	2.7	0.72	455	407	89	1.1
NDP kinase (Pédelacq *et al.*, 2002)	2.6	0.56	556 (3 × 186)	397	71	0.9
Hypothetical (*P. aerophilum* ORF, NCBI accession No. AAL64711; Fitz-Gibbon *et al.*, 2002)	2.6	0.58	494 (2 × 247)	451	91	0.6
Red fluorescent protein (Yarbrough *et al.*, 2001)	2.5	0.91	936 (4 × 234)	854	91	0.5
2-Aminoethylphosphonate (AEP) transaminase (Chen *et al.*, 2000)	2.6	0.84	2232 (6 × 372)	2037	91	0.7

considered a match for fragment assembly. There is only a slight dependence of the model building on this parameter, but for intermediate values (1.5–2.5 Å) somewhat more residues could be built than for lower or higher values. The r.m.s. error in the coordinates also increases slightly for these intermediate values, however. When the value of this parameter is very low, no chain assembly is performed. The number of residues built does not go to zero, however, because the chain can still be built up by extension of the templates from their ends.

3.2. Tests with structures solved by MAD and SAD

The procedure for automated main-chain model-building described here was further tested by applying it to a set of eight experimental maps with varying resolution, quality (figure of merit) and number of residues in the asymmetric unit. Two of these maps (NDP kinase and gene 5 protein) were used in the development of the algorithm, so that parameters could potentially be specifically optimized for them. The other six were not used to optimize parameters and therefore can give a somewhat more independent evaluation of the procedure. In each case, experimental MAD or SAD phases were first improved with statistical density modification (Terwilliger, 2000) including non-crystallographic symmetry information in the analysis. The resulting maps were used for model building. The values of the parameters tested in Figs. 1, 2 and 3 were fixed at values of $\theta = 30^\circ$, $r_{\rm min} = 1$ and $r_{\rm match} = 1.6$ Å. In the cases tested in Table 1, from 51 to 93% of the main chain could be built. Even the relatively poor map at 3.5 Å of granulocyte-stimulating factor could be partially interpreted, although the chain direction was incorrect in several instances for this model.

This r.m.s. coordinate difference between the models built with the present method and refined models ranged from 0.6 Å (for maps at resolutions of 2.1 and 2.6 Å) to 1.6 Å (for the map at a resolution of 3.5 Å). Considering that the models have been built from fragment libraries designed to match fragments from known proteins within about 0.5 Å and no refinement has been carried out, this agreement is quite close. It seems possible that even closer agreement might be achieved by using larger fragment libraries, but this would come at the expense of more time spent examining the fits of fragments to the map. Alternatively, the agreement could be improved by refinement of the models that are obtained.

The author is grateful to the NIH for generous support. This work was carried out as part of the *PHENIX* project and the methods described here are implemented in the software *RESOLVE* (Terwilliger, 2000), available from http://solve.lanl.gov.

References

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). *Nucleic Acids Res.* **28**, 235–242.
Chang, W. R., Jiang, T., Wan, Z. L., Zhang, J. P., Yang, Z. X. & Liang, D. C. (1996). *J. Mol. Biol.* **262**, 721–731.
Chen, C. C. H., Kim, A., Zhang, H., Howard, A. J., Sheldrick, G. M., Dunaway-Mariano, D. & Herzberg, O. (2000). Am. Crystallogr. Assoc. Annu. Meet., Abstract 02.06.03.
Cowtan, K. D. (1998). *Acta Cryst.* D**54**, 750–756.
Cowtan, K. D. (2001). *Acta Cryst.* D**57**, 1435–1444.
Feigenbaum, E. A., Engelmore, R. S. & Johnson, C. K. (1977). *Acta Cryst.* A**33**, 13–18.
Fitz-Gibbon, S. T., Ladner, H., Kim, U. J., Stetter, K. O., Simon, M. I. & Miller, J. H. (2002). *Proc. Natl Acad. Sci. USA*, **99**, 984–989.
Greer, J. (1985). *Methods Enzymol.* **115**, 206–224.
Hobohm, U., Scharf, M. & Schneider, R. (1993). *Protein Sci.* **1**, 409–417.
Holton, T., Ioerger, T. R., Christopher, J. A. & Sacchettini, J. C. (2000). *Acta Cryst.* D**56**, 722–734.
Huber, A. H., Nelson, W. J. & Weis, W. I. (1997). *Cell*, **90**, 871–882.
Jones, T. A. & Thirup, S. (1986). *EMBO J.* **5**, 819–822.
Jones, T. A., Zou, J.-Y., Cowan, S. W. & Kjeldgaard, M. (1991). *Acta Cryst.* A**47**, 110–119.
Kleywegt, G. J. & Jones, T. A. (1997). *Acta Cryst.* D**53**, 179–185.
Lamzin, V. S. & Wilson, K. S. (1993). *Acta Cryst.* D**49**, 129–147.
Levitt, D. G. (2001). *Acta Cryst.* D**57**, 1013–1019.
McRee, D. (1999). *J. Struct. Biol.* **125**, 156–165.
Massova, I., Martin, P., deMel, S., Tanaka, Y., Edwards, B. & Mobashery, S. (1996). *J. Am. Chem. Soc.* **118**, 12479–12480.
Morris, R. J., Perrakis, A. & Lamzin, V. S. (2002). *Acta Cryst.* D**58**, 968–975.

electronic reprint

Oldfield, T. (2002). *Acta Cryst.* D**58**, 487–493.
Pavelcik, F., Zelinka, J. & Otwinowski, Z. (2002). *Acta Cryst.* D**58**, 275–283.
Peat, T. S., Newman, J., Waldo, G. S., Berendzen, J. & Terwilliger, T. C. (1998). *Structure*, **6**, 1207–1214.
Pédelacq, J.-D., Piltch, E., Liong, E. C., Berendzen, J., Kim, C.-Y., Rho, B.-S., Park, M. S., Terwilliger, T. C. & Waldo, G. S. (2002). *Nature Biotechnol.* **20**, 927-932.
Perrakis, A., Morris, R. M. & Lamzin, V. S. (1999). *Nature Struct. Biol.* **6**, 458–463.
Rozwarski, D. A., Diederichs, K., Hecht, R., Boone, T. & Karplus, P. A. (1996). *Proteins*, **26**, 304–313.
Skinner, M. M., Zhang, H., Leschnitzer, D. H., Guan, Y., Bellamy, H., Sweet, R. M., Gray, C. W., Konings, R. N. H., Wang, A. H.-J. & Terwilliger, T. C. (1994). *Proc. Natl Acad. Sci. USA*, **91**, 2071–2075.
Swanson, S. M. (1994). *Acta Cryst.* D**50**, 695–708.
Terwilliger, T. C. (2000). *Acta Cryst.* D**56**, 965–972.
Terwilliger, T. C. (2001). *Acta Cryst.* D**57**, 1755–1762.
Vojtechovsky, J., Chu, K., Berendzen, J., Sweet, R. M. & Schlichting, I. (1999). *Biophys. J.* **77**, 2153–2174.
Yarbrough, D., Wachter, R. M., Kallio, K., Matz, M. V. & Remington, S. J. (2001). *Proc. Natl Acad. Sci. USA*, **98**, 462–467.

electronic reprint

electronic reprint

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Automated side-chain model building and sequence assignment by template matching

Thomas C. Terwilliger

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Automated side-chain model building and sequence assignment by template matching

Thomas C. Terwilliger

Mail Stop M888, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence e-mail: terwilliger@lanl.gov

Received 1 July 2002
Accepted 1 October 2002

An algorithm is described for automated building of side chains in an electron-density map once a main-chain model is built and for alignment of the protein sequence to the map. The procedure is based on a comparison of electron density at the expected side-chain positions with electron-density templates. The templates are constructed from average amino-acid side-chain densities in 574 refined protein structures. For each contiguous segment of main chain, a matrix with entries corresponding to an estimate of the probability that each of the 20 amino acids is located at each position of the main-chain model is obtained. The probability that this segment corresponds to each possible alignment with the sequence of the protein is estimated using a Bayesian approach and high-confidence matches are kept. Once side-chain identities are determined, the most probable rotamer for each side chain is built into the model. The automated procedure has been implemented in the *RESOLVE* software. Combined with automated main-chain model building, the procedure produces a preliminary model suitable for refinement and extension by an experienced crystallographer.

1. Introduction

Building side chains of an atomic model into an electron-density map is a very different problem to building the main chain. Normally the main chain is built first, so the approximate location of the C^{β} atom of the side chain and the approximate direction of the $C^{\alpha}-C^{\beta}$ and $C^{\alpha}-C$ bonds are already known by the time side chains are built. On the other hand, the identities of the side chains at each position in the map are normally not known at this stage, and at moderate resolution (~2–3 Å) in a map of moderate quality it can be very difficult to distinguish many of the side chains. Consequently, the bulk of the problem is not placing the side chains but rather identifying them.

A number of methods have been developed to address the problem of identifying and fitting side-chain density. Most of the methods use the electron density at the coordinates of atoms of a side-chain model to evaluate the model. Jones *et al.* (1991) used a rotamer library (Ponder & Richards, 1987) to assist in manual fitting of side chains. Oldfield (2002) and Levitt (2001) fit a rotamer libraries to side-chain density, considering density at the positions of the atomic coordinates. In contrast, Morris *et al.* (2002) directly build a side-chain model from the coordinates of free atoms representing peaks in the density. In another very different approach, Holton *et al.* (2000) used machine-learning techniques to identify side chains in a map using methods that are rotation-independent, so that they require C^{α} positions but not the directions of the

© 2003 International Union of Crystallography
Printed in Denmark – all rights reserved

electronic reprint

$C^{\alpha}-C^{\beta}$ or $C^{\alpha}-C$ bonds; the fitting of side chains to the density is then carried out by maximizing the correlation of density with that from a side-chain template. Pavelcik *et al.* (2002) described a method for template matching of arbitrary fragments of structure to a map by a rotation/translation search that could be used to identify side chains without knowledge of the main-chain coordinates.

Here, we describe a method for side-chain identification that uses the correlation of side-chain rotamer templates with density in the map to evaluate the fit of a side chain to the map. The side-chain templates are built from average density in refined models so that they reflect the patterns of density found in real structures. A Bayesian approach is used to estimate the probability that each side chain is present at each site. These probabilities are then used to align the protein sequence to the main-chain tracing of the map.

2. Methods

2.1. Side-chain rotamer library

Libraries of side-chain rotamers have been constructed numerous times (Ponder & Richards, 1987; Dunbrack & Cohen, 1997; Lovell *et al.*, 2000) and it would be possible to use one of these as the basis for modeling side chains. However, for the present purposes it was necessary to have a paired rotamer library and corresponding averaged template density library, so it was most convenient to construct both libraries at once from the same database. A library of side-chain rotamers was constructed using 574 refined protein structures chosen arbitrarily from non-redundant PDB files (Berman *et al.*, 2000; Hobohm *et al.*, 1993) with R factors of 20% or lower and a resolution of 1.8 Å or better. In order to limit the total number of rotamers for all amino acids, the maximum number of rotamers considered for any one amino acid was 40. For each amino-acid position in these refined protein structures, the coordinates of the main-chain N, C^{α} and C atoms were used to place the side chain in a standard orientation. Then, for each amino-acid type, all conformations of the amino-acid side chain in this group of structures were listed. A library was generated consisting of the smallest subset of these conformations that could be found such that every conformation in the list differed from a member of the library by at most 0.8 Å r.m.s. For several amino acids (methionine, lysine, glutamine, glutamate, asparagine and tyrosine) this was not possible with just 40 rotamers in the library and in these cases some configurations are not represented. Additionally, for arginine the $N^{\eta 1}$ and $N^{\eta 2}$ atoms were not included in the r.m.s.d. calculation for defining the libraries; even so, more than 40 rotamers would have been required to represent all configurations found and the list was truncated at 40 rotamers. A total of 503 side-chain rotamers were present in the entire library of side chains.

2.2. Side-chain rotamer electron-density templates

The library of side-chain rotamers was used to cluster all conformations of each side chain from the 574 refined protein structures into rotamer groups. These groups contained from one to 18 543 conformations. For each rotamer group, a template was then constructed from the average electron density for the entire group, calculated using the coordinates and thermal factors from the refined structures (mapped into the standard orientation). In this way, the electron-density templates reflected both the variability in side-chain conformation within a rotamer group and the pattern of thermal factors from atom to atom in a rotamer. The electron-density templates were sampled on a grid with a spacing of 1 Å. All points within 3 Å of an atom in a side chain in one or more of the conformations present in the corresponding rotamer group were contained in the templates and all other points excluded. The templates for different amino acids and for different rotamers were therefore sampled at partially overlapping sets of lattice points. The region defining the template for glycine is not well defined by these criteria and as a special case the template for glycine was calculated in the same region as the template for alanine.

2.3. Estimation of the side-chain probability

The relative probability that each of the 20 amino-acid side chains was located at a side-chain position in a polypeptide chain was estimated in several steps. The overall strategy at each position in the chain was to find the rotamer of each side chain that fitted the density best, then to use the fits of these 20 side chains to the density to obtain the probability for each possible side chain at that position.

Firstly, the correlation coefficient cc_{jk} of each side-chain rotamer density template j with the electron density at each side-chain position k in the polypeptide chain was determined. This was accomplished after transforming the density in the map to the standard reference frame defined by the main-chain N, C^{α} and C positions. For each side-chain type, only the best-fitting rotamer was considered further.

Next, a Z score was calculated for the fit of each side-chain template to each side-chain position. The Z score was based on the correlation cc_{jk} for the fit and the correlations for the fits of this template j to all other side-chain positions. The Z score calculated in this way describes how likely it is that the value of the correlation cc_{jk} would be obtained by chance. This was used to estimate the probability of measuring a value cc_{jk} for a template that is incorrect. We further assumed that the correct template can have any value of the correlation. This is a useful assumption because although we expect that a correct template will have a high value of the correlation, it is difficult to specify how high this value should be.

Using these assumptions, the mean correlation for the template j to all side-chain positions, $\langle cc_j \rangle$, was used as an estimate of the correlation to be expected for this side chain to arbitrary side-chain density (*i.e.* generally not associated with this rotamer). Similarly, the standard deviation of this correlation, $\langle \sigma_j \rangle$, was used as an estimate of the variation of this correlation to be expected for arbitrary side chains. Then the Z score, $Z_{jk} = (cc_{jk} - \langle cc_j \rangle)/\sigma_j$, was expected to be related to the probability of obtaining this value of the correlation by

electronic reprint

chance (that is, if side chain j is *not* the correct side chain at position k) using the relation $p(\mathrm{cc}_{jk}) \simeq \exp(-Z_{jk}^2/2)$. In order to prevent very poor fits from being confused with very good ones, $p(\mathrm{cc}_{jk})$ was taken to be unity for values of $Z < 0$ in this calculation.

Finally, the probability that amino-acid type i is the correct one at position k was calculated from Bayes' rule. The *a priori* probability for each amino acid p_{oj} was estimated from numbers of each amino acid j in the protein, n_j. The probability of observing the set of correlations $\{\mathrm{cc}_{jk}\}$ at position k if the correct amino acid at this position is i is the product over all of the other amino-acid types j (not including the correct one, i) of the probabilities of observing the correlations $\{\mathrm{cc}_{jk}\}$ by chance,

$$P(\{\mathrm{cc}_{jk}\}|i \text{ is the correct amino acid}) \simeq \exp(-\sum_{j\neq i} Z_{jk}^2/2).$$

Using Bayes' rule, we then estimate the probability, p_{ik}, that amino acid i is the correct one at position k, yielding, after some simplification,

$$p_{ik} = p_{oi}\exp(Z_{ik}^2/2)/[\sum_i p_{oi}\exp(Z_{ik}^2/2)]. \quad (1)$$

2.4. Alignment of fragments of main-chain model to the protein sequence

A fragment of main-chain model containing n residues was matched to the protein sequence using the matrix of probability estimates p_{ik} describing the probability that amino acid i is the correct type at position k in the model. As the main-chain model might sometimes be missing amino acids (commonly at loop positions) or might cross from one segment of chain to another incorrectly, a first step in the alignment was to identify the sub-fragment of the fragment which could be matched to the sequence with the highest probability. This was considered likely to be the longest segment that is contiguous in the protein chain. The match of this sub-fragment was identified and the remainder of the fragment was then considered independently as a separate fragment.

To accomplish the identification of a contiguous sub-fragment and its match to the sequence, the alignment procedure described next was carried out $n(n-1)/2$ times, once for each possible sub-fragment of the n-residue model fragment. Two different scoring algorithms were used in this process. For comparisons between sub-fragments of the same length, the probability of the alignment of each sub-fragment was used as the score. For comparisons between sub-fragments of different lengths, the probability estimates were not a good indicator of the relative qualities of the alignments and instead the score for each was $\langle Z\rangle N^{1/2}$, where N is the number of residues in the sub-fragment and $\langle Z\rangle$ is the mean Z score $\langle Z_{jk}\rangle$ of the amino acids in this alignment.

For a given sub-fragment or fragment with m residues, all possible alignments l of the model with m sequential amino acids in the protein were considered. The relative probability p_l that alignment l, with amino-acid type t_k matched to position k in the model, was correct was estimated assuming that the probability estimates for all the residues in the model were independent, leading to

$$p_l = \prod_k p_{t_k} / \sum_l \left(\prod_k p_{t_k}\right). \quad (2)$$

The sequence alignment was considered to be reliably identified if the probability for one match was at least 95% (that is, the combined probability of all other matches was 5% or less). Once a reliable sequence alignment was identified, the amino-acid sequence was mapped onto the model fragment. At each residue, the most probable rotamer corresponding to the amino acid assigned to that residue was built into the model.

3. Results and discussion

The reliability of the probabilistic method described here for side-chain assignment and sequence alignment was examined using the density-modified electron-density map for eight different experimental maps with varying resolution, quality (figure of merit) and number of residues in the asymmetric unit. The main chain was built into each map as described in Terwilliger (2003). The coordinates of the main-chain atoms were then used to place side chains.

We first evaluated the utility of (1) by determining how well it actually predicts the probability that a particular side chain is present at a particular position in the model. Fig. 1 shows a histogram of the fraction of correct amino-acid side-chain assignments as a function of the probability assessed using (1). A total of 4349 side-chain densities from the eight experimental electron-density maps were compared with 20 side-chain templates to generate the histogram. The correct side chain at each position was identified from the refined model of the corresponding protein. Overall, Fig. 1 shows that the probability estimates obtained from (1) give a very good indication of the actual probability that the assignments are correct for these test cases.

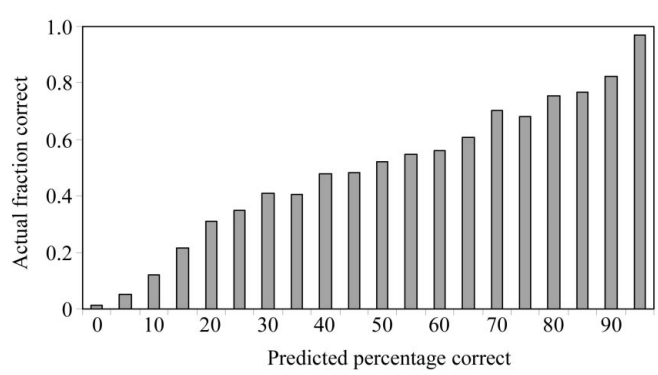


Figure 1
Fraction of correct amino-acid side-chain assignments as a function of the probability estimated from (1). For each residue in the main-chain models for the eight structures listed in Table 1, the relative probabilities for each of the possible side chains were obtained using (1). The correct side chains were identified as the nearest amino acid in the refined model of each structure. The fraction of correct amino-acid side-chain assignments is tabulated as a function of the probability estimates.

electronic reprint

Table 1
Test structures for which side-chain models have been built with *RESOLVE*.

Structure	Resolution (Å)	Figure of merit $\langle m \rangle$	Residues in refined model	Main chain built (%)	Side chains built (%)	Correct alignment (%)	Side-chain mean coordinate error (Å)	Side-chain r.m.s. coordinate error (Å)
Gene 5 protein (Skinner *et al.*, 1994)	2.6	0.62	87	61	11	100	1.2	1.4
Granulocyte-stimulating factor (Rozwarski *et al.*, 1996)	3.5	0.70	242	50	0	N/A	0	0
Initiation factor 5A (Peat *et al.*, 1998)	2.1	0.85	136	84	84	99	1.3	1.8
β-Catenin (Huber *et al.*, 1997)	2.7	0.72	455	81	62	100	1.2	1.7
NDP kinase (Pédelacq *et al.*, 2002)	2.6	0.56	556 (3 × 186)	56	37	98	1.2	1.6
Hypothetical (*P. aerophilum* ORF, NCBI accession No. AAL64711; Fitz-Gibbon *et al.*, 2002)	2.6	0.58	494 (2 × 247)	79	75	98	1.3	2.0
Red fluorescent protein (Yarbrough *et al.*, 2001)	2.5	0.91	936 (4 × 234)	88	88	99	1.2	1.8
2-Aminoethylphosphonate (AEP) transaminase (Chen *et al.*, 2000)	2.6	0.84	2232 (6 × 372)	85	81	99	1.3	1.8

We next evaluated how well the sequence-alignment probabilities estimated with (2) relate to the actual probabilities. Fig. 2 shows a histogram of the fraction of correct sequence alignments as a function of the probabilities of correct assignments estimated using (2). Assignments were considered correct if 90% or more of the residues in the alignment match the closest residue in the refined structure. A total of 85 very strong predictions with confidence >97% were obtained and all but one of these were correct assignments. Overall, the probability estimates for sequence alignment using (2) are fairly accurate, though there is less of a clear discrimination among alignments with moderate probability than in the case of the side-chain assignments.

The overall side-chain modeling results are summarized in Table 1. The side-chain modeling procedure identified and aligned the protein sequence to 71% of the 5131 residues in eight proteins. These eight proteins included two that were used in the development of the method (NDP kinase and gene 5 protein). They also included two that were among those in the database of proteins used to construct side-chain templates (gene 5 protein, PDB code 1vqb; β-catenin, PDB code 1dow). These three may therefore be slightly better fitted than a typical protein. The remaining five are likely to be relatively good indicators of the utility of the method. For the set of eight proteins as a whole, 99% of the sequence assignments were correct. The mean difference in coordinates between the side-chain atoms of the model and those of the corresponding refined structures was 1.3 Å and the r.m.s. difference was 1.8 Å, including lysine and arginine residues, where the positions of atoms in even the refined structures is often somewhat uncertain, but excluding atoms more than 10 Å away from any atom in the refined structures.

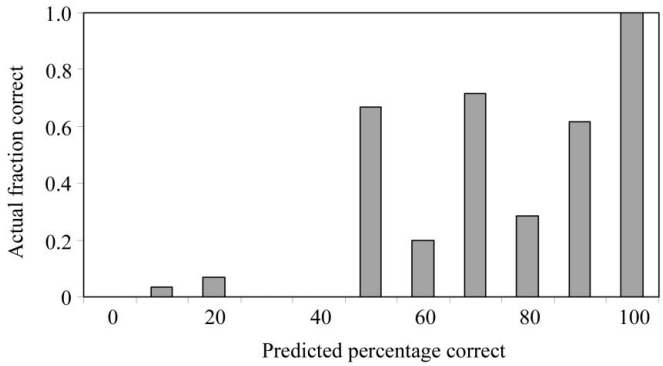


Figure 2
Fraction of correct fragment alignments as functions of the probabilities estimated from (2). For each main-chain fragment built, the sub-fragment with the highest weighted Z score was identified as described in §2. All alignments of this sub-fragment with the protein sequence were considered and the relative probabilities of each alignment were estimated with (2). An alignment was considered correct if the residue numbers of 90% of the residues in the fragment matched those of the nearest amino acid in the refined model.

To test the resolution-dependence of side-chain model building, the IF5A structure was built at a variety of resolutions, truncating the data in each case. This of course does not fully simulate the ability to build a model for a poorly diffracting crystal, as the data and phases are very good to the resolution cutoff in this test. Nevertheless, it can give an idea of what is possible with very good data. Fig. 3(*a*) shows the number of main-chain residues and side chains built as a function of the high-resolution cutoff. Fig. 3(*b*) shows the r.m.s. coordinate error in main-chain and side-chain atoms for the same models. Fig. 3 illustrates that in the presence of very good data, as much as 75% of the main chain and 50% of the side chains can be built at a resolution as low as 3.4 Å and with an r.m.s. coordinate error that is only slightly higher than at a resolution of 2.1 Å.

4. Conclusions

The probabilistic methods described here for identifying side chains and their rotamers in the electron density at positions derived from a main-chain tracing are found to be very effective. With a map of reasonable quality and a segment of ten residues or longer, the alignment to the sequence can often be identified with a confidence greater than 98%.

As a side effect of aligning the model to the sequence, the quality of the main-chain protein tracing can be considerably improved. This is largely owing to the identification of errors

electronic reprint

in the main chain and elimination of the corresponding segments of main-chain model. For example, mis-tracings caused by tracing the chain through a loop region using fewer residues than are present in the actual loop are removed in this procedure. The procedure of finding the longest sub-fragment of a main-chain fragment that had a very strong match to the protein sequence was very useful for identifying these cases. This removal of residues is the reason for the differences between the number of main-chain residues in the models in this work compared with those obtained when using only the main-chain tracing algorithm (Terwilliger, 2003).

The algorithm for side-chain fitting and alignment developed here does have significant limitations. The side-chain rotamer libraries used are not as complete or as accurate as others available (*e.g.* Dunbrack & Cohen, 1997; Lovell *et al.*, 2000). This means that not all reasonable rotamers in proteins can be accurately represented by one of those in the library and that some side chains will therefore be poorly fitted. This could be improved by using a more complete library, but at a cost of examining a larger number of templates for a fit to the electron-density map. A possible compromise would be to use a more complete library only for those side chains that are not well fitted with a rotamer from the standard one. It could also be improved by using a filtered library such as that of Lovell *et al.* (2000), which removes conformations that are unlikely to be correct, or by explicitly checking side chains for poor contacts. An additional limitation is that some common situations are not recognized by the rotamer libraries (and by the main-chain model building that precedes side-chain fitting). These include disulfide bonds in proteins, unusual amino acids and all non-protein electron density. In most of these situations, the model simply does not include the corresponding region. There could be cases where such density is misinterpreted in terms of main-chain and side-chain conformations that are in the corresponding libraries, however.

The author would like to thank Li-Wei Hung for discussion and the NIH for generous support. The work was carried out as part of the *PHENIX* project and the methods described here are implemented in the software *RESOLVE* (Terwilliger, 2000), available from http://solve.lanl.gov.

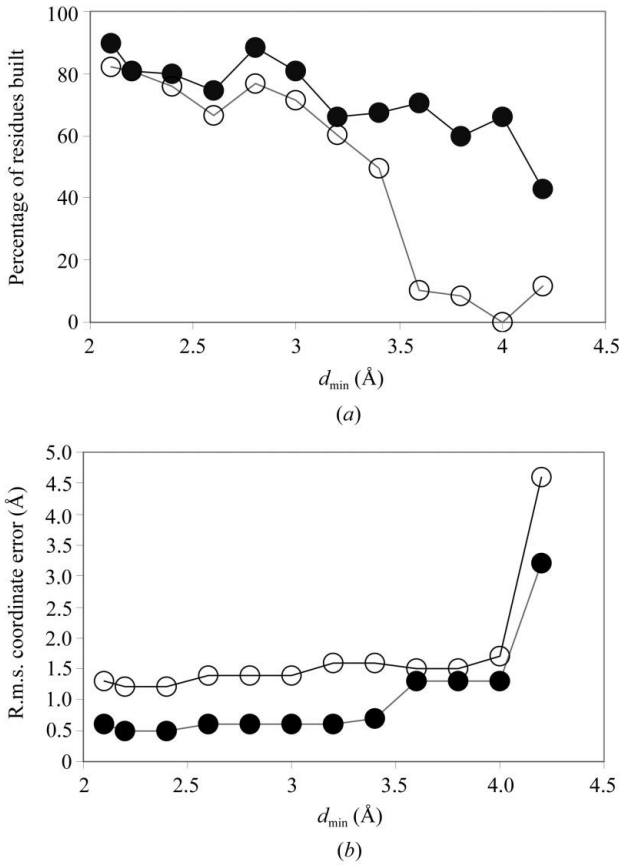


Figure 3
Effect of resolution on model building of IF5A. The phases and amplitudes for IF5A (Peat *et al.*, 1998) after density modification were truncated at varying resolutions and the resulting maps were used for automated main-chain and side-chain model building. (*a*) Percentage of the main chain (closed circles) and the side chains (open circles) in the refined structure that were built. (*b*) R.m.s. coordinate error for main-chain (closed circles) and side-chain (open circles) atoms. Side-chain atoms include C^{β}.

References

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). *Nucleic Acids Res.* **28**, 235–242.
Chen, C. C. H., Kim, A., Zhang, H., Howard, A. J., Sheldrick, G. M., Dunaway-Mariano, D. & Herzberg, O. (2000). Am. Crystallogr. Assoc. Annu. Meet., Abstract 02.06.03.
Dunbrack, R. L. & Cohen, F. E. (1997). *Protein Sci.* **6**, 1661–1681.
Fitz-Gibbon, S. T., Ladner, H., Kim, U. J., Stetter, K. O., Simon, M. I. & Miller, J. H. (2002). *Proc. Natl Acad. Sci. USA*, **99**, 984–989.
Hobohm, U., Scharf, M. & Schneider, R. (1993). *Protein Sci.* **1**, 409–417.
Holton, T., Ioerger, T. R., Christopher, J. A. & Sacchettini, J. C. (2000). *Acta Cryst.* D**56**, 722–734.
Huber, A. H., Nelson, W. J. & Weis, W. I. (1997). *Cell*, **90**, 871–882.
Jones, T. A., Zou, J. Y., Cowan, S. W. & Kjeldgaard, M. (1991). *Acta Cryst.* A**47**, 110–119.
Levitt, D. G. (2001). *Acta Cryst.* D**57**, 1013–1019.
Lovell, S. C., Word, J. M., Richardson, J. S. & Richardson, D. C. (2000). *Proteins Struct. Funct. Genet.* **40**, 389–408.
Morris, R. J., Perrakis, A. & Lamzin, V. S. (2002). *Acta Cryst.* D**58**, 968–975.
Oldfield, T. (2002). *Acta Cryst.* D**58**, 487–493.
Pavelcik, F., Zelinka, J. & Otwinowski, Z. (2002). *Acta Cryst.* D**58**, 275–283.
Peat, T. S., Newman, J., Waldo, G. S., Berendzen, J. & Terwilliger, T. C. (1998). *Structure*, **6**, 1207–1214.
Pédelacq, J.-D., Piltch, E., Liong, E. C., Berendzen, J., Kim, C.-Y., Rho, B.-S., Park, M. S., Terwilliger, T. C. & Waldo, G. S. (2002). *Nature Biotechnol.* **20**, 927–932.
Ponder, J. W. & Richards, F. M. (1987). *J. Mol. Biol.* **193**, 775–791.
Rozwarski, D. A., Diederichs, K., Hecht, R., Boone, T. & Karplus, P. A. (1996). *Proteins*, **26**, 304–313.
Skinner, M. M., Zhang, H., Leschnitzer, D. H., Guan, Y., Bellamy, H., Sweet, R. M., Gray, C. W., Konings, R. N. H., Wang, A. H.-J. & Terwilliger, T. C. (1994). *Proc. Natl Acad. Sci. USA*, **91**, 2071–2075.
Terwilliger, T. C. (2000). *Acta Cryst.* D**56**, 965–972.
Terwilliger, T. C. (2003). *Acta Cryst.* D**59**, 38–44.
Yarbrough, D., Wachter, R. M., Kallio, K., Matz, M. V. & Remington, S. J. (2001). *Proc. Natl Acad. Sci. USA*, **98**, 462–467.

electronic reprint

electronic reprint

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Improving macromolecular atomic models at moderate resolution by automated iterative model building, statistical density modification and refinement

Thomas C. Terwilliger

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Improving macromolecular atomic models at moderate resolution by automated iterative model building, statistical density modification and refinement

Thomas C. Terwilliger

Mail Stop M888, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence e-mail: terwilliger@lanl.gov

Received 18 January 2003
Accepted 6 May 2003

An iterative process for improving the completeness and quality of atomic models automatically built at moderate resolution (up to about 2.8 Å) is described. The process consists of cycles of model building interspersed with cycles of refinement and combining phase information from the model with experimental phase information (if any) using statistical density modification. The process can lead to substantial improvements in both the accuracy and completeness of the model compared with a single cycle of model building. For eight test cases solved by MAD or SAD at resolutions ranging from 2.0 to 2.8 Å, the fraction of models built and assigned to sequence was 46–91% (mean of 65%) after the first cycle of building and refinement, and 78–95% (mean of 87%) after 20 cycles. In an additional test case, an incorrect model of gene 5 protein (PDB code 2gn5; r.m.s.d. of main-chain atoms from the more recent refined structure 1vqb at 1.56 Å) was rebuilt using only structure-factor amplitude information at varying resolutions from 2.0 to 3.0 Å. Rebuilding was effective at resolutions up to about 2.5 Å. The resulting models had 60–80% of the residues built and an r.m.s.d. of main-chain atoms from the refined structure of 0.20 to 0.62 Å. The algorithm is useful for building preliminary models of macromolecules suitable for an experienced crystallographer to extend, correct and fully refine.

1. Introduction

Iterative model building and refinement has proven to be an exceptionally powerful tool for automatic interpretation of macromolecular electron-density maps where the diffraction data extend beyond about 2.3 Å (Lamzin & Wilson, 1993; Perrakis *et al.*, 1997, 1999, 2001; Morris *et al.*, 2002). In this approach, implemented in *ARP* (Lamzin & Wilson, 1993) and later in *wARP* (Perrakis *et al.*, 1999), electron density in a map is interpreted initially in terms of peaks corresponding to atomic coordinates. These 'free atoms' are subsequently refined and interpreted in terms of a macromolecular structure, which can be further refined. The refined model or models are then used to provide updated estimates of crystallographic phases, leading to a new electron-density map, and the process is repeated until no further improvements to the model occur.

The free-atom model-building approach works well when data is available to near-atomic resolution or better (<2.3 Å; Perrakis *et al.*, 1999), but is limited by the need to identify peaks of density at the positions of atomic coordinates. At lower resolution, atoms are not well defined in the electron density and the free-atom method of initiating model building has not been as useful, although related methods have been used to improve electron-density maps at resolutions up to 3 Å (Vellieux, 1998). Recently, several methods for automated

© 2003 International Union of Crystallography
Printed in Denmark – all rights reserved

electronic reprint

model building at moderate resolution (<3 Å) have been developed. Each of these methods relies on features of macromolecular electron-density maps on a larger scale than individual atoms to begin model building. Oldfield (2002) described a method to identify helices and sheets and then extended these segments one amino acid at a time to trace a polypeptide. Levitt (2001) uses an interpretation of the connected regions of the map (the 'bones' of Greer, 1985) to identify helices and sheets and then also extends them to trace a polypeptide. Ioerger & Sacchettini (2002) used a pattern-matching approach to identify C^{α} positions and trace polypeptide backbones. We recently described another method (Terwilliger, 2001*a*, 2003*a*,*b*) for identifying the locations of helices and sheets based on the template-convolution method of Cowtan (1998), followed by correlation-based refinement of the position and orientations of the templates and choosing a fragment of a helical or sheet region from a library constructed from refined protein structures. These helices and sheets are then extended using tripeptide fragments from a library constructed from a set of refined protein structures.

Here, we show that the quality and completeness of automatic model building at moderate resolution can be substantially improved by alternating model-building cycles with cycles of phase improvement. The phase improvement is carried out with statistical density modification (previously known as maximum-likelihood density modification; Terwilliger, 2000) and can include information based on the refined partial model, information from experiments and information from classical density-modification sources such as solvent flattening and non-crystallographic symmetry.

2. Methods

2.1. Initial phase calculations from SAD or MAD data

Initial phase calculations were carried out using statistical density modification with *RESOLVE* (Terwilliger, 2000) based on phase probability distributions obtained from SAD or MAD data using *SOLVE* (Terwilliger & Berendzen, 1999). Non-crystallographic symmetry (NCS) in the structures was identified from any NCS present in the heavy-atom sites and was verified by analysis of the correlation of density at NCS-related positions in the *SOLVE* electron-density map (Terwilliger, 2002*a*,*b*). NCS was used as a source of prior information about the electron-density map in much the same way as the flatness of the solvent region (Terwilliger, 2000, 2002*b*). The statistical density-modified map and the NCS operations, if any, were used as the input to automated model building.

2.2. Model building

Automated model building was carried out as described previously (Terwilliger, 2003*a*,*b*). This procedure requires an electron-density map, the sequences of any protein chains and any non-crystallographic symmetry information that is available. It produces an atomic model consisting of linked fragments of polypeptide chain from fragment libraries and side chains from rotamer libraries.

2.3. Refinement

Restrained maximum-likelihood refinement was carried out with *REFMAC*5 (version 5.1.24; Murshudov *et al.*, 1997) and default parameters for a poor low-resolution model, except that no scaling of reliability of phases was performed. Phase information from the current best phase set was included in refinement. Overall thermal factor refinement was used with tight restraints (Wmat = 0.15) and damping of shifts was included (Pdamp = 0.5, Bdamp = 0.5). A bulk-solvent model was included with Bbulk = 200 and SCbulk = −0.05. It should be noted that these parameters were not optimized and that optimal values are likely to depend on the resolution of the data and the quality of the model. A total of 20 cycles of refinement were carried out for each application of *REFMAC*5. Reflections were divided randomly into a test set (5%) and a working set (95%) at the beginning of iterative refinement and the same test set was used throughout the process. Non-crystallographic symmetry restraints were not included in refinement; however, some model-based non-crystallographic symmetry information could be propagated through the image-based phasing procedure (which includes non-crystallographic symmetry), so there is a possibility that the free *R* factors for cases with non-crystallographic symmetry could be slightly biased. A user-defined test set can be read in using the *CCP*4 conventions (Collaborative Computational Project, Number 4, 1994) in order to reduce this potential non-crystallographic symmetry bias (Kleywegt, 1996).

2.4. Estimation of electron density based on one model

Electron density was calculated from unrefined or partially refined models in two steps. Firstly, electron density was calculated directly from the model for all points within the distance rad_max of an atom, where rad_max corresponds to the resolution of the data or 2.5 Å, whichever is larger. The electron density calculated in this way is therefore only defined at points near to atoms. An overall thermal factor and an incremental thermal factor for side-chain atoms (depending on the number of bonds between the atom and C^{α}) were then estimated by maximizing the correlation of the calculated electron density with the density in the current best electron-density map. In cases where no prior electron-density map exists, these parameters were not optimized.

2.5. Estimation of electron density based on several non-independent models

To combine estimates of electron density from several atomic models, a real-space procedure related to the reciprocal-space weighting procedure of Perrakis *et al.* (1997) was used. The potential advantage of a real-space averaging method is that two models that cover partially overlapping regions of the asymmetric unit can be combined in different ways in the regions where they overlap and the regions where

electronic reprint

only one model has density. Two methods were used to combine electron density from multiple models. In the first (unweighted average) method, the electron density at each point was the simple average of the electron-density values for all models that have density defined at that point. In the second (weighted average) method, the covariances of the electron densities for each pair of the various models were calculated in the regions where both members of each pair are defined. This covariance matrix was then used to calculate a minimum-variance estimate of the electron density as described by Read (2001). This calculation requires estimates of the correlations between each electron-density map and the true map. These correlations and their overall average cc_{avg} were estimated as the mean correlations of F_{obs} with F_{calc}, estimated in shells of resolution. Although the map correlation and the structure-factor amplitude correlation are not expected to be equal, they have the same range (−1 to 1), similar values and similar trends (increasing values with increasing quality of the model), which is sufficient for the present purpose. In cases where the covariance matrix was singular or any weights on any electron-density maps were negative, the map with the most negative weight was removed and the calculation was repeated. For all points where electron density from some models was not defined, the weights on the remaining models were increased to yield the same sum of weights. The two methods of estimation of electron density based on several models were generally both used during a set of cycles of model building and phase recombination, with the weighted average method being used on most cycles and the unweighted average method used every fifth cycle.

2.6. Statistical density modification with an electron-density target for part of the asymmetric unit (image-based phase estimation)

Information about the electron density in part of the asymmetric unit was used as a source of phase information in statistical density modification in the same way as information about solvent flatness or NCS symmetry. For each of these sources of information, an estimate of the probability distribution for possible values of electron density at each point in the map is needed. For the overall distributions of density in the solvent- and macromolecule-containing regions, these distributions have been described (Terwilliger, 2000) and consist of fits of distributions for solvent and protein regions calculated from model data, broadened by Gaussian functions. For NCS-related points in the map, the distributions are modeled by a single Gaussian with a width based on the r.m.s. difference between densities at NCS-related points (Terwilliger, 2002*b*). For the calculated electron-density map, the distributions were also modeled by a single Gaussian function. Model density was scaled to the density in the current best electron-density map (if any) and used as the target electron density. The uncertainty in the target electron-density values σ was calculated from the estimates made above of the mean correlation cc_{avg} of the model and true electron density and the r.m.s. value of the current electron-density map, $\rho_{r.m.s.}$, using the approximate relation $\sigma = \rho_{r.m.s.}(1 - cc_{avg}{}^2)^{1/2}$. If no electron-density map was available, then the r.m.s. value of the model electron-density map was used in this relation instead. Once probability distributions for electron density at each point in the asymmetric unit are defined, the map probability function (previously known as the map likelihood function; Terwilliger, 2001*b*) can be used to estimate phase probabilities from this information alone or in combination with prior phase information.

2.7. Iterative phase combination using statistical density modification

Phase combination by statistical density modification was carried out iteratively. For each iteration, the electron-density map produced in the previous iteration (or a starting density-modified experimental map) was used as the starting electron-density map for density modification. Any prior phase probability information and the starting values of NCS operators used were identical to those used in the initial statistical density-modification calculation. The probability that each point was in the solvent was recalculated after each iteration using the starting electron-density map. In this process, the calculated electron density from the model was the principal source of information about the expected map density that varied from iteration to iteration of the model-building and density-modification process. Three cycles of density modification were carried out during each iteration of statistical density modification. Additional cycles had little effect because all the sources of information about expected values of density in the map were constant during a given iteration and the statistical density-modification procedure converged rapidly. Once density modification was complete, a new map was calculated and the process was repeated.

2.8. Cross-validated statistical density modification with information from a model (omit prime-and-switch phasing)

A reduced-bias electron-density map was calculated from an atomic model in two steps. Firstly, target electron density was estimated from the model as described above and one cycle of image-based phase estimation was carried out to yield a starting set of phases and figures of merit. Next, the asymmetric unit was divided into approximately 20 omit regions. In each cycle of cross-validation, prime-and-switch phasing was carried out as described previously (Terwilliger, 2001*b*) beginning with the image-based starting set of phases, but additionally including the target electron-density map based on the model for all points except those in one omit region (Shah *et al.*, 1997). Three cycles of prime-and-switch phasing with the omit electron-density target were carried out as part of each cycle of cross-validation, yielding an 'omit' electron-density map defined in the region where model electron density was not included. The omit regions from all the cycles of cross-validation were then combined to create a composite 'omit prime-and-switch' electron-density map.

electronic reprint

2.9. Combination of model building and model refinement

Model building and refinement were combined in one of two ways: a simple alternation of model building and refinement and a multi-step procedure of model building, refinement, model extension and side-chain refitting. In the multi-step procedure, a model is built into an electron-density map as described previously (Terwilliger, 2003*a*,*b*). The model is then refined and the refined model is used as a starting point for a model-rebuilding step. In the rebuilding step, chains in the refined model are trimmed back to match electron density in the current map and are then extended using tripeptide-fragment libraries in the same way as during initial model building (Terwilliger, 2003*a*). The side chains are identified in the same fashion (Terwilliger, 2003*b*), except that now the definition of the side-chain orientation is based on a refined model, not the initial model. In the multi-step procedure this refinement, extension and side-chain refitting process was carried out twice. In each iteration of the whole process the model was rebuilt, but fragments of the model from the previous iteration were used as starting points for rebuilding in addition to any helix or strand positions found in the FFT-based pattern-matching process used for initial model building (Terwilliger, 2003*a*).

3. Results and discussion

The key step in this iterative model-building, density-modification and refinement procedure is to use electron density from a refined model as a source of information for statistical density modification. The *ARP*/*wARP* procedure (Perrakis *et al.*, 1999) has demonstrated clearly that a model can be built and refined with some accuracy beginning with a

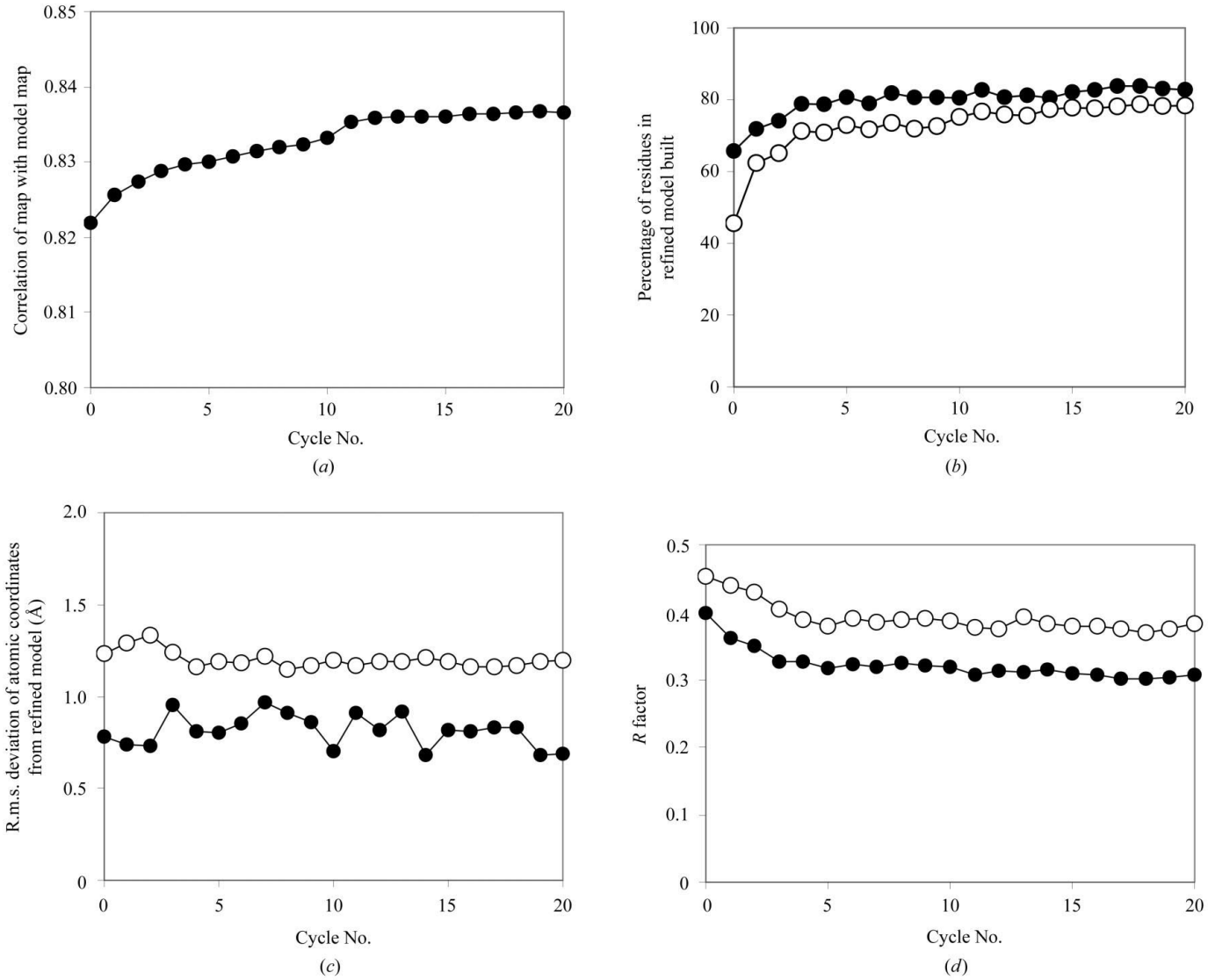


Figure 1
Iterative model building of UTP synthase (Gordon *et al.*, 2001) at 2.8 Å. (*a*) Correlation of statistical density-modified map with map calculated from reference refined model of UTP synthase (Gordon *et al.*, 2001) at the end of each cycle. (*b*) Percentage of main-chain atoms (filled circles) and side-chain atoms (open circles) built in each cycle. (*c*) R.m.s. coordinate difference between models built in each cycle with the reference refined model for main-chain atoms (filled circles) and side-chain atoms (open circles). (*d*) Working *R* factor (filled circles) and free *R* factor (open circles) at the end of each cycle.

electronic reprint

map that has a significant level of noise and that the electron density calculated from such a model can be more accurate (in the region occupied by the model) than the original map. The novel aspects of the present method are the use of a model-building procedure that is effective at moderate resolution (Terwilliger, 2003*a*,*b*) and the use of statistical density modification in the phase-recombination step of iterative model building and refinement.

3.1. Iterative model building, density modification and refinement with experimental phase information

Fig. 1 shows the progress of iterative model building in the case of SAD data from UTP synthase at a resolution of 2.8 Å (Gordon *et al.*, 2001). To evaluate the quality of models built by this procedure, the model of UTP synthase refined at a resolution of 2.0 Å (PDB code 1e8c) was used as a reference. Fig. 1(*a*) shows the correlation of the density-modified map at the beginning of each cycle with the map based on the reference model. On the zeroth cycle this density-modified map is that produced by statistical density modification without using model information (Terwilliger, 2000) and for this UTP synthase SAD data the starting correlation was 0.822. Over the course of 20 cycles of model building, this correlation gradually increased to 0.837. Each of these cycles consisted of density modification using electron density from the current model, model building, refinement of the model and two cycles of chain extension and refinement. In Figs. 1(*b*)–1(*d*), the characteristics of the refined models at the end of each cycle are shown. As in Fig. 1(*a*), the zeroth cycle corresponds to the model built and refined on the basis of the initial density-modified map. In this zeroth cycle of model building, 71% of main-chain residues and 52% of the corresponding side chains were built. By the end of 20 cycles, 79% of the both main-chain residues and side chains were built. The overall accuracy of atomic coordinates improved slightly during the course of model building. In the zeroth cycle the r.m.s. difference in position between main-chain atom coordinates in the model built by the present procedure and those in the refined reference model was 0.78 Å; after 20 cycles it was reduced to

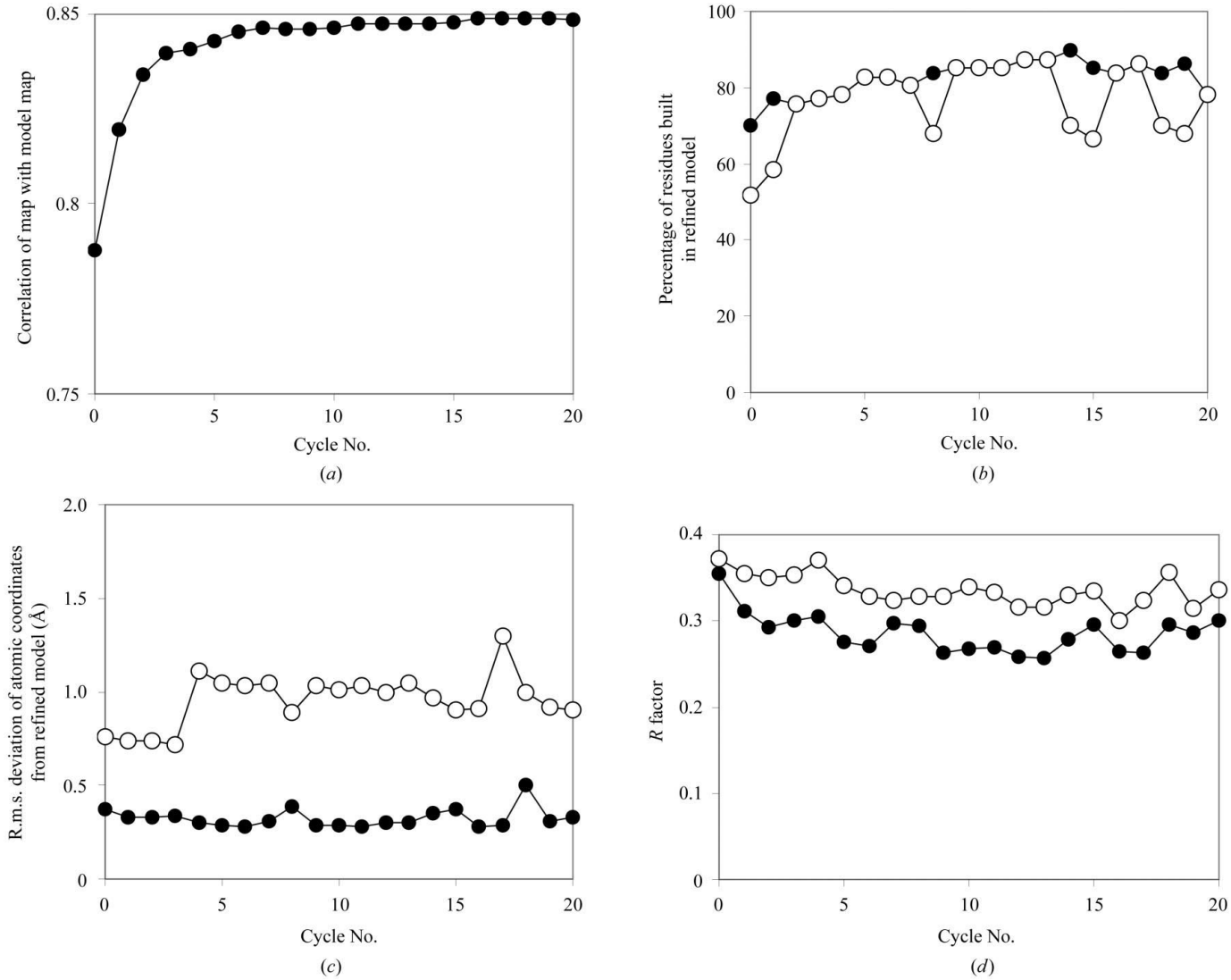


Figure 2
Iterative model building of gene 5 protein (Skinner *et al.*, 1994) at 2.6 Å. (*a*)–(*d*) as in Fig. 1.

electronic reprint

Table 1
Test structures built using iterative model building and refinement.

Structure	UTP synthase (Gordon *et al.*, 2001)	β-Catenin (Huber *et al.*, 1997)	2-Aminoethyl-phosphonate (AEP) transaminase (Chen *et al.*, 2002)	Gene 5 protein (Skinner *et al.*, 1994)	Hypothetical (*P. aerophilum* ORF; NCBI accession No. AAL64711; Fitz-Gibbon *et al.*, 2002)	NDP kinase (Pédelacq *et al.*, 2002)	Initiation factor 5A (Peat *et al.*, 1998)	Red fluorescent protein (Yarbrough *et al.*, 2001)
Resolution (Å)	2.8	2.7	2.6	2.6	2.6	2.4	2.1	2.0
Type of experiment	SAD	MAD	SAD	MAD	MAD	MAD	MAD	MAD
Figure of merit at start of model building $\langle m \rangle$	0.73	0.72	0.84	0.62	0.58	0.56	0.85	0.91
Residues in reference refined model†	1012 (2 × 506)	455	2232 (6 × 372)	86	494 (2 × 247)	556 (3 × 186)	136	936 (4 × 234)
% of main-chain built								
Cycle 1	66	81	92	71	86	59	81	90
Cycle 20	83	95	94	79	95	85	85	91
% of side chains built								
Cycle 1	46	64	91	52	85	53	81	50
Cycle 20	78	86	93	79	95	85	85	91
R.m.s. coordinate difference‡								
Main chain	0.69	0.92	0.48	0.33	0.26	0.31	0.21	0.33
Side chain	1.2	1.25	1.09	0.9	1.14	1.12	0.87	1.16
Change in map correlation with map based on reference refined model† from beginning to 20th cycle	0.015	0.009	0.002	0.061	0.010	0.012	0.003	0.003
Working *R* factor								
Cycle 1	0.40	0.35	0.27	0.36	0.30	0.39	0.33	0.34
Cycle 20	0.31	0.27	0.26	0.30	0.27	0.28	0.31	0.28
Free *R* factor								
Cycle 1	0.45	0.39	0.30	0.37	0.35	0.42	0.33	0.36
Cycle 20	0.38	0.31	0.30	0.34	0.32	0.33	0.32	0.31

† The reference refined model in each case is either the deposited PDB entry for this structure or the unpublished refined structure, in each case built without using *RESOLVE* model building. ‡ R.m.s. coordinate difference between model at the 20th cycle and reference refined model (Å).

0.69 Å. The working *R* factor (at 2.8 Å) of the models decreased from 0.40 at the end of refinement of the initial model in the zeroth cycle to 0.31 in the 20th cycle. The corresponding free *R* factors decreased from 0.45 to 0.38 (however, there could be a slight bias in these free *R* factors as the twofold symmetry of UTP synthase was used in the density-modification steps).

Fig. 2 shows the benefit of iterative model building in the case of slightly higher resolution (2.6 Å) data from gene 5 protein (Skinner *et al.*, 1994). The reference model was PDB entry 1vqb, refined at 1.8 Å (Skinner *et al.*, 1994). The correlation of the density-modified maps with the map based on the reference model improved very substantially from 0.79 to 0.85 during the course of iterative model building in this case. Automatic model building was able to place 79% of the main-chain residues and 52% of side chains in the first cycle and 79% of both main chain and side chains in the 20th cycle (Fig. 2*b*). The r.m.s. difference between main-chain atoms and the refined coordinates of gene 5 protein (Skinner *et al.*, 1994) decreased from about 0.37 to 0.33 Å during the iterative model building (Fig. 2*c*) and for side-chain atoms it became slightly worse overall, increasing from 0.76 to 0.90 Å. The working *R* factor at 2.6 Å decreased from 0.36 to 0.30 during the course of iterative refinement and model building and the free *R* factor decreased from 0.37 to 0.34. Fig. 3 illustrates representative sections of the refined model (in yellow), the model after one cycle of building (red) and the model after 20 cycles of building (green).

Table 1 summarizes iterative model building results for eight proteins, including the UTP synthase and gene 5 protein cases shown in Figs. 1 and 2. In all eight cases, the iteration of model building resulted in a substantially more complete and more accurate model than was obtainable in the first cycle of model building. Overall, the fraction of the models built and assigned to sequence was 46–91% (mean of 65%) after the first cycle of building and refinement, and 78–95% (mean of 87%) after 20 cycles.

The preceding examples show that iterative statistical density modification, model building and refinement can be useful in improving the completeness of atomic models at moderate resolution (at least up to about 2.8 Å) in cases where a starting set of experimental phase probability estimates is available. The experimental phase probabilities are very useful in this procedure because they can be combined with model-based information during every cycle of the process and often contribute as much or more to the phase information as the model.

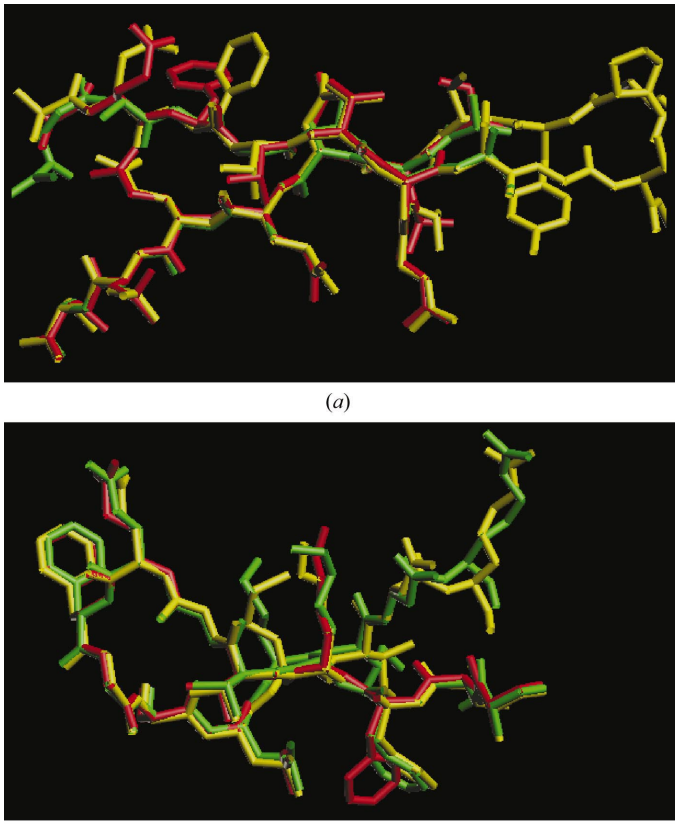
3.2. Iterative model building, density modification and refinement without experimental phase information

A more difficult problem is that of iterative model-building when no experimental phase probability distributions are available, such as in the case of rebuilding models in molecular replacement (Rossmann, 1972). The iterative model-building and refinement process carried out by *ARP*/*wARP* (Perrakis *et*

electronic reprint

al., 1999) has been very successful in this application (Perrakis *et al.*, 2001). In addition to the absence of experimental phase information in this case, model bias arising from the starting model can exist. To reduce model bias, we use a variation on our method of 'prime-and-switch' phasing (Terwilliger, 2001*b*) to calculate a reduced-bias initial electron-density map. In the method described earlier (Terwilliger, 2001*b*), a starting set of phases is calculated from a model and then in an iterative process phases are estimated by maximizing the agreement of the features of the map with expectations (*e.g.* a flat solvent or the presence of NCS), without reference to the starting set of phases. In the variation used here, a similar process is carried out but using some additional information and an 'omit' procedure, as described above. For each cycle, several 'omit' sub-cycles are carried out. In each sub-cycle, a calculated electron-density map is included as the information for image-based phasing (see §2) for all points in the asymmetric unit outside of an 'omit' region. The omitted regions for all the sub-cycles are then combined to form a composite electron-density map.

We used the gene 5 protein structure to test the application of iterative model building, density modification and refinement to a case of model rebuilding. The structure of gene 5 protein has been determined several times by X-ray crystallographic methods (McPherson *et al.*, 1979; Brayer & McPherson, 1983; Skinner *et al.*, 1994). The two more recent determinations were carried out using crystals of gene 5 protein in the same space group *C2* crystal form, first by MIR methods (Brayer & McPherson, 1983) and later by MAD phasing (Skinner *et al.*, 1994). We take the structure of Skinner *et al.* (1994) (PDB code 1vqb) as our reference in this analysis because it is at the higher resolution of these structures (1.8 Å); it has subsequently been refined at even higher resolution (1.6 Å; S. Su, Y.-G. Gao, H. Zhang, T. C. Terwilliger & A. H.-J. Wang, unpublished results; PDB code 1gvp) and it is very similar to a structure built on the basis of NMR data (Folkers *et al.*, 1994). The structure of Brayer & McPherson (1983) (PDB code 2gn5) was determined at the moderate resolution of 2.3 Å and differs from the higher resolution structure 1vqb in the loops and in the register of the β-strands. The overall r.m.s. difference between corresponding protein atoms in 2gn5 and 1vqb is 1.75 Å for main-chain atoms and 3.53 Å for side-chain atoms.



(*a*)

(*b*)

Figure 3
Segments of gene 5 protein models built automatically. (*a*) Residues 11–33. (*b*) Residues 66–80. In each case the refined model is in yellow, the model after one cycle of building is in red and the model after 20 cycles of building is in green. Figures constructed with *O* version 8.0 (Jones *et al.*, 1991).

We used the structure 2gn5 as a starting point for iterative model building, density modification and refinement. In this procedure, the structure-factor amplitudes used were those measured from the *C2* crystal form of gene 5 protein and which had been used as the basis for refinement of the 1vqb structure (Skinner *et al.*, 1994). These structure-factor amplitudes were measured to a resolution of 1.8 Å. For the present purpose, data at varying resolutions were used to assess the utility of the method. Fig. 4(*a*) shows the number of residues built and assigned to sequence using data to 2.3, 2.5 and 2.7 Å. Using data to 2.3 Å, 70 of the 87 residues in gene 5 protein could be built and side chains could be built and correctly assigned to the sequence for all of them. The total number of residues built (whether side chains were built or not) increased from 46 in the first cycle (with six side chains built and assigned to sequence) to 70 in the 50th cycle (with all assigned to sequence). At a resolution of 2.5 Å, 61 residues could be built in 50 cycles, of which 47 residues could be assigned to the sequence. At 2.7 Å, 52 residues could be built in 50 cycles, but just six residues could be assigned to the sequence.

Fig. 4(*b*) shows the r.m.s. coordinate difference between partially refined intermediate models built using data to 2.3, 2.5 and 2.7 Å and the corresponding atoms in the reference model 1vqb (Skinner *et al.*, 1994). At a resolution of 2.3 Å, the r.m.s. coordinate difference decreases from 1.75 Å (for the starting model) to just 0.2 Å over the course of 50 cycles. At resolutions of 2.5 and 2.7 Å the coordinate differences are somewhat higher: 0.62 and 1.02 Å, respectively.

Fig. 4(*c*) shows the number of residues built as a function of resolution as well as the number of side chains placed in the corresponding models, while Fig. 4(*d*) shows the corresponding main-chain coordinate differences from the reference model 1vqb. At resolutions of about 2.5 Å or better, the iterative algorithm is capable of building much of the main chain (61 or more of 87) and side chains (43 or more of 87) and the r.m.s. coordinate difference between these models and the reference model 1vqb is 0.6 Å or less.

electronic reprint

3.3. Basis for model improvement through iterative model building, density modification and refinement

There are several reasons why iterative cycles of model building and density modification might be expected to improve the overall completeness and accuracy of the model produced. The most obvious one, and the principal reason for applying the method, is that the map used for model building can be more accurate after inclusion of phase information from the partial model. Over the course of iterative model building, the model contains a larger number of atoms and the resulting phase information improves. While this seems likely to be the major contribution to the utility of the method, it may not be the only important factor because the extent of phase improvement is relatively small (on average, an increase in the effective figure of merit of 0.015 over the course of iterations in the eight test cases). A possible additional mechanism whereby a small improvement in the map could lead to a large improvement in the overall completeness of model building is that the inclusion of the refinement step leads iteratively to improved side-chain placement. Side-chain atom placement is dependent on the main-chain atoms in this procedure, as the side chains are identified and placed by superimposing templates for side-chain rotamers on the map using the coordinates of main-chain N, Ca and C atoms. Consequently, it seems possible that part of the large improvement in the quantity of side-chain atoms placed is owing to the refinement of main-chain atomic positions.

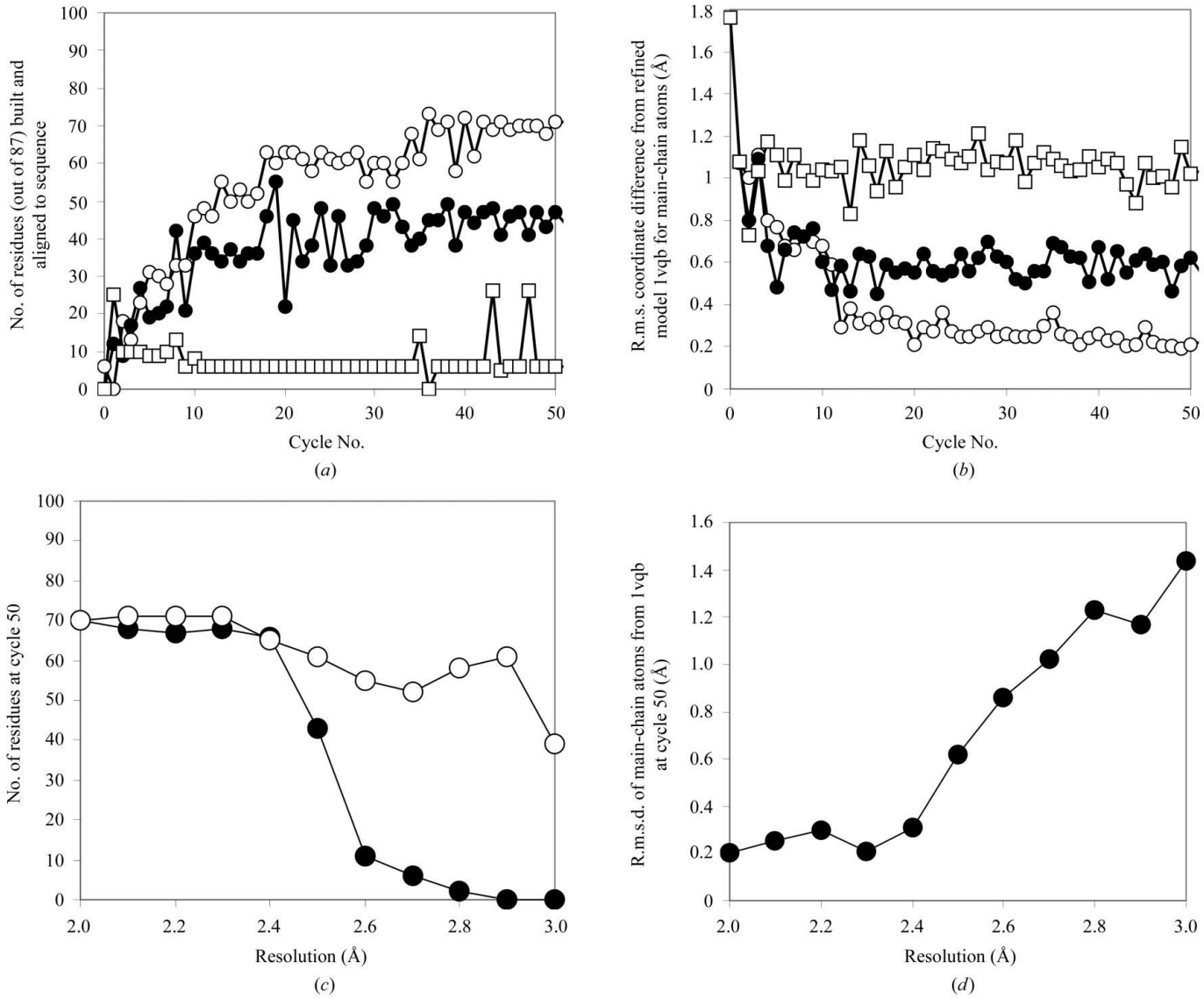


Figure 4
Iterative rebuilding of gene 5 protein beginning with structure 2gn5 and using structure-factor amplitudes corresponding to 1vqb. (*a*) Number of residues built and aligned to sequence as a function of cycle number and resolution of data used (open circles, 2.3 Å; closed circles, 2.5 Å; open squares, 2.7 Å). Gene 5 protein has 87 amino-acid residues; the refined model 1vqb contains 86. (*b*) R.m.s. coordinate difference between refined intermediate models and 1vqb for main-chain atoms (symbols as in *a*). (*c*) Number of residues built in 50 cycles as a function of resolution of data used. Open circles, main chain; closed circles, side chains. (*d*) R.m.s. coordinate difference between refined intermediate models at cycle 50 as a function of the resolution of the data used.

electronic reprint

3.4. Other algorithms for iterative model building at moderate resolution

The procedures described here were carried out with statistical density-modification procedures (Terwilliger, 2000) and with an automatic model-building procedure (Terwilliger, 2003*a*,*b*) based on placing fragments from a library built from refined protein structures. The approach is not specific to these particular methods, however. Other means of phase combination such as σ_A-weighted phase recombination (Read, 2001) and other model-building procedures such as those of Ioerger & Sacchettini (2002), Levitt (2001) or Oldfield (2002) that can function at moderate resolution and procedures that include atomic refinement could also potentially yield improvement with an iterative approach.

3.5. Limitations of the method

The algorithm described here is useful for building a preliminary model, but is not suitable in its current form for fully automatic model building because it does not build a complete model and it does not fully check the model it builds for consistency with known features of macromolecules. At present, only features in its database are recognized; unusual amino acids, ligands, water molecules and nucleic acids are not yet in the databases used. The model-building software performs rudimentary checks for overlap of atomic positions (Terwilliger, 2003*a*) and nearly all the model building is carried out with templates from refined protein structures, but the algorithm does not currently include a systematic check of conformations or van der Waals contacts. An additional limitation is that non-crystallographic symmetry restraints are currently not applied during the refinement process. It is likely that considerably improved models could be obtained by including them. Owing to these limitations, the current algorithm can provide an experienced crystallographer with a very good starting point for final model building and refinement but not with a final model.

4. Conclusions

Iterative model building and phase combination is found to yield considerably more accurate and more complete models than simply building a model into an electron-density map for cases where phase information is available at moderate resolution (<2.8 Å). The use of automated model-building algorithms capable of building models at moderate resolution has therefore extended the range of applicability of iterative model building and refinement (Perrakis *et al.*, 1999) up to about 2.8 Å. The procedures described here have been implemented in version 2.03 of *RESOLVE* and are available from http://solve.lanl.gov.

The author would like to thank Li-Wei Hung for discussion, the NIH for generous support and J. Remington, O. Herzberg, W. Weis and E. Gordon for the use of their data. This work is carried out as part of the PHENIX project (Adams *et al.*, 2002).

References

Adams, P. D., McCoy, A. J., Grosse-Kunstleve, R. W., Hung, L.-W., Ioerger, T. R., Moriarty, N. W., Read, R. J., Sacchettini, J. C., Sauter, N. K. & Terwilliger, T. C. (2002). *Acta Cryst.* D**58**, 1948–1954.
Brayer, G. D. & McPherson, A. (1983). *J. Mol. Biol.* **169**, 565–596.
Chen, C. C. H., Kim, A., Zhang, H., Howard, A. J., Sheldrick, G. M., Dunaway-Mariano, D. & Herzberg, O. (2002). *Biochemistry*, **41**, 13162–13169.
Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* D**50**, 760–763.
Cowtan, K. D. (1998). *Acta Cryst.* D**54**, 750–756.
Fitz-Gibbon, S. T., Ladner, H., Kim, U. J., Stetter, K. O., Simon, M. I. & Miller, J. H. (2002). *Proc. Natl Acad. Sci. USA*, **99**, 984–989.
Folkers, P. J. M., Nilges, M., Folmer, R. H. A., Konings, R. N. H. & Hilbers, C. W. (1994). *J. Mol. Biol.* **236**, 229–246.
Gordon, E. J., Flouret, B., Chantalat, L., van Heijenoort, J., Mengin-Lecreulx, D. & Dideberg, O. (2001). *J. Biol. Chem.* **276**, 10999–11006.
Greer, J. (1985). *Methods Enzymol.* **115**, 206–224.
Huber, A. H., Nelson, W. J. & Weis, W. I. (1997). *Cell*, **90**, 871–882.
Ioerger, T. R. & Sacchettini, J. C. (2002). *Acta Cryst.* D**58**, 2043–2054.
Jones, T. A., Zou, J. Y., Cowan, S. W. & Kjeldgaard, M. (1991). *Acta Cryst.* A**47**, 110–119.
Kleywegt, G. J. (1996). *Acta Cryst.* D**52**, 842–857.
Lamzin, V. S. & Wilson, K. S. (1993). *Acta Cryst.* D**49**, 129–147.
Levitt, D. G. (2001). *Acta Cryst.* D**57**, 1013–1019.
McPherson, A., Jurnak, F. A., Wang, A. H. J., Molineux, I. & Rich, I. (1979). *J. Mol. Biol.* **134**, 379–400.
Morris, R. J., Perrakis, A. & Lamzin, V. S. (2002). *Acta Cryst.* D**58**, 968–975.
Murshudov, G. N., Vagin, A. A. & Dodson, E. J. (1997). *Acta Cryst.* D**53**, 240–255.
Oldfield, T. (2002). *Acta Cryst.* D**58**, 487–493.
Peat, T. S., Newman, J., Waldo, G. S., Berendzen, J. & Terwilliger, T. C. (1998). *Structure*, **6**, 1207–1214.
Pédelacq, J.-D., Piltch, E., Liong, E. E., Berendzen, J., Kim, C.-Y., Rho, B.-S., Park, M. S., Terwilliger, T. C. & Waldo, G. S. (2002). *Nature Biotechnol.* **20**, 927–932.
Perrakis, A., Harkiolaki, M., Wilson, K. S. & Lamzin, V. S. (2001). *Acta Cryst.* D**57**, 1445–1450.
Perrakis, A., Morris, R. M. & Lamzin, V. S. (1999). *Nature Struct. Biol.* **6**, 458–463.
Perrakis, A., Sixma, T. K., Wilson, K. S. & Lamzin, V. S. (1997). *Acta Cryst.* D**53**, 448–455.
Read, R. J. (2001). *Acta Cryst.* D**57**, 1373–1382.
Rossmann, M. G. (1972). *The Molecular Replacement Method.* New York: Gordon & Breach.
Shah, S. A., Shen, B. W. & Brünger, A. T. (1997). *Structure*, **5**, 1067–1075.
Skinner, M. M., Zhang, H., Leschnitzer, D. H., Guan, Y., Bellamy, H., Sweet, R. M., Gray, C. W., Konings, R. N. H., Wang, A. H.-J. & Terwilliger, T. C. (1994). *Proc. Natl Acad. Sci. USA*, **91**, 2071–2075.
Terwilliger, T. C. (2000). *Acta Cryst.* D**56**, 965–972.
Terwilliger, T. C. (2001*a*). *Acta Cryst.* D**57**, 1755–1762.
Terwilliger, T. C. (2001*b*). *Acta Cryst.* D**57**, 1763–1775.
Terwilliger, T. C. (2002*a*). *Acta Cryst.* D**58**, 2213–2215.
Terwilliger, T. C. (2002*b*). *Acta Cryst.* D**58**, 2082–2086.
Terwilliger, T. C. (2003*a*). *Acta Cryst.* D**59**, 38–44.
Terwilliger, T. C. (2003*b*). *Acta Cryst* D**59**, 45–49.
Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**, 849–861.
Vellieux, F. M. D. (1998). *Acta Cryst.* D**54**, 81–85.
Yarbrough, D., Wachter, R. M., Kallio, K., Matz, M. V. & Remington, S. J. (2001). *Proc. Natl Acad. Sci. USA*, **98**, 462–467.

electronic reprint

electronic reprint

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Statistical density modification using local pattern matching

Thomas C. Terwilliger

Copyright © International Union of Crystallography

Author(s) of this paper may load this reprint on their own web site provided that this cover page is retained. Republication of this article or its storage in electronic databases or the like is not permitted without prior permission in writing from the IUCr.

Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Statistical density modification using local pattern matching

Thomas C. Terwilliger

Mail Stop M888, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Correspondence e-mail: terwilliger@lanl.gov

Received 1 May 2003
Accepted 8 July 2003

A method for improving crystallographic phases is presented that is based on the preferential occurrence of certain local patterns of electron density in macromolecular electron-density maps. The method focuses on the relationship between the value of electron density at a point in the map and the pattern of density surrounding this point. Patterns of density that can be superimposed by rotation about the central point are considered equivalent. Standard templates are created from experimental or model electron-density maps by clustering and averaging local patterns of electron density. The clustering is based on correlation coefficients after rotation to maximize the correlation. Experimental or model maps are also used to create histograms relating the value of electron density at the central point to the correlation coefficient of the density surrounding this point with each member of the set of standard patterns. These histograms are then used to estimate the electron density at each point in a new experimental electron-density map using the pattern of electron density at points surrounding that point and the correlation coefficient of this density to each of the set of standard templates, again after rotation to maximize the correlation. The method is strengthened by excluding any information from the point in question from both the templates and the local pattern of density in the calculation. A function based on the origin of the Patterson function is used to remove information about the electron density at the point in question from nearby electron density. This allows an estimation of the electron density at each point in a map, using only information from other points in the process. The resulting estimates of electron density are shown to have errors that are nearly independent of the errors in the original map using model data and templates calculated at a resolution of 2.6 Å. Owing to this independence of errors, information from the new map can be combined in a simple fashion with information from the original map to create an improved map. An iterative phase-improvement process using this approach and other applications of the image-reconstruction method are described and applied to experimental data at resolutions ranging from 2.4 to 2.8 Å.

1. Introduction

Electron-density maps corresponding to macromolecules such as proteins have features that differ in fundamental ways from those found in maps calculated with random phases. These differences have been used in many ways, ranging from improving the accuracy of crystallographic phases to evaluating the quality of electron-density maps. For example, maps corresponding to proteins often have large regions of

© 2003 International Union of Crystallography
Printed in Denmark – all rights reserved

electronic reprint

relatively featureless solvent and large regions containing of polypeptide chains, while a map calculated with random phases has similar fluctuations in density everywhere (Bricogne, 1974). This observation is the basis of the powerful solvent-flattening approach (Bricogne, 1974; Wang, 1985) as well as methods for evaluating the quality of macromolecular electron-density maps (*e.g.* Terwilliger & Berendzen, 1999). Similarly, the presence of non-crystallographic symmetry in macromolecular electron-density maps has been useful in phase improvement (Bricogne, 1974; Rossmann, 1972; Kleywegt & Read, 1997). Additionally, maps corresponding to macromolecules can be interpreted in terms of atomic models, providing a powerful basis for map-quality evaluation and improvement (Agarwal & Isaacs, 1977; Lunin & Urzhumtsev, 1984; Lamzin & Wilson, 1993; Perrakis *et al.*, 1997, 1999, 2001; Morris *et al.*, 2002). On a statistical level, the density in the protein region of a macromolecular electron-density map has a distribution that is very different to that in a map calculated with random phases. This has been extensively used in histogram matching and related methods for phase improvement (Harrison, 1988; Lunin, 1988; Zhang & Main, 1990; Zhang *et al.*, 1997; Goldstein & Zhang, 1998; Nieh & Zhang, 1999; Cowtan, 1999).

In this work, the focus is on local patterns of density that are common in macromolecular protein structures. Macromolecules are built from small regular repeated units and the packing of these units is highly constrained owing to van der Waals interactions. Owing to the regularity of macromolecules on a local scale, their electron-density maps have local features that are distinctive and very different from those of maps calculated from random phases (Lunin, 2000; Urzhumtsev *et al.*, 2000; Main & Wilson, 2000; Wilson & Main, 2000; Colovos *et al.*, 2000). This property has been used to evaluate the quality of electron-density maps and to improve phases at low resolution. Lunin (2000), Urzhumtsev *et al.* (2000), Main & Wilson (2000) and Wilson & Main (2000) use histogram and wavelet analysis to improve electron density in low-resolution maps by requiring the wavelet coefficients to be similar to those of model structures. Colovos *et al.* (2000) analyze the local features of high- and medium-resolution electron-density maps and compare them with those of model maps to evaluate the quality of the maps and suggest that their approaches may also be useful for phase improvement.

We recently developed a method for density modification that consisted of the identification of the locations of helical or other highly regular features in an electron-density map, followed by statistical density modification using an idealized version of this density as the 'expected' electron density nearby (Terwilliger, 2001). This method was shown to yield some phase improvement, but suffered the serious disadvantage that after an initial cycle the features that were initially identified became greatly accentuated and few new features could be found. We suspect that this is a consequence of the inherent feedback in the method, where a feature in the original electron density that partially matches a helical template is restrained to look like this template, making it an even better match for the template in the next round (even if the true density in the region is not helical). We have therefore developed a very different approach to using the information inherent in local features of an electron-density map which does not have this feedback and which therefore might have substantially improved capability for phase improvement.

Here, we show that the local patterns of density surrounding any point in a map can be used to estimate the electron density at that point. This observation makes it possible to begin with an electron-density map with errors, to obtain a new estimate of the density at each point in the map without using the density at that point and thereby to construct a new estimate of electron density that has errors which are nearly uncorrelated with the errors in the original map. This recovered 'image' of the electron density has many uses, including phase improvement and evaluation of map quality.

2. Methods

2.1. Estimation of electron density from local patterns in a map

The central approach of this work is to use the density surrounding each point in a map to construct a new estimate of electron density at that point. There are three overall steps. The first two create templates and evaluate statistics of these templates using data from experimental or model maps, with and without additional errors. The third applies these results to other maps. In the applications described here, we have used density-modified experimental maps obtained from MAD or SAD data at a resolution of 2.6 Å to create the templates and histograms, but a similar procedure could be carried out using either experimental or model maps at any resolution. In the first step, N templates of averaged density are created. These templates were based on the local density in a density-modified experimental protein electron-density map and are grouped by correlation coefficient. Secondly, the relationship between the density at point x and the template which has the highest correlation with the density near x is tabulated using additional density-modified experimental electron-density maps. Finally, the method is applied to other experimental maps. The density near each point x in a map is used to construct a new estimate of the density at x. In this process, the local density is corrected in a way that removes the information about the density at x from all its neighbors.

2.2. Removal of information about density at x from local density

In our approach, the goal is to obtain an estimate of the value of the electron density at a point x in the unit cell in such a way that the new estimate has errors that are not correlated with the errors in the original electron-density map at x. To do this, the method uses information from the electron density at points surrounding the point x in obtaining a new estimate of the value of the electron density at x. One way to remove the information about the electron density at x would simply be to consider the electron density in a spherical shell around the point x. If the inner radius of the shell were large enough, then

electronic reprint

the values of electron density inside the shell would be relatively uncorrelated with the electron density at x. The choice of an inner radius, however, is not obvious because the electron-density map is a Fourier sum of terms with widely varying spatial frequencies. Consequently, there is significant correlation between values of electron density at point x with points even as far away as the resolution of the map. Additionally, it is disadvantageous to exclude all density close to x in the calculations because the patterns to be considered are very local.

An alternative method is to create a local density function for points near x that has values that are similar to the electron density near x, but that are adjusted in such a way that the values are uncorrelated with the electron density at x. This modified local density $g_x(\Delta x)$ will depend on the coordinate difference Δx between each point near x and x. The function $g_x(\Delta x)$ is a function of both x and Δx and therefore must be calculated separately for each point x and offset Δx in the map. We would like the value of the function $g_x(\Delta x)$ to be generally similar to the value of the electron density at $x + \Delta x$, which we will represent by $\rho(x + \Delta x)$. As Δx is increased, we would like $g_x(\Delta x)$ to become very close to $\rho(x + \Delta x)$. That is, we would like

$$g_x(\Delta x) \simeq \rho(x + \Delta x), \tag{1}$$

$$g_x(\Delta x) \rightarrow \rho(x + \Delta x) \text{ for large } \Delta x. \tag{2}$$

We would also like the function $g_x(\Delta x)$ to be uncorrelated everywhere with the value of the electron density at x, given by $\rho(x)$. The function $g_x(\Delta x)$ gives modified values of the density at $x + \Delta x$. We would like to be able to say that if we compare the modified density at $x + \Delta x$ [given by $g_x(\Delta x)$] with the density at x [given by $\rho(x)$], these quantities should be unrelated [that is, $g_x(\Delta x)$ does not contain information about the value of $\rho(x)$]. One way to specify this is to require that for any offset Δx, if we go through the entire map and calculate $g_x(\Delta x)$ for each point x, then $g_x(\Delta x)$ and $\rho(x)$ are to be uncorrelated,

$$\langle g_x(\Delta x)\rho(x)\rangle_x = 0 \ \forall \ \Delta x. \tag{3}$$

A final desirable property of $g_x(\Delta x)$ for the current purpose is to have its value at $\Delta x = 0$ be equal to the mean value of $g_x(\Delta x)$ for nearby points Δx. The reason this is desirable is that we would like to compare local patterns to a template based on the correlation of densities and have no contribution from the mean value of local density. Setting the value of $g_x(\Delta x)$ to any fixed value (*e.g.* zero) at $\Delta x = 0$ would introduce a contribution that comes from the mean value of local density $\rho(x)$ to the correlation between $g_x(\Delta x)$ and a template. A way to remove information about the mean value of local density is to specify the requirement that

$$g_x(\Delta x = 0) = \langle g_x(\Delta x)\rangle_{\Delta x}, \tag{4}$$

where all values of Δx in the region to be used later in calculations of correlations of densities are considered in the averaging.

A function $g_x(\Delta x)$ that has all these properties is

$$g_x(\Delta x) = \rho(x + \Delta x) - [\rho(x) - \langle\rho(x + \Delta x)\rangle_{\Delta x}]W(\Delta x), \tag{5}$$

where the weighting function $W(\Delta x)$ is given by

$$W(\Delta x) = U(\Delta x)/[1 - \langle U(\Delta x)\rangle_{\Delta x}], \tag{6}$$

and where the function $U(\Delta x)$ is the normalized value of the Patterson function near the origin, calculated from the electron-density map itself using the relation

$$U(\Delta x) = \langle\rho(x)\rho(x + \Delta x)\rangle_x/\langle\rho^2(x)\rangle_x. \tag{7}$$

In essence, $g_x(\Delta x)$ is equal to the value of the electron density at $x + \Delta x$, after correction for the difference between $\rho(x)$, the value of the electron density at x, and $\langle\rho(x + \Delta x)\rangle_{\Delta x}$, the mean of nearby values, all using the weighting function $W(\Delta x)$. It can be verified by substitution that both (3) or (4) are satisfied by this function. Additionally, (1) and (2) are satisfied because the normalized rotationally averaged Patterson function is normally quite small everywhere except near the origin and normally becomes very small for points far from the origin.

2.3. Local pattern identification

The first step in the procedure for density modification by pattern matching is to obtain templates that correspond to common patterns of local electron density. These templates are generated using the local electron density near each point x in density-modified experimental electron-density maps, modified to remove information from the central point x, as described in the previous section. The maps can be calculated at any resolution, but a set of templates is normally associated with a particular resolution (typically $d_{\min}$ = 2.6 Å). The approach used here to obtain templates is hierarchical. First, three separate sets of $N_{\max}$ (typically 40) templates are generated using only points in an electron-density map that have low, medium or high electron density. A subset (typically 40) of these templates that have low mutual correlation is then selected. Finally, an even smaller subset of $N_{\rm final}$ (typically 20) templates is chosen from this group in order to maximize the predictive power of the templates while maintaining a fixed number of total templates.

To generate a set of templates, each grid point in an electron-density map is considered, one at a time, only including points that are associated with either low ($\rho < \overline{\rho} - 0.8\sigma$), medium ($\overline{\rho} - 0.2\sigma < \rho < \overline{\rho} + 0.2\sigma$) or high electron density ($\overline{\rho} + 1.5\sigma < \rho$), where $\overline{\rho}$ and σ are the mean and standard deviation of the map, depending on the set of templates to be created. For each appropriate grid point (x), the modified local electron density $g_x(\Delta x)$ is calculated for all neighboring points within a radius $r_{\max}$ (typically, $r_{\max}$ = 2 Å when $d_{\min}$ = 2.6 Å). This modified electron density is compared with all existing templates using the correlation coefficient of density in the template with the modified local density as a measure of similarity. The grid used is normally the same grid as is used for all FFT, NCS-averaging and other density calculations and is typically between 1/6 and 1/4 of the resolution of the map. The number of points typically used in a template is approximately 100. For each existing template, $N_{\rm rot}$ different rotations of the template are considered so as to

electronic reprint

attempt to match the modified local density in any orientation and the highest correlation coefficient of the match for all rotations of the template is noted. In the examples considered here, we use a total of $N_{\rm rot}$ = 158 rotations to sample the possible three-dimensional rotations of an object with a rotation of about 50° relating neighboring orientations. If the correlation coefficient of the local modified electron density at this point x with an existing template k is greater than $\mathrm{CC}_{\rm min}$ (typically, $\mathrm{CC}_{\rm min}$ = 0.85), then the local modified density at this point is included in the definition of template k by rotating the density to match the current template k and including the rotated local modified density in the average density for this template. If the local modified electron density does not have a correlation with any existing template greater than $\mathrm{CC}_{\rm min}$, then the local modified density is used to start a new template. Once $N_{\rm max}$ templates have been created (typically, $N_{\rm max}$ = 40), then the local modified density at each subsequent point is included in whichever template it matches most closely.

By repeating the generation of templates using points in the electron-density map that have low, medium and high density, a relatively diverse set of templates is created. Next, a subset (typically 1/3) of these is chosen based on mutual correlation coefficients in order to obtain a set of templates with the minimum possible similarity to each other. To do this, the correlation coefficients of all pairs of templates are calculated and the template with the highest correlation to another template is eliminated. The process is repeated until the desired number of templates is obtained. The final selection of templates based on predictive power is carried out after analyzing the statistics associated with each of the $N_{\rm max}$ templates obtained at this stage, as described in a later section.

2.4. Statistics of local patterns: general approach

The second overall step in this process is to identify the relationships between the correlation of each template with local modified density in a map and the value of the electron density at x. This is peformed for experimental maps both with and without added errors. There are many possible ways to describe these relationships, but a simple approach used here is to break it down into two parts.

The first part consists of an examination of the statistics of high-quality experimental maps. We have found that the electron density at a point x in a map is quite strongly dependent on the two templates k and l that have the highest (k) and next-highest (l) correlation coefficients with the local modified density at x. That is, for electron-density maps of proteins, the probability distribution $p(\rho|k, l)$ can be very informative about the electron density ρ at x.

The second part is to consider the relationship between maps with and without added errors. The approach is to begin with the observed correlation coefficients of all the templates at a point x to a map that contains errors and then to use these in a calculation of the probability that a particular pair of templates k and l would have the highest two correlation coefficients in the corresponding high-quality map. In this case, the statistics of density for the high-quality maps $p(\rho|k, l)$ obtained above can then be applied.

To carry this process out, a second set of probabilities are needed. These are the probabilities $p(\mathrm{CC}_k|\mathrm{CC}_{{\rm obs},k})$ that the correlation coefficient for template k to a point x in a high-quality map would have the value CC_k, given the observation that this template has a correlation coefficient of $\mathrm{CC}_{{\rm obs},k}$ to the same point in a map with additional errors. To account for differing levels of error in the experimental map, these probabilities are tabulated as a function of the overall figure of merit of the map with errors.

To apply these probability distributions to data near the point x in a new ('observed') electron-density map, the correlation coefficient of each template k to the local modified density near x is first determined (once again, after trying many rotations and choosing the one for each template that maximizes the correlation coefficient). This set of correlation coefficients $\{\mathrm{CC}_{\rm obs}\}$ and the two probability distributions $p(\rho|k, l)$ and $p(\mathrm{CC}_k\mathrm{CC}_{{\rm obs},k})$ can then be combined as follows to obtain an estimate of the electron density ρ at x in a high-quality version of the same map.

If we somehow knew which two templates k and l have the highest correlation coefficients to the local modified density near x in a high-quality version of the new 'observed' map, then we could use our probability distribution $p(\rho|k, l)$ directly to estimate the probability distribution for ρ. We do not know the identity of k and l, but suppose instead that we had probabilities, $p(k, l|\{\mathrm{CC}_{\rm obs}\})$, for each possible pair k and l based on the correlation coefficients observed for the 'observed' map. Combining these, we could write that

$$p(\rho|\{\mathrm{CC}_{\rm obs}\}) = \sum p(\rho|k, l)p(k, l|\{\mathrm{CC}_{\rm obs}\}), \qquad (8)$$

where the sum is over all possible pairs of templates k and l. An estimate of the electron density at x can then be obtained from the weighted mean

$$\rho_{\rm est} = \int \rho\, p(\rho|\{\mathrm{CC}_{\rm obs}\})\, \mathrm{d}\rho. \qquad (9)$$

The probability, $p(k, l|\{\mathrm{CC}_{\rm obs}\})$, that the pair k and l have the highest correlation coefficients to the local modified density near x in a high-quality version of the 'observed' map can in turn be estimated from the observed correlation coefficients of all the templates to this map, $\{\mathrm{CC}_{\rm obs}\}$, in several steps. We separate the probability into two parts, one for the probability that template k has the highest correlation and one for the probability that template l has the next highest, given that template k has the highest correlation,

$$p(k, l|\{\mathrm{CC}_{\rm obs}\}) = p(l|k, \{\mathrm{CC}_{\rm obs}\})p(k|\{\mathrm{CC}_{\rm obs}\}). \qquad (10)$$

We can now estimate the probability that template k has the highest correlation with the (non-existent) high-quality version of the 'observed' map. We will integrate over all possible values of CC_k, the correlation of template k with the high-quality map. For each value of CC_k, we will calculate the probability that this is indeed the value of the correlation of template k, given by $p(\mathrm{CC}_k) = p(\mathrm{CC}_k|\mathrm{CC}_{{\rm obs},k})$, and the probability that all other templates have a correlation coefficient less than CC_k,

electronic reprint

$$p(k|\{\mathrm{CC_{obs}}\}) = \int p(\mathrm{CC}_k) \prod_{j \neq k} p(\mathrm{CC}_j < \mathrm{CC}_k)\, \mathrm{dCC}_k, \quad (11)$$

where the integral is over all values of CC_k. The probability that template l has the next-highest correlation is given by

$$p(l|\{k, \mathrm{CC_{obs}}\}) = \int p(\mathrm{CC}_l) \prod_{j \neq k,l} p(\mathrm{CC}_j < \mathrm{CC}_l)\, \mathrm{dCC}_l. \quad (12)$$

2.5. Statistics of local patterns: tabulating histograms

An important part of this step consists of generating histograms of values for the electron density at x as a function of the correlation coefficients of the $N_{\max}$ templates with the local modified density at x. Each of the $N_{\max}$ templates is compared with the modified local density at all points in a set of high-quality maps. At each point x, the two templates k and l that have the highest and next-highest correlation coefficients, respectively, with the local modified density at x are identified (after rotation to maximize this value). The value of the (unmodified) electron density $\rho(x)$ is then tabulated as a function of k and l. These histograms are then normalized to yield an estimate of the probability distribution, $p(\rho|k, l)$.

The second part of this step is to obtain probability distributions, $p(\mathrm{CC}_k|\mathrm{CC}_{\mathrm{obs},k})$, relating the correlation coefficient value, $\mathrm{CC}_{\mathrm{obs},k}$, observed for a particular template at a point x in a map that contains added errors to the correlation coefficient, CC_k, that would be observed for the identical template at the identical point x in the corresponding map without any added errors. These probability distributions are calculated by using paired sets of high-quality experimental maps with and without added errors. At each point in a map, the correlation coefficient of each template k to the map without added errors, CC_k, and the correlation to the map with added errors, $\mathrm{CC}_{\mathrm{obs},k}$, are noted. Normalization of the resulting histograms leads to an estimate of the probability, $p(\mathrm{CC}_k|\mathrm{CC}_{\mathrm{obs},k})$, that CC_k is the correlation to the map without added errors if the value $\mathrm{CC}_{\mathrm{obs},k}$ is observed in the map with added errors. This calculation is repeated for maps with varying levels of additional errors by creating simulated phase sets with Gaussian distributions of phase errors with varying overall values of the cosine of phase error, $\langle\cos\Delta\varphi\rangle$, ranging typically from 0.5 to 0.8. In application to new 'observed' map, the probability distribution obtained using data with added phase errors with a mean cosine $\langle\cos\Delta\varphi\rangle$ similar to the figure of merit of the experimental map is used.

2.6. Selection of templates based on predictive power

The final selection of N_{final} templates is based on predictive power. A subset of N_{final} templates is selected from the $N_{\max}$ templates obtained earlier using high-quality electron-density maps. The subset is selected to maximize the correlation between the electron density calculated using (9) and the electron density in the maps. The histograms that form the basis of (9) are calculated from experimental density for one set of proteins and the correlation is calculated for another. The pair of templates that yields the highest correlation is first identified. Then, one by one, the template that increases this correlation by the largest amount is added to the group, until N_{final} templates are chosen.

2.7. Indexing the rotations for each template to reduce computational requirements

The slowest step in applying the procedures described here consists of calculating the maximum correlation of local modified density with each of the N_{final} templates, considering as many as 158 rotations of each template (or local density) for each point. We have developed a simple indexing system that reduces the number of rotations that need to be considered for each template. The index for a point x is based on the density at M points near x (typically, $M = 9$ and the points are chosen to be approximately uniformly distributed on a sphere of radius $0.9r_{\max}$ centered at x). Point m is given an local index i_m from 0 to 3, based on the local density at that point ($\rho \leq -\sigma$, $-\sigma < \rho \leq 0$, $0 < \rho \leq \sigma$ or $\rho > \sigma$), where σ is the r.m.s. of the entire map. An overall index I is then calculated for the local density from the relation

$$I = \sum i_m 4^{(m-1)},$$

where the sum is over the M nearby points. Next, the relationship between the index I and the best rotation is tabulated for each of the templates using high-quality experimental maps containing added errors. For each point in each map used above to calculate statistics of the correlation of templates with local modified density, the index I is calculated and the optimal rotation is noted for each template. An indexing table is then constructed in which each index I is associated with a list of preferred rotations for each template. The table is constructed so that about 95% of the time the optimal rotation for a given template is contained in the list. This indexing procedure reduces the number of rotations that need to be considered by about a factor of five. Other indexing methods could be applied that might further reduce the number of rotations to be considered (*e.g.* Funkhouser *et al.*, 2003).

2.8. Using local patterns to create a new estimate of electron density

The pattern of density near a point x in an electron-density map can be analyzed using (8) to produce a probability distribution, $p(\rho|\{\mathrm{CC_{obs}}\})$, for the electron density at x. The estimate from (9) of density at x, ρ_{est} (and the uncertainty in this estimate, σ_{est}, if desired), can then be used to construct a new estimate of the electron density in the map. This 'recovered image' of the electron-density map can be visualized with or without smoothing, can be used as a target for statistical density modification (Terwilliger, 2000) or can be combined directly with the original electron-density map to obtain an improved map.

We have used an iterative procedure to combine the information from the recovered image with the information present in an experimental electron density (Fig. 1). In the first cycle, the starting phase probabilities are experimental values

electronic reprint

and in all cycles the amplitudes are experimental values. In each cycle, the starting phases and amplitudes are subjected to density modification (*e.g.* statistical density modification or other related methods) to obtain the best possible electron-density map without using any pattern-based information. This density-modified map is then analyzed for local patterns and an image of the map is recovered. Thirdly, the density in the recovered image is used all by itself to estimate phase probabilities. This third step is carried out here using statistical density modification (Terwilliger, 2000) as described below, but could be performed using σ_A-based methods (Read, 1986). Finally, the phase probabilities from the recovered image are combined with the original experimental phase probabilities to yield the starting phase probabilities for the next cycle. The process is iterated until changes in the density-modified map from cycle to cycle are small (typically one to five cycles). The density-modified map from the final cycle is then suitable for interpretation.

2.9. Using statistical density modification to estimate phases based on a target electron-density function

Statistical density modification (Terwilliger, 2000) is a procedure for calculating crystallographic phase probabilities based on the agreement of the map resulting from these phases with prior expectations. Any set of prior expectations about the map can be included in this procedure. In particular, if an estimate of electron density is available for all points in the map (*e.g.* the recovered image obtained in the procedure described above), then this estimate can be used as prior information about the map. In this procedure, observed values of the amplitudes of structure factors are used and an estimate of uncertainty in the electron density is required. This procedure is used to estimate phase probabilities from a recovered image, where the expected electron density is simply the best estimate from (9) and the uncertainty is taken to be a constant everywhere given by the r.m.s. of a map calculated with the observed structure-factor amplitudes.

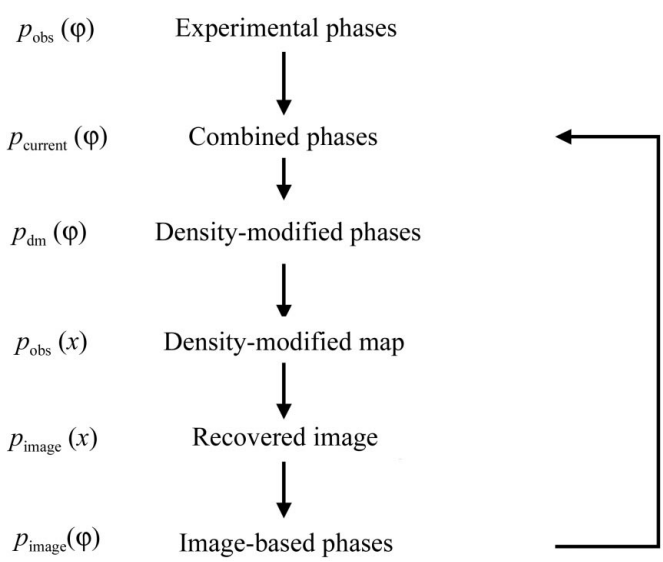


Figure 1
Outline of procedure for density modification using local patterns.

3. Results and discussion

3.1. Removing information about electron density at *x* from the local electron density

An important aspect of the pattern-matching density-modification method presented here is that it is designed to yield an estimate of the electron density that has errors uncorrelated with the errors in the original map. This is accomplished by using only information from the region around a point x to estimate the density at x and not including any information about the density at x in the process, as described in §2. Fig. 2 illustrates this process of removing information about electron density at x. Fig. 2(*a*) shows a section of a density-modified MAD electron-density map for initiation factor 5A (IF5A; Peat *et al.*, 1998) in the region near a particular point x (the point x is designated by a star at the center of the figure). Note that the density at x is positive in this case. In Fig. 2(*b*), the density is adjusted to remove the information about the density at x from x and from all neighboring points. This calculation essentially consists of subtracting the origin of a normalized Patterson function corresponding to this map, multiplied by the value of the density at x minus the mean local density, from all neighboring points, as described in §2. This calculation has the effect of setting the value of the density at x to the mean density in the local region, setting the density very near x to intermediate values and leaving the value of points far from x unchanged.

3.2. Common local patterns in protein electron-density maps

The analysis of local patterns in electron-density maps was carried out using the density-modified MAD electron-density map from IF5A, calculated at a resolution of 2.6 Å (PDB code 1bkb; Berman *et al.*, 2000; Peat *et al.*, 1998). This was a very

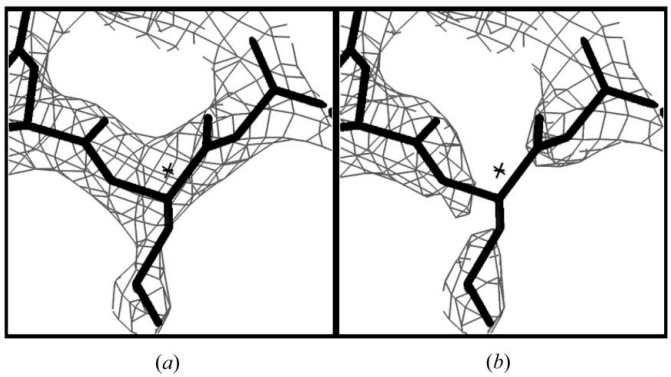


Figure 2
Creating the local modified density function $g_x(\Delta x)$. (*a*) Density in the IF5A electron-density map is shown with contours at 1.5σ. The atomic model used to calculate the map is shown and the central point ('x') is marked with an asterisk. (*b*) Modified local density $g_x(\Delta x)$ calculated using (5) corresponding to the map in (*a*) is shown. All electron-density maps were created with *MAPMAN* (Kleywegt & Jones, 1996) and *O* version 8.0 (Jones *et al.*, 1991).

electronic reprint

clear map with a correlation coefficient to the map calculated from the final refined model of IF5A of 0.82. Local patterns were analyzed for regions centered on each point in this grid, only considering points within 2.5 Å of an atom in the model. Local patterns were identified as described in §2 using the modified local density surrounding each point. This approach removes information about the density at x from the nearby density. The patterns are selected after considering rotations about the central point, so any rotational differences between templates are not significant in determining their features.

The final templates were chosen on the basis of their predictive power. The N_{max} = 40 templates that were initially created using the model electron-density map for IF5A were then compared with all points in two other density-modified experimental electron-density maps, the armadillo repeat of β-catenin (Huber *et al.*, 1997) and red fluorescent protein (Yarbrough *et al.*, 2001), and correlation coefficients for each template at each point were obtained. The same 40 templates were then compared in the same way with the IF5A map. Finally, subsets of the 40 templates were considered. For each subset of templates, the β-catenin and red fluorescent protein electron-density maps were used to generate histograms and the IF5A map was used to compare the estimates of electron density obtained using (9) with IF5A electron density. In the first cycle of identifying templates, all pairs of templates were considered and the pair yielding the highest correlation was chosen. In subsequent cycles, the additional template that yielded the greatest improvement in correlation was chosen. Fig. 3(*a*) (open circles) shows the correlation of estimated and model density as a function of the number of templates used. Much of the information is contained in just two templates and almost all the rest is in the first 20. Based on this observation, we have used 20 templates for the remainder of this work.

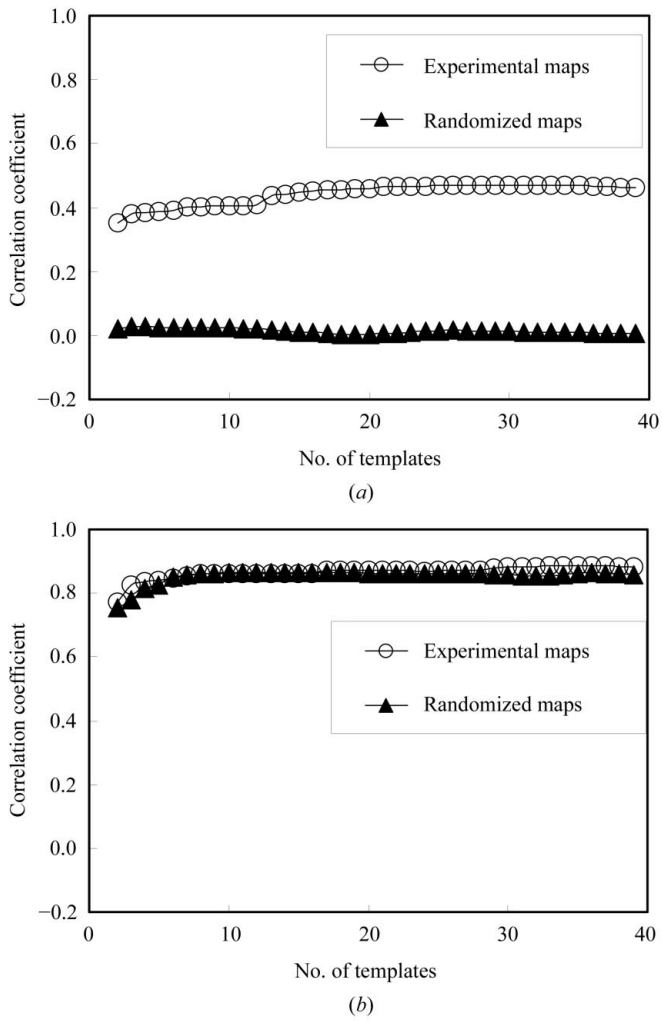


Figure 3
Predictive power of templates. (*a*) Correlation of the recovered density function with true density for the IF5A map (open circles) and for the randomized IF5A map (closed triangles). The correlation of ρ_{est} calculated from (9) with model density ρ is plotted as a function of the number of templates used. For the open circles, the templates were derived from the IF5A map, the histograms from β-catenin and red fluorescent protein maps and the model density and recovered density were from the IF5A map. For the closed triangles, phases were randomized for all three maps before carrying out the calculations. (*b*) As in (*a*), except that the local density was not adjusted to remove information about the density at the central point, so that $g_x(\Delta x) = \rho(x + \Delta x)$.

The fundamental property of macromolecular electron-density maps that is used in our approach is that different local patterns of density in these maps are associated with different values of the density at their central point. The open circles in Fig. 3(*a*) show that such an association exists and that only a small number of templates are needed to describe it. We next tested whether a similar association exists for random maps. The closed triangles in Fig. 3(*a*) were obtained in the same way the open circles, except that all the maps were calculated after randomizing all the crystallographic phases. The closed triangles in Fig. 3(*a*) show that there is essentially no association between local patterns of density and density at their central points for the random maps. This means that the correlations between patterns and densities at their central points is a feature of protein-like maps and not a feature of maps with random phases.

An important part of the present approach was the removal of information about the density at a point x in the analysis of the patterns surrounding x using (5). The reason for doing this was to obtain an estimate of the density at point x that is independent of the current value of density at that point. Fig. 3(*b*) shows that this choice of methods is also important for discriminating between patterns that arise from noise and those that arise from protein-like features. Fig. 3(*b*) was calculated in exactly the same way as Fig. 3(*a*), except that the local density was not adjusted to remove information about the value of the density at the central point and a completely new set of templates and statistics was used, reflecting this different approach. This was accomplished by not applying (5) to the local density. The open circles in Fig. 3(*b*) show that if the local density is not adjusted to remove information about the central point, then templates can be obtained that give a very high correlation between the value of the density calculated from (9) and the actual density. However, this correlation is likely to be almost entirely due to the fact that information about the central point is included in both the

electronic reprint

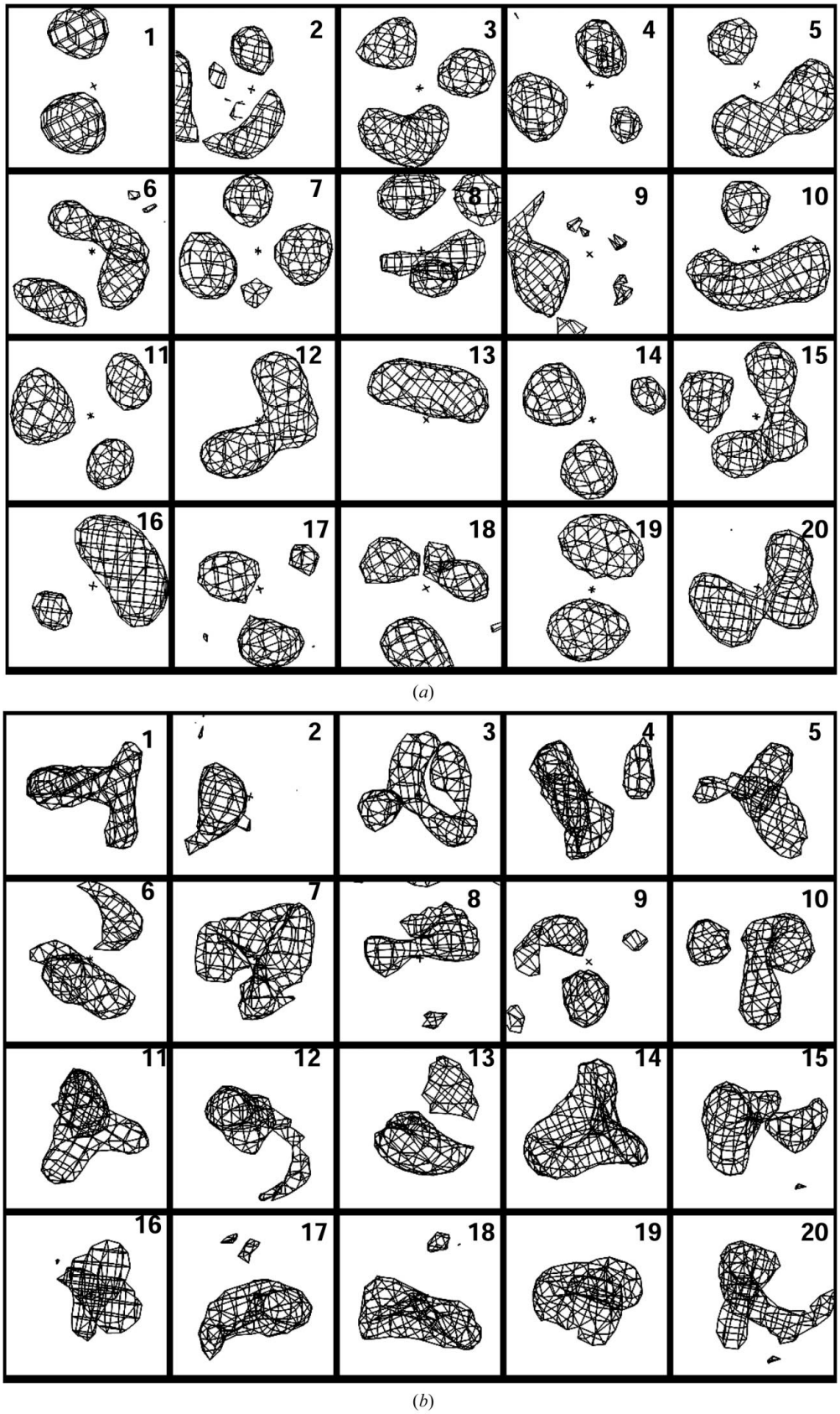


Figure 4
Templates of local density calculated at a resolution of 2.6 Å. The templates are arranged in order of decreasing contribution to the information about the density at the central point. The sections shown are 8 × 8 Å; only the spherical region 4 Å in diameter at the center of each figure is used in the pattern-matching process. Contours at $+1.5\sigma$ (*a*) and -1.5σ (*b*, templates in the same orientation as in *a*) are shown.

templates and the correlations. Supporting this interpretation, the closed triangles in Fig. 3(*b*) show that randomized maps give essentially the same correlations as protein electron-density maps when the information about the central point is not removed from the calculations.

Figs. 4(*a*) and 4(*b*) show contours of positive density corresponding to the $N_{max} = 20$ templates obtained. The templates are arranged in order of decreasing contribution to the estimates of density. The patterns are very simple, typically containing one to three spherical or extended regions of positive density and one or more rings or regions of negative density in various relations to the central point. Some of the pairs of templates are similar (for example, Nos. 17 and 18) and, as shown in Fig. 3, the number could be reduced further with just a small reduction in predictive power. The patterns found in some of the templates are related in a simple way to atomic coordinates in the structures used to generate the templates. For example, Fig. 2 shows the density surrounding a point located near a C^{α} atom, the junction of three chains of atoms. This density, after removing the information about the density right at this point, is most closely similar to pattern No. 12 in Fig. 3, which consists of a curved lobe of density adjacent to the origin.

The core of the method described here is the association of different templates with different expected values of electron density at the point that is at the center of the templates. The electron density near a point x in a map is compared with the 20 templates and the two templates that match the density most closely

electronic reprint

Table 1
Templates of local electron density calculated at a resolution of 2.6 Å.

Template	Mean density at center (arbitrary units, with mean of map equal to zero)	Variance of mean density
1	−0.29	0.60
2	0.06	0.73
3	−0.63	0.59
4	−0.55	0.60
5	−0.38	0.81
6	0.49	0.95
7	−0.68	0.56
8	−0.05	0.72
9	−0.40	0.55
10	−0.32	0.70
11	−0.41	0.74
12	0.62	0.87
13	0.37	0.72
14	−0.46	0.66
15	0.46	1.00
16	−0.17	0.76
17	−0.03	0.78
18	−0.15	0.66
19	−0.27	0.81
20	0.49	1.00

are identified. The procedure is first performed with high-quality experimental maps to associate pairs of templates with expected density and then with an observed map to estimate the values of electron density in a high-quality version of the observed map. In order to use as much information as possible, the process is carried out in a probabilistic fashion, considering the possibility that any pair of patterns might best match the density in a high-quality version of the observed map.

The 20 patterns are each associated with different average values of density at their central points. For example, template No. 1 contains two spherical regions of positive density situated on opposite sides of the origin. At locations where this pattern is the one that best matches the density in model maps, the mean density at the central point is about -0.3 ± 0.6 (on an arbitrary scale with the mean of the map equal to zero). Template No. 12 contains a curved lobe of positive density immediately adjacent to the origin. Template No. 12 is associated with mean density of about 0.6 ± 0.9. Table 1 lists the density associated with locations where each of the 20 templates best match the local modified density in model maps.

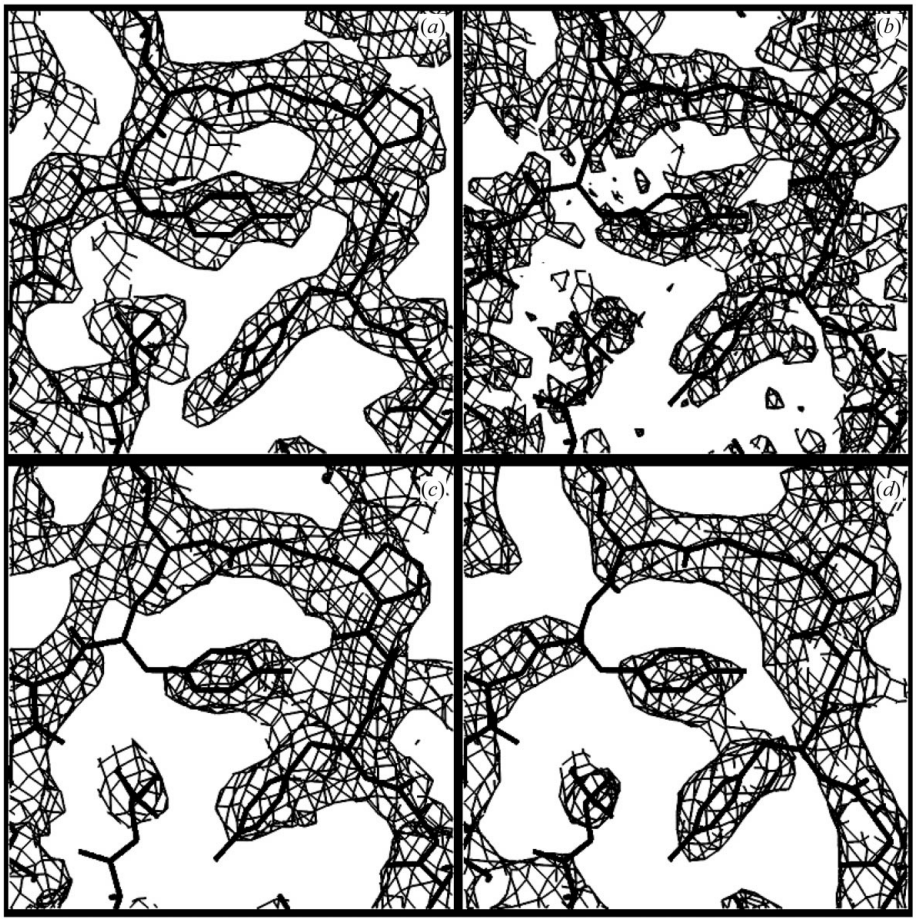


Figure 5
Template matching using model electron density with errors based on the structure of gene 5 protein at a resolution of 2.6 Å. (*a*) Model map with Gaussian phase errors adjusted to yield a correlation to the perfect map of 0.81. (*b*) Estimated electron density reconstructed from the map in (*a*). (*c*) Density in (*b*) after smoothing with a spherical smoothing function with a radius of 1.5 Å. (*d*) Map calculated with model structure-factor amplitudes and with phases estimated using statistical density modification based on the reconstructed density in (*c*). All contours are at 0.8σ.

3.3. Reconstructing model electron density using correlations with local patterns

The templates shown in Fig. 4 and the density typically associated with them listed in Table 1 can be used to reconstruct an image of an electron-density map. Fig. 5 shows an example using model data so that errors can be readily analyzed. Fig. 5(*a*) shows a section of model electron density with errors calculated using the structure of gene 5 protein (PDB code 1vqb; Skinner *et al.*, 1994) at a resolution of 2.6 Å. The errors in the phases were adjusted so that the map had a correlation coefficient to the perfect map of 0.81. The estimated electron density reconstructed from this map is shown in Fig. 5(*b*) and a version of this density, smoothed with a radius of 1.5 Å, is shown in Fig. 5(*c*). Finally, phases were estimated using statistical density modification based on the model structure-factor amplitudes from the reconstructed density (Fig. 5*d*). The reconstructed density has a correlation coefficient to the original (model) map of 0.19, the smoothed image has a correlation of 0.38 and the map calculated with phases obtained from the reconstructed density and model amplitudes has a correlation coefficient of 0.46.

electronic reprint

As model data were used to obtain the images in Fig. 5, it is possible to analyze the errors in the recovered image and determine whether they are in fact independent of the errors in the original map. The errors in electron-density maps are somewhat complicated as they come from errors in phase angles. A simplified error model in which the values of the electron density in two maps $y_1(x)$ and $y_2(x)$ have correlated errors is assumed for the present analysis. For convenience, in this analysis the maps $y_1(x)$, $y_2(x)$ are each normalized to an r.m.s. value of unity and a mean of zero. In this error model, each map has a component that is related to $t(x)$, the true density in a perfect map (also normalized in the same way), each map has a component $c(x)$ that is an error term unrelated to $t(x)$ but that is the same in the two maps and each map has an independent error term $e_1(x)$ and $e_2(x)$. As this is model data, we know the values of $t(x)$ as well as the values of $y_1(x)$ and $y_2(x)$,

$$y_1(x) = \alpha_1 t(x) + c(x) + e_1(x), \tag{13}$$

$$y_2(x) = \alpha_2 t(x) + c(x) + e_2(x). \tag{14}$$

In this model case, the coefficients α_1 and α_1 can be estimated from the known maps $t(x)$, $y_1(x)$ and $y_2(x)$,

$$\alpha_1 \simeq \langle y_1(x) t(x) \rangle, \tag{15}$$

$$\alpha_2 \simeq \langle y_2(x) t(x) \rangle. \tag{16}$$

We can then estimate the correlation of errors $\mathrm{CC}_{\mathrm{errors}}$ with the relation

$$\mathrm{CC}_{\mathrm{errors}} \simeq \frac{\langle [y_1(x) - \alpha_1 t(x)][y_2(x) - \alpha_2 t(x)] \rangle}{\{\langle [y_1(x) - \alpha_1 t(x)]^2 \rangle \langle [y_2(x) - \alpha_2 t(x)]^2 \rangle\}^{1/2}}. \tag{17}$$

Using (17), we find that the correlation coefficient of the errors in the starting map with errors with the errors in the recovered map in Fig. 5(*b*) is −0.01. The same calculation for the recovered smoothed map in Fig. 5(*c*) leads to a correlation coefficient of the errors of −0.02. Similarly, the calculation for the map in Fig. 5(*d*) obtained using phases calculated from the recovered image and model amplitudes lead to a correlation of errors of −0.04. This indicates that the errors in the recovered image are not correlated with the errors in the original map.

We have found that the independence of errors is not as perfect when density-modified phases are used. To examine this, we started with model phases and amplitudes, introduced errors into the phases, leading to an electron-density map with a correlation to the perfect map of 0.6, and then carried out statistical density modification on this map (not including any local pattern information), leading to a density-modified map with a correlation to the perfect map of 0.83. This density-modified map was then analyzed for local patterns as described above. In this case the smoothed recovered image had a correlation to the perfect map of 0.50. The correlation of errors with the density-modified map was 0.21, considerably higher than in the case where the map used for pattern identification had completely random errors. This suggests that the method might not be quite as effective when used on density-modified maps as on experimental maps.

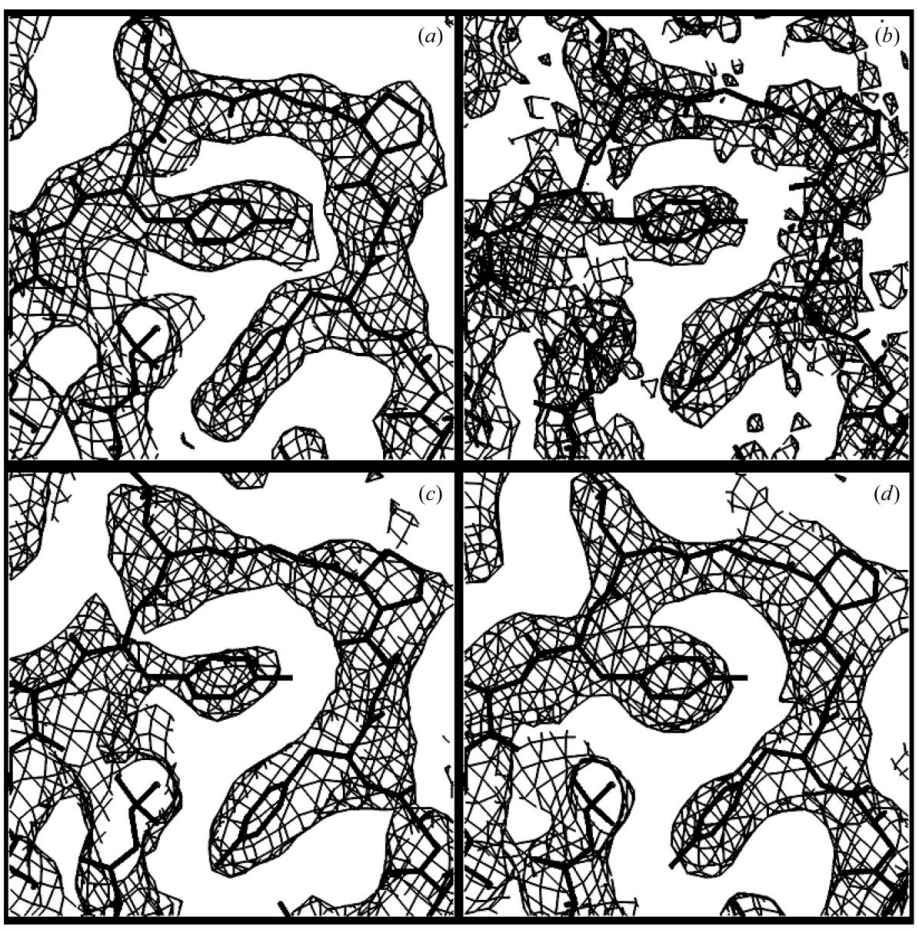


Figure 6
Template-matching using gene 5 protein MAD data. As in Fig. 5, but using experimental MAD data instead of model data.

3.4. Reconstructing electron density from density-modified experimental maps using correlations with local patterns

The analysis described above was carried out with electron density calculated from models so that the error analysis could be performed in detail. We next applied the method to electron density obtained from a MAD experiment so that its utility with real data could be examined. The electron density obtained after applying statistical density modification (Terwilliger, 2000) to three-wavelength MAD data from gene 5 protein (PDB code 1vqb;

electronic reprint

Skinner *et al.*, 1994) was used as the starting point for this analysis. This *RESOLVE* electron-density map had a correlation coefficient of 0.79 to the model density calculated from PDB entry 1vqb. Fig. 6(*a*) shows a section through this density-modified map. Local pattern analysis was applied to this map as described above. Fig. 6(*b*) shows the image that was recovered from this map, Fig. 6(*c*) shows a smoothed version of this image and Fig. 6(*d*) shows the map obtained using phases calculated from the recovered image and observed structure-factor amplitudes. The recovered image in Fig. 6(*b*) has a correlation of 0.25, the smoothed recovered image in Fig. 6(*c*) has a correlation of 0.42 and the map calculated using phases from the recovered image in Fig. 6(*d*) has a correlation of 0.52.

An approximate version of the error analysis described in the previous section for Fig. 4 was carried out for the maps in Fig. 6. In this analysis, the 'true' density was taken to be the density calculated from the model of gene 5 protein (PDB code 1vqb). The correlation of errors between the starting *RESOLVE* map in Fig. 6(*a*) with the errors in the recovered image in Fig. 6(*b*) was 0.15 and the correlation of errors between the starting *RESOLVE* map with the errors in the smoothed recovered image in Fig. 6(*c*) was 0.23. The correlation of errors in the map calculated using phases from the recovered image in Fig. 6(*d*) with the errors in the starting *RESOLVE* map was 0.36. This means that the errors are not highly correlated in this analysis, but that they are also not completely independent. Part of the correlation of 'errors' could be because of the fact that the 'true' density is not known and the errors are estimated using model density for gene 5 protein. Consequently, any errors in this model density would lead to correlation of 'errors' in all the maps in this analysis.

3.5. Combination of phase information from local pattern identification with experimental phase information

Fig. 6(*d*) shows an electron-density map calculated using observed structure-factor amplitudes for gene 5 protein and phase probabilities obtained using statistical density modification on the reconstructed image in Fig. 6(*b*). These phase probabilities were then combined with the original phase probabilities from the three-wavelength MAD experiment to yield a set of phase probabilities and a new electron-density map. The original *SOLVE* electron-density map (Terwilliger & Berendzen, 1999) using experimental phases is shown in Fig. 7(*a*). This map has a correlation with the model gene 5 protein map of 0.56. The electron-density map calculated from combined phases is shown in Fig. 7(*b*). This new electron-density map has a correlation to the model map of 0.65. Finally, the combined phases and the experimental structure-factor amplitudes were used in statistical density modification using the same parameters as those used to obtain the original *RESOLVE* phase probabilities. The resulting map is shown in Fig. 7(*c*); it is very similar to the original *RESOLVE* map shown in Fig. 5(*a*), but is slightly improved, with a correlation to the model gene 5 protein map of 0.82 (compared with 0.79 for the original *RESOLVE* map).

A key element of the process used here is to remove information about the density at each point x from the analysis of patterns of density around of x. We tested the importance of this step by repeating the entire process of generating templates and histograms and then applying them to the gene 5 protein MAD data, but without removing this information. In this case, the recovered image had a higher correlation with the model map than in the test case described above (0.55 compared with 0.25) and the smoothed recovered image had a correlation of 0.59, compared with 0.42. On the other hand, the correlation of errors between the recovered image and the starting *RESOLVE* map was also much higher (0.68 compared with 0.15), as was the correlation of errors between the smoothed recovered image and the starting *RESOLVE* map (0.85 compared with 0.23). Finally, the

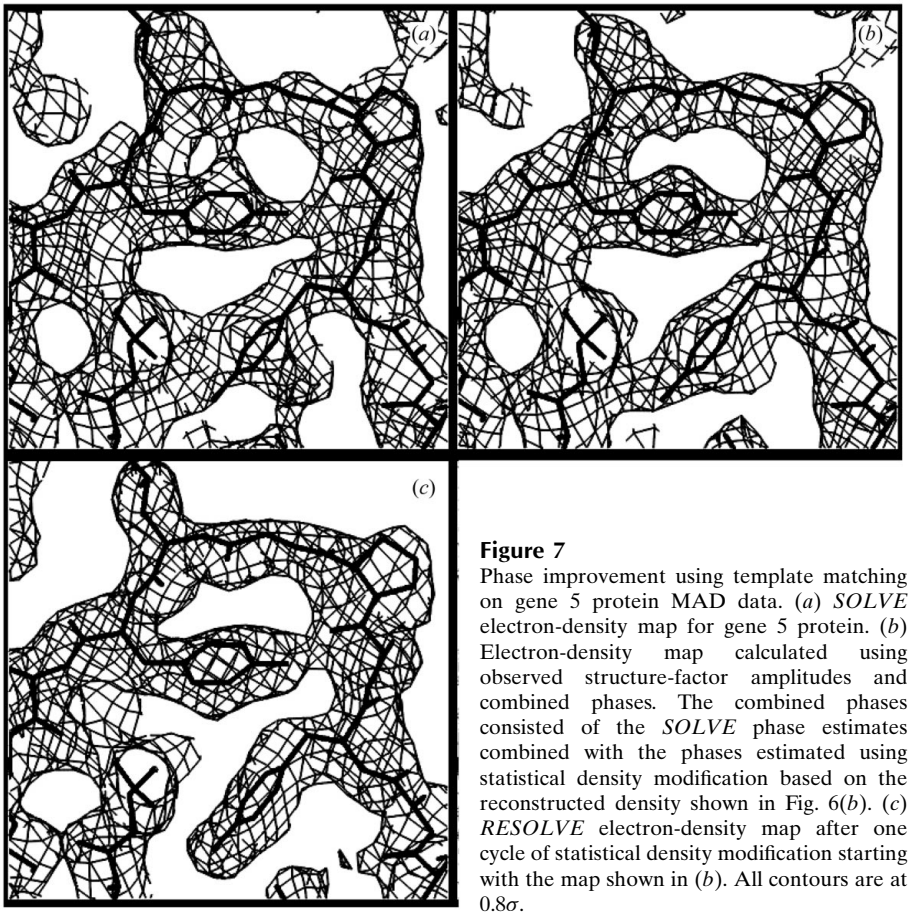


Figure 7
Phase improvement using template matching on gene 5 protein MAD data. (*a*) *SOLVE* electron-density map for gene 5 protein. (*b*) Electron-density map calculated using observed structure-factor amplitudes and combined phases. The combined phases consisted of the *SOLVE* phase estimates combined with the phases estimated using statistical density modification based on the reconstructed density shown in Fig. 6(*b*). (*c*) *RESOLVE* electron-density map after one cycle of statistical density modification starting with the map shown in (*b*). All contours are at 0.8σ.

electronic reprint

Table 2
Application of iterative statistical density modification with local pattern recognition.

For each experimental data set, density modification was carried out using default inputs for *RESOLVE* (Terwilliger, 2000) and phase probabilities calculated using *SOLVE* (Terwilliger & Berendzen, 1999). The process shown in Fig. 1 was then carried out, including the identification and use of local patterns of density. Non-crystallographic symmetry was not included in any density-modification procedures in these tests. The correlation coefficient of the resulting electron-density maps to those calculated with phases obtained from the refined models of each structure are listed. Additionally, the number of residues that could be automatically modeled and assigned to sequence and the number that could be modeled (whether or not assigned to sequence) with *RESOLVE* (Terwilliger, 2003*a*,*b*) using default parameters are listed. As the number of residues obtained with automated model building is somewhat sensitive to the parameters and details of the methods used, models were built with versions 2.02, 2.03, 2.04 and 2.05 of *RESOLVE* and the average numbers of residues built are reported.

Structure	UTP-synthase†	Armadillo repeat of β-catenin‡	Gene 5 protein§	Hypothetical (*P. aerophilum* ORF)¶	NusA††	NDP-kinase‡‡
Resolution (Å)	2.8	2.7	2.6	2.6	2.4	2.4
Type of experiment	SAD	MAD	MAD	MAD	SAD	MAD
RESOLVE map correlation to model map						
With local patterns	0.760	0.874	0.815	0.821	0.847	0.649
Without local patterns	0.727	0.872	0.786	0.811	0.648	0.586
Residues in refined model	1012 (2 × 506)	455	86	494 (2 × 247)	344	556 (3 × 186)
Main-chain residues built by *RESOLVE* (%)						
With local patterns	72	78	72	76	56	76
Without local patterns	72	78	69	76	49	76
Side-chain residues built by *RESOLVE* (%)						
With local patterns	34	58	52	65	21	18
Without local patterns	24	58	51	61	5	4

† Gordon *et al.* (2001). ‡ Huber *et al.* (1997). § Skinner *et al.* (1994). ¶ NCBI accession No. AAL64711; Fitz-Gibbon *et al.* (2002). †† D. H. Shin, H. T. Nguyen, J. Jancarik, H. Yokota, R. Kim & S.-H. Kim, unpublished work; PDB code 1l2f. ‡‡ Pédelacq *et al.* (2002).

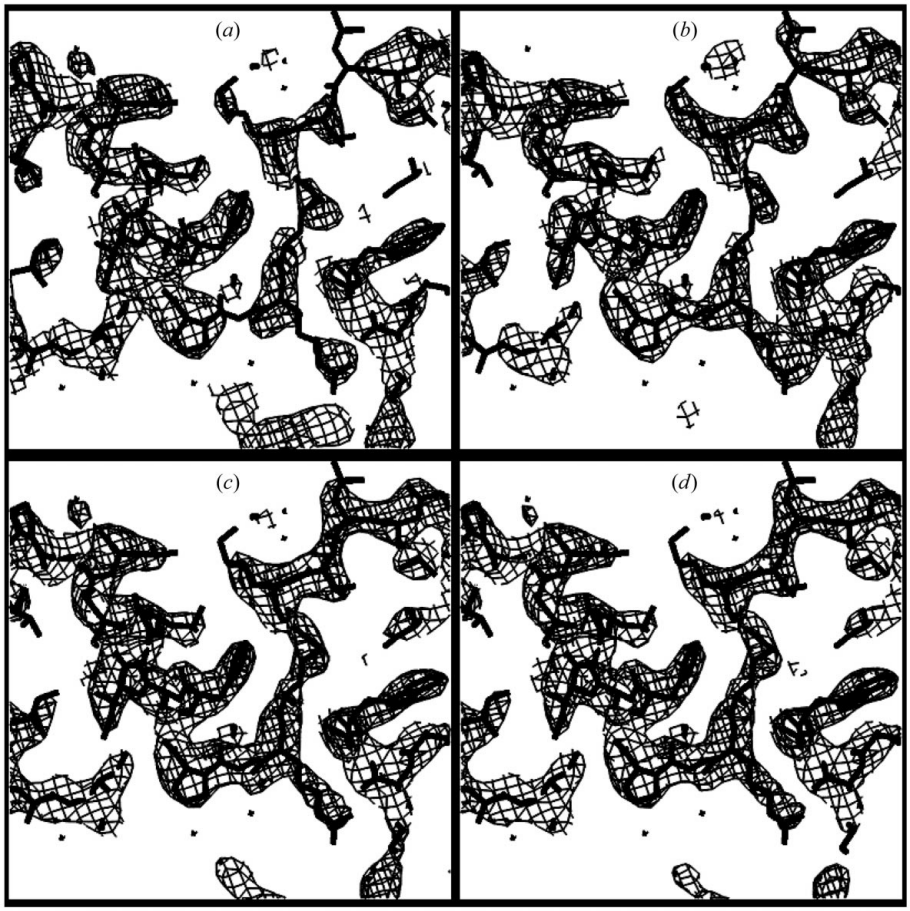


Figure 8
Phase improvement using template matching on nusA SAD data. (*a*) *RESOLVE* electron-density map for nusA protein calculated without pattern matching. (*b*), (*c*) and (*d*) Electron-density maps after one, three and five cycles of density modification including pattern matching, respectively. All contours are at 1.5σ.

resulting combined phases were used as a starting point for density modification, but in this case no improvement in the final map was obtained (correlation coefficient with the model map of 0.79 in both cases), supporting the idea that this step is an important element in the process.

3.6. Iterative local pattern identification and density modification

Fig. 1 illustrated an iterative process for phase improvement based on the local pattern identification described here. In this process, the pattern-identification step is always carried out on the best available map and then the resulting phase information is combined with experimental phase information to yield an improved starting point for density modification. The first cycle in this iterative process for phase improvement is identical to the process described above. Subsequent cycles simply iterate the process. Fig. 8 shows the results of applying the process to SAD data collected on nusA protein from *Thermotoga maritima* (D. H. Shin, H. T. Nguyen, J. Jancarik, H. Yokota, R. Kim & S.-H. Kim, unpublished data; PDB code 1l2f) at a resolution of 2.4 Å. Fig. 8(*a*) shows a section through the *RESOLVE*

electronic reprint

electron-density map obtained without using local pattern matching. Figs. 8(*a*), 8(*b*) and 8(*c*) show the density-modified map after one, three and five cycles using local pattern matching. The correlation coefficient of the starting *RESOLVE* electron-density map with a map calculated from the refined model of nusA is 0.65; the map after five cycles has a correlation of 0.85.

Table 2 summarizes the results of applying this process to experimental data from crystals of several different proteins. The greatest improvement in map quality was obtained for cases where the original *RESOLVE* map had a correlation with the model map of less than 0.7, with smaller improvements obtained when the *RESOLVE* map was better than this. To provide a rough measure of the utility of the method, the automatic model-building capability of *RESOLVE* was applied to the maps obtained for each structure with and without information from local patterns (Table 2). The percentage of main-chain residues built was essentially the same with and without information from local patterns for all the structures except nusA, which increased from 49 to 56% with the use of local patterns. On the other hand, the percentage of residues assigned to sequence and side chains built increased, on average, from 11 to 24% for those structures where the map correlation was considerably improved (UTP-synthase, nusA, NDP-kinase). This indicates that the map improvement can be enough to make a significant difference in the ability of automated procedures to build a complete atomic model.

Although the templates used in this procedure were calculated using data to 2.6 Å, the procedure is not strongly dependent on resolution. Using the nusA data as a test case, the effect of resolution was examined by truncating the analysis at resolutions of 2.4 (all data), 2.6, 2.8 and 3.0 Å, respectively. The correlation of the original *RESOLVE* maps at each of these resolutions with the model maps calculated at the same resolutions were similar (0.65, 0.66, 0.69 and 0.69, respectively), as were the correlations of the final maps density modified including the local pattern information (0.85, 0.85, 0.85 and 0.86, respectively).

4. Prospects

We have shown here that local features of electron-density maps can be used as an important source of information in a density-modification procedure. The improvements in map quality obtained using the information from local patterns range from none (0.87 to 0.87 for β-catenin) to small (from 0.79 to 0.82 in correlation coefficient for gene 5 protein) to very substantial (from 0.65 to 0.85 in correlation coefficient for nusA).

The computational requirements of the methods are moderate. Carrying out a complete set of five cycles of pattern identification and density modification using local patterns takes 90 min on a Compaq 833 Mhz Alpha for the 'hypothetical' protein from *P. aerophilum* listed in Table 2 (494 amino acids); standard density modification without using local pattern information takes about 5 min. Memory requirements are moderate as well: the libraries of patterns and indexing tables are large and (along with other parts of the software) require approximately 700 MB of swap space or more.

There are many additional applications of the procedures that we have developed here. A key aspect of the methods is that the image that is recovered from an electron-density map has errors that are relatively uncorrelated with those in the original map. This allows the use of the recovered image in phase improvement in the moderate-resolution range demonstrated here. It is also possible that the same approaches could be used for low-resolution as well as very high resolution phasing and phase extension. Additionally, the independence of errors means that an image recovered from a random map will have little or no correlation to the original map, while an image recovered from a map that has protein-like features will have a correlation. Consequently, the method could be used to evaluate the quality of protein electron-density maps. Similarly, points that are in the solvent region of a crystal will have local features unlike those found in the protein region and the methods described here could be used to distinguish the protein from solvent regions.

A weakness of the pattern-matching approach developed here is that it cannot readily distinguish protein-like features that are the result of systematic bias or errors in a map from those that actually reflect protein structure. This may be reflected in the small but significant correlation of errors between the density-modified model gene 5 protein map and its recovered image described above. Perhaps more importantly, it means that the method in its present form is not as well suited to improving maps that contain significant bias towards protein-like patterns of density, such as those obtained using phases from an atomic model, as it is to improving maps in which the errors are essentially random, such as those obtained by experiment.

A useful extension of the methods described here will be to recalculate the templates and histograms using data in various resolution ranges and using various radii for the regions considered in obtaining templates and to apply the appropriate set to experimental data. The effects of the grid spacing used in calculations could also be investigated. The use of correlations to more than two templates could be used in (8) in estimates of local density (although our preliminary investigations indicated that using a third template added very little information to the calculation). In each of the cases described here, the templates and histograms were obtained from model maps calculated at a resolution of 2.6 Å. The use of templates at varying resolutions could potentially increase the applicability of the method to a wider resolution range. Other extensions include examining the patterns in different classes of protein structures and in crystals that contain other structures such as nucleic acids or various ligands.

The author would like to thank the NIH for generous support, many colleagues for discussion, W. Weis for the use of β-catenin MAD data, E. Gordon for the use of dUTPase data,

electronic reprint

J. Remington for the use of RFP MAD data, S.-H. Kim for the use of nusA data and helpful reviewers for useful suggestions. The work has been carried out as part of the PHENIX project and methods described here are implemented in the software *RESOLVE* version 2.05, available from http://solve.lanl.gov.

References

Agarwal, R. C. & Isaacs, N. W. (1977). *Proc. Natl Acad. Sci. USA*, **74**, 2835–2839.
Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). *Nucleic Acids Res.* **28**, 235–242.
Bricogne, G. (1974). *Acta Cryst.* A**30**, 395–405.
Colovos, C., Toth, E. A. & Yeates, T. O. (2000). *Acta Cryst.* D**56**, 1421–1429.
Cowtan, K. (1999). *Acta Cryst.* D**55**, 1555–1567.
Fitz-Gibbon, S. T., Ladner, H., Kim, U. J., Stetter, K. O., Simon, M. I. & Miller, J. H. (2002). *Proc. Natl Acad. Sci. USA*, **99**, 984–989.
Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D. & Jacobs, D. (2003). *ACM Trans. Graph.* **22**, 83–105.
Goldstein, A. & Zhang, K. Y. J. (1998). *Acta Cryst.* D**54**, 1230–1244.
Gordon, E. J., Flouret, B., Chantalat, L. & van Heijenoort, J (2001). *J. Biol. Chem.* **276**, 10999–11006.
Harrison, R. W. (1988). *J. Appl. Cryst.* **21**, 949–952.
Huber, A. H., Nelson, W. J. & Weis, W. I. (1997). *Cell*, **90**, 871–882.
Jones, T. A., Zou, J. Y., Cowan, S. W. & Kjeldgaard, M. (1991). *Acta Cryst.* A**47**, 110–119.
Nieh, Y. P. & Zhang, K. Y. J. (1999). *Acta Cryst.* D**55**, 1893–1900.
Kleywegt, G. J. & Jones, T. A. (1996). *Acta Cryst.* D**52**, 826–828.
Kleywegt, G. J. & Read, R. J. (1997). *Structure*, **5**, 1557–1569.
Lamzin, V. S. & Wilson, K. S. (1993). *Acta Cryst.* D**49**, 129–147.
Lunin, V. Y. (1988). *Acta Cryst.* A**44**, 144–150.
Lunin, V. Y. (2000). *Acta Cryst.* A**56**, 73–84.
Lunin, V. Y. & Urzhumtsev, A. G. (1984). *Acta Cryst.* A**40**, 269–277.
Main, P. & Wilson, J. (2000). *Acta Cryst.* D**56**, 618–624.
Morris, R. J., Perrakis, A. & Lamzin, V. S. (2002). *Acta Cryst.* D**58**, 968–975.
Peat, T. S., Newman, J., Waldo, G. S., Berendzen, J. & Terwilliger, T. C. (1998). *Structure*, **6**, 1207–1214.
Pédelacq, J.-D., Piltch, E., Liong, E. E., Berendzen, J., Kim, C.-Y., Rho, B.-S., Park, M. S., Terwilliger, T. C. & Waldo, G. S (2002). *Nature Biotechnol.* **20**, 927–932.
Perrakis, A., Harkiolaki, M., Wilson, K. S. & Lamzin, V. S. (2001). *Acta Cryst.* D**57**, 1445–1450.
Perrakis, A., Morris, R. M. & Lamzin, V. S. (1999). *Nature Struct. Biol.* **6**, 458–463.
Perrakis, A., Sixma, T. K., Wilson, K. S. & Lamzin, V. S. (1997). *Acta Cryst.* D**53**, 448–455.
Read, R. J. (1986). *Acta Cryst.* A**42**, 140–149.
Rossmann, M. G. (1972). *The Molecular Replacement Method.* New York: Gordon & Breach.
Skinner, M. M., Zhang, H., Leschnitzer, D. H., Guan, Y., Bellamy, H., Sweet, R. M., Gray, C. W., Konings, R. N. H., Wang, A. H.-J. & Terwilliger, T. C. (1994). *Proc. Natl Acad. Sci. USA*, **91**, 2071–2075.
Terwilliger, T. C. (2000). *Acta Cryst.* D**55**, 1863–1871.
Terwilliger, T. C. (2001). *Acta Cryst.* D**57**, 1755–1762.
Terwilliger, T. C. (2003*a*). *Acta Cryst.* D**59**, 38–44.
Terwilliger, T. C. (2003*b*). *Acta Cryst.* D**59**, 45–49.
Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**, 501–505.
Urzhumtsev, A. G., Lunina, N. L., Skovoroda, T. P., Podjarny, A. D. & Lunin, V. Y. (2000). *Acta Cryst.* D**56**, 1233–1244.
Wang, B.-C. (1985). *Methods Enzymol.* **115**, 90–112.
Wilson, J. & Main, P. (2000). *Acta Cryst.* D**56**, 625–633.
Yarbrough, D., Wachter, R. M., Kallio, K., Matz, M. V. & Remington, S. J. (2001). *Proc. Natl Acad. Sci. USA*, **98**, 462–467.
Zhang, K. Y. J., Cowtan, K. & Main, P. (1997). *Methods Enzymol.* **277**, 53–64.
Zhang, K. Y. J. & Main, P. (1990). *Acta Cryst.* A**46**, 41–46.

electronic reprint

Back to RESOLVE table of contents

Running RESOLVE

- *Summary*
- *A basic script*
- *Notes on input/output mtz files*
- *Keywords for RESOLVE*
- *Keywords for RESOLVE_PATTERN*

Summary

To run RESOLVE, you need:

- *SOLVE and RESOLVE installed on your computer*
- *the CCP4 suite installed on your computer (or at least the file "symop.lib" somewhere)*
- *A CCP4 mtz file with at least FP, PHI, FOM for your dataset*
- *An estimate of the solvent content of your crystal*
- *To set the CCP4 environmental variables for file control and symmetry (modifying as appropriate for the location of symop.lib on your system):*

```
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
setenv CCP4_OPEN UNKNOWN
```

The most basic RESOLVE script (see the sample scripts for more cases):

```
#!/bin/csh
#
# Here is a minimal script to run RESOLVE on MAD/MIR/SAD etc data:
#
# Set CCP4 variables for symmetry information and
# for file handling:
#
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
setenv CCP4_OPEN UNKNOWN
#
# Now run RESOLVE:
#
resolve<<EOD
hklin solve.mtz
```

```
LABIN FP=FP PHIB=PHIB FOM=FOM HLA=HLA HLB=HLB HLC=HLC HLD=HLD
hklout resolve.mtz
solvent_content 0.4             ! your solvent content goes here.
            ! Next line is the file with your protein sequence.
seq_file protein.seq
EOD
#
# Now "resolve.mtz" has the output amplitudes, phases,
# and figure of merit in columns labelled: FP PHIM FOMM.
# A model of your structure is in resolve.pdb.
#
```

- RESOLVE will read from "solve.mtz" and write out new structure factor amplitudes and phases.
- If you have a file "ha.pdb" in your directory, RESOLVE will try to use the PDB coordinates in it to identify and apply NCS.
- Additionally, Hendrickson-Lattman coefficients are written.
- A model of your structure is built and placed in resolve.pdb
- You can use "resolve.mtz" in the same way as you use any mtz file in the CCP4 suite.

Notes on input and output mtz files (see the sample scripts)

- **RESOLVE expects to read data from a CCP4 mtz file**
  - **at least FP, PHI, FOM are needed for MAD/MIR/SAD data**
  - **for prime-and-switch, you need FP FC PHIC FOM, with optional FWT (from sigmaa)**
  - **it works much better for MAD/MIR/SAD data if you also input Hendrickson-Lattman coefficients (HLA HLB HLC HLD)**
  - **you specify the columns with the LABIN keyword as in the CCP4 suite**
  - **(LABIN FP=yourFPcolumn PHIB=yourPHIBestcolumn FOM=yourFOMcolumn HLA=yourHLAcolumn etc)**
  - **If your input file has reflections with non-zero F within your resolution range, it will automatically attempt to phase all of them.**
  - **You can input a FreeR_flag column.**
- **RESOLVE will write a CCP4 mtz file**
  - **The output columns of data are H K L FP SIGFP PHIM FOMM HLAM HLBM HLCM HLDM FreeR_flag**
  - **FP is just what was input; SIGFP is 1.0 if not input. NOTE: the SIGFP behavior may not be what you expect: If you do not specify SIGFP in your labin statement, resolve will write out SIGFP = 1.0. If you specify SIGFP=something, the data in column "something" will be copied out to SIGFP in the output file. If you use the default labin (FP=FP SIGFP=SIGFP PHIB=PHIB FOM=FOM HLA=HLA HLB=HLB HLC=HLC HLD=HLD) it will copy it out correctly.**
  - **PHIM is the modified "best" or centroid phase**
  - **FOMM is the figure of merit of PHIM. Use in a map with F = FP * FOMIM * exp(i PHIM).**
  - **HLAM, HLBM, HLCM, HLDM are Hendrickson-Lattman coefficients for the final phases.**
  - **Reflections with FreeR_flag = 0 are used as the "free" set in RESOLVE. Normally all reflections are used for all steps except the identification of the solvent fraction and a database for density statistics (i.e., the FreeR_flag is ignored for most steps).**
- **RESOLVE will write out a PDB file with a model of your structure**
  - **The model will usually not be fully complete, depending on the quality of the map**
  - **Some loops won't be present in most cases**
  - **Side chains will be built on the better parts of the model if you specify a file with the protein sequence**
  - **Side chains not in density will be truncated to CB**

Keywords for resolve

KEYWORD	DEFAULT	WHAT IT IS
access_file	solve2.access	Name of solve2.access file. If it is not in the /usr/local/lib/solve/ directory or in the current directory or in the directory $SOLVEDIR then you will want to tell RESOLVE where it is
hklin	solve.mtz	mtz file with input phases and phase probabilities
hklout	resolve.mtz	mtz file with output phases
LABIN FP=FP PHIB=PHIB FOM=FOM HLA=HLA HLB=HLB HLC=HLC HLD=HLD		LABIN statement identifying the columns of data in the hklin mtz file
mask_cycles	5	Number of cycles in which masks are redone and images are compared to map
minor_cycles	10	Number of minor cycles per mask_cycle
no_build		Don't build a model
build_only		Just build the model (no density modification)
assemble_only		Just assemble the model and write it out (using info in peak_file and fragment_file)
build_outside_model		Just build the model outside the region defined by model xxx.pdb
superquick_build		Build the model as quickly as possible, searching for fragments on coarse grid (works great for good maps and 10x faster than version 2.03 model-building) NOTE: superquick_build/quick_build/ thorough_build are 3 choices for 1 parameter
quick_build		(Default) Standard model-building protocol (3x faster than version 2.03)
thorough_build		Look exhaustively for ways to build the model. Similar to version 2.03. Not always any better than "quick_build"

aggressive_build Build aggressively, allowing some incorrect residues or even entirely incorrect residues.
Same as "macro_cycles 3". Not recommended.

conservative_build Build conservatively (same as version 2.03). Same as "macro_cycles 1".

seq_file protein.seq File containing protein sequence
Format: 1-letter code sequence of each chain, separated
by lines starting with &rt&rt&rt . No need to put in duplicate chains.

no_expand_ncs Don't expand number of copies of each chain beyond what was found with ha sites

seq_prob_min 0.95 Minimum confidence of sequence match to place side chains on a fragment

use_met_in_align Use heavy-atom positions as markers for MET positions

dist_cut_met 2.0 Consider a heavy-atom close to SD of Met if within this distance (default=2/3 of resolution)

richardson_rotamers Use Richardson Penultimate rotamer library from
SC Lovell, JM Word, JS Richardson and DC Richardson (2000) " The Penultimate Rotamer Library" Proteins: Structure Function and Genetics 40 389-408.

noget_peaks Skip searching for helices/strands and use data from peak_file

noget_fragments Skip searching for fragments and use data from fragment_file

peak_file resolve_peaks.dat Intermediate file with locations of all helices/strands considered

fragment_file fragments.dat Intermediate file with coordinates of all fragments considered in model-building

fom_cut 0.15 Set initial resolution for density modification to be where the FOM is about .15

s_step 0.02 Steps in s=1/d to take during phase extension

solvent_content 0.3 Fraction of unit cell in solvent region (if specified, do not search for optimal fraction)

use_input_solv Do not search for optimal solvent content
Use value from solvent_content if specified

Use value from resolve.solvent if it exists and solvent_content is not specified.

resolution Resolution limits (default=whatever is in input mtz file)

res_start 2.5 Start out density modification at this resolution, then extend to maximum

phase_extend Start out density modification at resolution res_start (set automatically), then extend to maximum

no_phase_extend Start out density modification at final resolution. This is default (changed in version 2.09)

phases_from_solve Input phases are not yet density-modified (default if "hklstart" is not set)

phases_from_resolve Input phases are already density-modified (default if "hklstart" is set)

use_free_for_test Use free set for solvent content/histogram tests (default unless Nfree<500)

use_all_for_test Use all data for solvent content/histogram tests

create_free Create FreeR_flag if it does not exist (default)

(This does not mean that the FreeR_flag is used)

include_free Use all data for main cycles (default)

exclude_free Do not include free data for main cycles (does not work well)

database 1 Use database entry #1 for histograms of protein/solvent density

use_input_db Do not search for optimal histogram from database
Use value from "database" if specified
Use value from resolve.database if it exists and "database" is not specified.

wang_radius_cycle 6. 4. (default: variable start; 4 end) Starting and ending radius (A) in Wang method for getting solvent mask
NOTE: RESOLVE will automatically set this for you in a reasonable way if
you don't specify it.

wang_radius_finish 4. (default: 4.0) Ending radius (A) in Wang method for getting solvent mask

wang_radius_start 6. (default: variable) Starting radius (A) in Wang method for getting solvent mask

wang_radius 6. Radius to be used for all cycles
NOTE: In RESOLVE you don't want or need a small radius for getting the
solvent probability. It works better with a medium-large radius (but really it
makes very little difference what radius you use)

hklstart mtz file with a starting set of phases. RESOLVE will start with
these phases (but use probabilities from hklin). Useful for
running a few cycles, getting an output resolve_1.mtz, then
continuing on from there. Goes with labstart.

labstart FP=FP PHIB=PHIB FOM=FOM LABIN statement for hklstart. only PHIB and FOM used.

hklperfect mtz file with a model set of phases
labperfect FP=FP PHIB=PHIB FOM=FOM LABIN statement for hklperfect
difference_map calculate difference map FP hklin-FP hklperfect
phase_with_perf use PHIB and FOM from hklperfect in diff map (default is use PHIB and FOM from hklin)

cc_ratio 0.1 Cut off prime-and-switch phasing at resolution where CC
of FP with FC < cc_ratio

verbose print out a lot of data every cycle, not just at first and last.

nohl don't calculate HL coefficients at end of resolve (saves time)

new_ncs_group start a new group of NCS operators

ncs_domain_pdb domain_1.pdb PDB file identifying all points in the asymmetric unit that are part of this NCS group. See also rad_mask.

rota_matrix rotation matrix for NCS symmetry. Three rota_matrix lines define a matrix:
Enter the identity as first molecule 1.
rota_matrix 1 0 0
rota_matrix 0 1 0
rota_matrix 0 0 1
The rotation matrix applies to orthogonal Angstrom coordinates.
The matrix and translation maps molecule j on to molecule 1.
This is what you get from the ccp4 program lsqkab if molecule 1 is
the reference (xyzin1) and molecule j is the working molecule (xyzin2)

tran_orth Translation vector for NCS symmetry element, in orthogonal Angstroms
Enter after rota_matrix

center_orth Approximate center of mass of NCS symmetry element, in orthogonal Angstroms
Enter after rota_matrix. Required only for element 1

fraction_ncs 0.15 fraction of asymmetric unit occupied by 1 molecule (default = fraction protein / N)

invert Invert NCS matrices (they correspond to mapping molecule 1 on to molecule j)

fix_ncs Do not refine NCS operators

ncs_restrict n Only consider NCS if there are n operators. ncs_restrict 3 looks only for trimers.

force_ncs Use input or heavy-atom-site NCS symmetry even

if there is very low correlation

rad_ncs_mask_max 6. Do not allow NCS mask to have radius greater than rad

ha_file ha.pdb Use the entries in this file (PDB format) to look for non-crystallographic symmetry

compare_file xxx.pdb Use coords in this file to define asymmetric unit for output resolve_compare.pdb
NOTE: has no effect on contents of resolve.pdb

no_find_ncs Don't try to find NCS from heavy-atom sites even if the file "ha.pdb" exists

ncs_only Find NCS and do nothing else (no density modification, nothing)

do_all_cycles Do all the assigned cycles, even if nothing is happening

no_fill Don't fill in missing reflections (default) (best usually)

fill Fill in missing reflections (risky)

r_match (Default=1.0) Maximum distance between CA in different fragments to link during assembly

r_min (Default=1.) Lowest minimum rho/sigma for main-chain atoms

r_min_side (Default=0.3) Lowest minimum rho/sigma for side-chain atoms

r_overall (Default=0.75) Minimum rho_bar/avg for starting a segment in model-building

r_end (Default=0.5) Minimum rho_bar/avg for continuing a segment in model-building

z_cut (Default=0.5) Minimum Z-score to keep a fragment after refinement

z_cut_extend (Default=-0.5) Minimum Z-score to keep a fragment after extension

macro_cycles 3 (Default = 2) Number of iterations of lowering thresholds in model-building.
NOTE: normally do not use more than 3 or the model becomes poor.
Each cycle beyond 1 r_min and r_min_side decrease by 0.5 and
z_cut and z_cut_extend decrease by 1.

no_unassigned Do not write out residues not assigned to sequence in model-building

rad_mask 2.5 Radius for calculation of solvent mask and for inclusion in image-based phasing and for NCS region identification

image Use map calculated from PDB file ("composite_pdb") or from input phases as target for electron density. Only density within rad_mask of atoms used if from PDB file.

image_only Do not do solvent flattening/NCS etc. Just use the image as a target. Requires the keyword "image" as well. Produces phases similar to "sigmaa".

prior_weight 1.0 (Default=1.0) Weight on the input phases. Prior_weight 0 is used in prime-and-switch phasing.

prime_and_switch Use prime-and-switch phasing. Use input phases only to calculate initial map. Input phase probabilities not used at all.

no_erase_protein Do not let P(protein) be less than the starting value in prime-and-switch phasing.

n_image_cycle Number of cycles of using the image (map) based on a model in density modification

composite_pdb xxx.pdb_ root name of a set of PDB files to be used to construct a composite image

composite_pdb_first 0 first PDB file for composite is xxx.pdb_0

composite_pdb_last 20 last PDB file for composite is xxx.pdb_20

pdb_in refmac.pdb Name of PDB file to be used in starting model-building

extend_only Just trim and extend the chains in pdb_in; don't rebuild from scratch

no_merge_ncs_copies Do not merge NCS copies during extend (default = merge_ncs_copies)

side_avg_min 0.0 Truncate side-chain atoms in model-building if mean density is < side_avg_min

LABIN FP=FP FC=FC PHIC=PHIC FOM=WCMB FWT=FWT LABIN statement suitable for prime-and-switch using SIGMAA phases.
FWT is optional; if present initial map calculated with FWT exp(i PHIC).

pattern_phase Use image in cc_map_file as target for image-based phasing (no prior phase information,
no solvent flattening, no NCS.

cc_map_file recovered_map.dat binary file with map for pattern_phase targeting (only read by RESOLVE)

coarse_grid Use coarse grid (same as RESOLVE 2.02) for maps. Must match RESOLVE_PATTERN

n_restore 1 Number of times to restart density modification using mask
from previous run but starting with original phases. Default = 0 (no restarting)

no_restore Same as n_restore 0

scale_refl 0.5 Weighting on map probability function (default = 0.5)

scale_refl_start 0.05 Weighting on map probability function (default

= 0.5) on first cycle of density modification

scale_refl_end 0.5 Weighting on map probability function (default = 0.5) on last cycle of density modification

trim Trim pdb_in file back to match density (default)

no_trim Take main-chain of pdb_in file as is and use as a basis for model-building without trimming

i_ran_seed value of random seed

no_cut_up_model Don't cut up pdb_in into little pieces (default)

cut_up_model Cut pdb_in up into little and big pieces and try them all as starting points for model-building

build_image Use FFT-based search to find helices/strands and create an output map (dump.map) with reconstructed image of the map. You can then use dump.map with "pattern_phase"

evaluate_model Compare the model defined by the model keyword with the map calculated from hklin

model resolve_best.pdb model to evaluate with evaluate_model

ncs_mask_file mask.mtz write out a CCP4-style mtz file with FP PHIM FOM that yield a map showing the NCS asymmetric unit.

protein_mask_file mask.mtz write out a CCP4-style mtz file with FP PHIM FOM that yield a map showing the protein region.

add_mask Require that the region defined by the PDB file read in with model xxx.pdb is protein
use rad_mask radius in definition of region

no_ha do not write heavy-atom sites out to the resolve.pdb model file

ha_occ 1.0 set occ of heavy-atoms written out from ha.pdb to resolve.pdb to this value (default=0.0)

start_chain 1 23 start chain 1 with residue number 23 (default = 1)

start_segment n read segments files starting with number "n"

max_segment m read up to "m" segments files. These are files with information about helices/strands etc.

score_only score an electron density map (skew, ha_ncs, correlation of map from map-probability phases with original map, correlation of local rms)

score_tert score tertiary structure

loop_only Fit a loop only. Requires pdb_in and extend_only

n_random_loop 20 Number of loop conformations to try. If negative, try only randomized configurations.

rms_random_loop 0.3 RMS random variation in loop coordinates

n_random_frag 0 Number of randomized fragment conformations to try

rms_random_frag 0.3 RMS random variation in frag coordinates

pieces_only 10 Used with cut_up_model;

do not include the uncut part and cut into sizes pieces_only. If negative,

start at every amino acid and make a fragment of length pieces_only and try to extend.

skip_hetatm Ignore HETATM records in PDB files.

cut_1 0.5 Minimum rho for residue/average to continue segment

cut_2 0.75 Minimum rho for residue/average to start/stop a segment

dist_close 1.0 (0.5 for cross of 2 models) Maximum distance of CA to be considered the same (A)

rebuild_in_place Rebuild the model in segments, preserving sequence alignment.

n_try_rebuild 1 Number of times to try to find each segment

replace_existing Replace existing segments in rebuild_in_place even if worse than starting coordinates (default is to keep existing if better)

rho_min_main_low 1.0 Minimum density at atoms in loop. Default=1, try 0.5; goes with rho_min_main_base

rho_min_main_base 1.0 Minimum density at atoms in loop. Default=1, try 0.5; goes with rho_min_main_low

no_sub_segments Fit entire segments to sequence (do not break up)

omit_box n Omit all points in omit box n from density modification

n_box_target m Try to set up m omit boxes

omit_boundary 2.0 Increase the size of the omit region defined by omit_box
in all directions by this amount

complete_omit Do not include solvent flattening or histogram matching in omit region (default)

no_complete_omit Include solvent flattening and histogram matching even in omit region

complete_omit_hist Include histogram matching even in omit region

Keywords for RESOLVE ligand fitting

ligand_file ligand.pdb PDB file with N copies of a ligand in random stereochemically ideal conformations. All atoms must be in the same order in all copies

n_ligand_pos 300 Number of rotation/translation positions for a ligand to consider

n_ligand_pos_ref 100 Number of rotation/translation positions for a ligand to refine

n_group_search 3 Number of large groups within ligand to search for.

If zero, groups from input file used without rotation/translation to start fitting

group_search xx Use group xx in ligand to start ligand fitting

n_keep_plac 100 Number of placements of groups to keep in ligand search

n_indiv_tries_min 20 Minimum nmber of top placements of large groups to try individually in ligand search

n_indiv_tries_max 20 Maximum nmber of top placements of large groups to try individually in ligand search

ligand_resno 20 Residue # for ligand

n_template_atom 45 Number of atoms in ligand (overrides guess made by resolve)

no_local_search Search whole map for ligand

search_dist 10 Search within this distance of top rms region in map for local searches

search_center 3 10 5 Local search is around this point (in A)

fit_phi_range -20 20 Try angles from -20 to 20 degrees relative to the torsion angles in ligand_file

fit_phi_inc 20 Sample angles in range fit_phi_range with increment fit_phi_inc

delta_phi_ligand 50 Rotation angle to sample overall placements of large groups in ligand fitting

acceptable_offset 1.0 Maximum allowed offset of atoms from overlapping placements in ligand search (i.e., placements of an atom from each of 2 directions in a cyclic molecule)

cut_close 1.5 Minimum allowed separation between atoms in separate rigid groups.

ignore_map Ignore the map and just generate a random conformation.

model xxx.pdb Generate FC from model and use Fo-Fc map to fit ligand.

Keywords for resolve_pattern

KEYWORD	DEFAULT	WHAT IT IS
access_file	solve2.access	Name of solve2.access file. If it is not in the /usr/local/lib/solve/ directory or in the current directory or in the directory $SOLVEDIR then you will want to tell RESOLVE where it is
hklin	solve.mtz	mtz file with input phases and phase probabilities
hklout	resolve.mtz	mtz file with output phases
LABIN FP=FP PHIB=PHIB FOM=FOM HLA=HLA HLB=HLB HLC=HLC HLD=HLD		LABIN statement identifying the columns of data in the hklin mtz file
resolution		Resolution limits (default=whatever is in input mtz file)
		Use value from resolve.database if it exists and "database" is not specified.

recover_image		Analyze map calculated with hklin phases and amplitudes for local patterns; write out binary map with the recovered image to cc_map_file
cc_map_file	recovered_map.dat	File with recovered map (binary file; only read by RESOLVE)
coarse_grid		Use coarse grid (same as RESOLVE 2.02) for maps. Must match RESOLVE
path_patterns	$SOLVEDIR/patterns/	Location of library files for RESOLVE_PATTERN

```
#!/bin/csh
set main_version = 2
set minor_version = 08
set edit_version = 07
echo " "
echo "RESOLVE difference map script version $main_version.$minor_version.$edit_version "
echo " "
echo "Date: `date`"
echo "Working directory: `pwd`"
echo " "
#
#                  RESOLVE difference map script
#                    05-Jan-2005
#
#                   Tom Terwilliger
#            Los Alamos National Laboratory
#
#
#    Input:  FP, model
#
#    Methods:  Just calculate a difference map using phases from model
#
#    Output: Map coefficients FP PHIM FOMM in resolve_diff.mtz
#           CCP4-style map file in resolve_diff.map
#
#=====================================================================
#     EDIT THE NEXT FEW SETS OF LINES TO MATCH YOUR DATA AND SYSTEM
#
#     PLEASE NOTE: each of the " = " below must have a SPACE on either
#       side (hklin = solve.mtz  NOT hklin=solve.mtz )
#
#=====================================================================
#
#  Anything special for your location (SOLVEDIR etc);
#
setenv SOLVEDIR /usr/local/lib/solve/
#
# location of resolve and resolve libraries;  place for large scratch files
#
set resolve =  /u1/terwill/resolve/work/resolve.linux
setenv TMPDIR .                        # large scratch files go here
#
set hklin = FOBS.mtz
set labin = 'FP=FP SIGFP=SIGFP'
set labin_cont = "
#
```

```
#   input model
#
set pdb_in =  PARTIAL_MODEL.pdb
#
#   resolution
#
set dmin =  NONE #   NONE to use all data...high-res limit
set dmax =  NONE #   NONE to use all data...low-res limit
#
#==============================================================================
#               Normally no need to edit below here...
#==============================================================================
#==============================================================================
#
set main_version_minimum = 2
set minor_version_minimum = 08
#
setenv SYMOP $SOLVEDIR/symop.lib
setenv SYMINFO $SOLVEDIR/syminfo.lib
setenv CCP4_OPEN UNKNOWN
unlimit
limit coredumpsize 0
#
set ip = 0
#-------------------------------------------------------------
#  check for all the library files we will need etc
#
if ( $resolve == NONE ) then
 echo "Sorry, you need to define resolve ..."
 exit
endif
foreach program ($resolve)
if ( $ip ) echo "Checking for the program $program ..."
if ( -f $program ) then
 if ( $ip ) echo "OK"
 goto ok
endif
if (  $program == NONE ) then
 if ( $ip ) echo "This program is not used"
 goto ok
endif
which $program >& tmp.dat
set test = `head -1 tmp.dat`
if ( $#test != 1 ) goto bad
if ( -f $test ) then
 if ( $ip ) echo "OK"
 goto ok
```

```
endif
bad:
 echo "Sorry, the program $program does not exist?"
 echo "Please check its definition in this script..."
 exit
endif
ok:
end
#
#
foreach lib_file ( symop.lib syminfo.lib)
#
if ( $ip ) echo "Checking for library file $SOLVEDIR/$lib_file..."
if ( ! -f $SOLVEDIR/$lib_file) then
 echo "Sorry, the library file $lib_file does not exist?"
 echo "Please check the definition of SOLVEDIR in your script: $SOLVEDIR"
 exit
endif
end
#-------------------------------------------------------------
#
# check for resolve
if ( -f resolve.ok ) rm resolve.ok
if ( -f resolve.version ) rm resolve.version
$resolve <<EOD  >& resolve.log
 quit
EOD
if ( ! -f resolve.ok ) then
 echo "Sorry please check the location of resolve..."
 echo "    is it really $resolve?"
 exit
endif
#  make sure this sort of worked...
set test_ok = `cat resolve.ok`
set test_ok_mem = $test_ok[$#test_ok-$#test_ok]
if ( $test_ok_mem != "ALLOCATED" ) then
 echo "Sorry, resolve was not able to run properly at all"
 echo "The end of the resolve log file says..."
 echo "-------------------------------------------------------"
 tail -12 resolve.log
 echo "-------------------------------------------------------"
 exit
endif
if ( ! -f resolve.version ) then
 echo "Sorry, this script requires version $main_version_minimum.$minor_version_minimum or higher of resolve"
 exit
endif
```

```
@ version = `cat resolve.version|head -1`
set version_minimum = "$main_version_minimum$minor_version_minimum"
if ( $version < $version_minimum )then
 echo "Sorry, this script requires version $main_version_minimum.$minor_version_minimum or higher of resolve"
 exit
else
 if ( $ip ) echo "Running version $version of resolve"
endif
#
# make sure we have these files...
#
if ( -f $hklin ) then
   echo "Data are in the mtz file $hklin"
else
   echo "Sorry, cannot find the mtz file $hklin"
   exit
endif
echo "LABIN information: $labin $labin_cont"
echo " "
#
if ( ( $pdb_in != NONE ) && ( ! -f $pdb_in ) ) then
 echo "Sorry, cannot find your pdb_in file $pdb_in"
 exit
endif
#
#-----------------------------------------------------------------
#  figure out if this machine uses grep -a or just grep for text files:
echo "A" > test_a.dat
 set test_grep = `grep -a "a" test_a.dat >& tmp.dat`
if ( $status ) then
#  there was an error...do not use grep -a
 set grep_type = "grep"
else
 set grep_type = "grep -a"
endif
set test_grep = `$grep_type "A" test_a.dat`
if ( $#test_grep != 1 ) then
 echo "Sorry, unable to set the grep command on this system...giving up"
 exit
endif
rm test_a.dat
#-----------------------------------------------------------------

#
@ grep_phib = `echo $labin $labin_cont | $grep_type 'PHIB='|wc -m`
@ grep_fp = `echo $labin $labin_cont | $grep_type 'FP='|wc -m`
#
```

```
if ( $grep_fp == 0 ) then
 echo "Sorry, FP is always required in the labin line..."
 exit
endif
if (  $pdb_in == NONE ) then
 echo "Sorry you need a pdb_in file"
 exit
endif
#
if ( $dmin == NONE || $dmax == NONE )then
  set resolution_line = ""
else
  set resolution_line = "resolution $dmin $dmax"
endif
echo " "
echo "${resolution_line} "
echo " "
#
set blank = " "
#
#          SETUP:
#
# ==========================================================================
# ==========================================================================
# hklin_image will be our combined phase information
#
set hklin_image = start_image.mtz
set labin_image = 'FP=FP PHIB=PHIM FOM=FOMM HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM'
set labin_image_cont = 'FreeR_flag=FreeR_flag'
#
set hklin_exp = exp.mtz
set labin_exp = 'FP=FP PHIB=PHIM FOM=FOMM HLA=HLAM HLB=HLBM HLC=HLCM HLD=HLDM'
set labin_exp_cont = 'FreeR_flag=FreeR_flag'
#
echo "Copying information from $hklin to $hklin_exp"
${resolve}<<EOD >>resolve.log
! Copying over $hklin to $hklin_exp
hklin $hklin
labin $labin
labin $labin_cont
hklout $hklin_exp
$resolution_line
mask_cycles 1
minor_cycles 0
no_build
EOD
#
```

```
#  make sure this sort of worked...
if ( `cat resolve.ok` != 'OK' ) then
 echo "Sorry, resolve was not able to finish even the first cycle"
 echo "The end of the resolve log file says..."
 echo "-------------------------------------------------------"
 tail -10 resolve.log
 echo "-------------------------------------------------------"
 exit
endif

#
#
set pdb_start = $pdb_in            #   starting PDB file
cp $pdb_in refine.pdb_0
#
echo " "
#
${resolve}<<EOD >>resolve.log
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
$resolution_line
fcalc $pdb_in
hklout fcalc.mtz
EOD
#
#  make sure this sort of worked...
if ( `cat resolve.ok` != 'OK' ) then
 echo "Sorry, resolve was not able to finish on the first cycle"
 echo "The end of the resolve log file says..."
 echo "-------------------------------------------------------"
 tail -10 resolve.log
 echo "-------------------------------------------------------"
 exit
endif
#
$resolve<<EOD>>resolve.log
hklin $hklin_exp
labin $labin_exp
labin $labin_exp_cont
hklperfect fcalc.mtz
labperfect FP=FP PHIB=PHIM
difference_map
phase_with_perf
hklout resolve_diff.mtz
EOD
#
```

```
if ( -f resolve_map.mtz ) then
$resolve<<EOD>>resolve.log
hklin resolve_diff.mtz
labin FP=FP PHIB=PHIM FOM=FOMM
mask_cycles 1
minor_cycles 0
no_build
ccp4_map_file resolve_diff.map
EOD
 echo "resolve_diff.mtz  ... Coefficients for difference map only FP PHIM FOMM"
 echo "resolve_diff.map  ... ccp4-style map from resolve_diff.mtz"
endif
#==================================================================
```

How to interpret the solve.status file

The solve.status file keeps a running summary of the status of SOLVE. As additional steps are carried out by SOLVE, the results are added to the end of the solve.status file. It is convenient to look at this file continuously using the unix command "tail -30f solve.status" which will show the last 30 lines in the file, updating them every few seconds.

The solve.status file will show you:

- The title of the experiment if you have input one
- The step that SOLVE is currently on (e.g., SCALE_MAD, SOLVE)
- The time SOLVE has spent on this step
- If the program has begun automated structure determination (SOLVE) then the solve.status file will show statistics on the best solution found so far:
    - The figure of merit (m) and overall Z-score
    - The coordinates, occupancies and overall B-factors for each atom in the solution
    - The peak heights (normalized to the rms of the maps) of each heavy atom site in cross-validation difference Fouriers
    - The current resolution that SOLVE is working at
- The total time that SOLVE has been running
- If SOLVE is unable to carry out a command (e.g., an input file is missing) then the bottom of the solve.status file will tell you where to look for more detailed information about what happened (usually it will tell you to look at the end of the logfile).

The solve.status file can give you a pretty good indication of whether SOLVE has found a good solution. A correct solution will have high values of peak heights for all the heavy atom sites in the cross-validation difference Fouriers (for most good solutions these peak heights will all be more than 5 or 10 sigma). A good solution will have a high Z-score (typically above 20; this score is higher for structures with more heavy atoms). A good solution will have a figure of merit around 0.6 to 0.7 as well.

See How to interpret the output from SOLVE as well for information on this.

Contents	Index

Commands for controlling how SOLVE works in automated mode

This is a list of the most commonly-used keywords for SOLVE that apply to SOLVE operation in automated mode. Also see the list of all SOLVE keywords and the list of all commands .

```
These keywords are usually in your solve.setup file:

SYMFILE xxxxx       symmetry file for this space group
CELL   a b c alpha beta gamma
RESOLUTION dmin dmax

RES_PHASE  xx        Carry out phasing heavy-atom searches at resolution XX
                     (but write out all data in resolution range set by the
                      keyword "resolution"). Usually you should let SOLVE set this.

FFTGRID  xs xe xtotal ys ye ytotal zs ze ztotal    grid for FFT calculations
PATTGRID xs xe xtotal ys ye ytotal zs ze ztotal    grid for Patterson
EZDGRID  xs xe    ys ye    zs ze                   grid for NEWEZD map

These keywords are usually in your main command file:

VERBOSE                         write out a lot of output to logfile
RAWMADFILE  xxx.int   File xxx.int will be read in as data for
                      the current mad wavelength
RAWNATIVEFILE  xxx.int   File &xxx.int& will be read in as data for
                         the native
RAWDERIVFILE  xxx.int   File xxx.int will be read in as data for
                        the current derivative
READ_INTENSITITES  (default)  The raw data files contain intensity measurements
READ_AMPLITUDES    The raw data files contain amplitudes (F) not intensities (I)
                   (This is valid only with READFORMATTED)

PREMERGED          The data in all RAWMADFILEs have H K L and 4 other columns:
                   I+/F+, sigma, I-/F-, sigma
UNMERGED           The data in all RAWMADFILEs have H K L and 2 other columns:
                   I/F, sigma

READDENZO        All datafiles are written by Scalepack.  For unmerged data
                 they will be read with the formatting:(6i4,i6,2i2,i3,2f8.0) and
                 nsym*2+1 lines are skipped at the top of the file. For
                 merged data the formatting is:  (3i4,4f8.0) and
                 3 lines are skipped at the top of the file.

READFORMATTED      All datafiles will be read with "*" formatting and
                   contain H K L I/F sigma or H K L I+/F+ sigma I-/F- sigma

READTREK          The datafiles were written by d*trek and contain columns
```

```
                  with intensities

LABIN             specify column assignments for HKLIN in standard CCP4
                  fashion (FC=FC1 PHIC=PHIC FOM=FOM) etc

HKLIN  xxx.mtz    mtz file containing scaled amplitudes

PHASES_FORMATTED xxx.fmt   File  xxx.fmt contains H K L FC PHIC FOM and
                    the phases and fom will be used in SOLVE with
                    difference Fouriers to find initial sites

PHASES_LABIN        specification of column assignments for PHASES_MTZ file
                    Normal use is FC=FC PHIC=PHIC FOM=FOM. NOTE: MUST
                    come before PHASES_MTZ!

PHASES_MTZ xxx.mtz    as PHASES_FORMATTED, but mtz-file. FC PHIC FOM required

NSKIP n            Skip exactly n lines at the top of each data file
NSKIP 0            Do not skip any lines at the top of each data file
NSKIP -1           Skip 0 lines at the top of each data file
                    unless the keywords READDENZO and PREMERGED
                    are set in which case the default number of lines
                    are skipped (see above)

RATMIN  xx          Minumum ratio of F/sig to read in data for a
                    reflection at all is xx [default=2.0].  This is
                    useful for eliminating weak data.

FPFM_ONLY          Toss all acentric reflections where either F+
                    or F- is missing [this is the default for MAD data]

FP_OR_FM           Use F+ or F- as an estimate of Fbar if F+ and
                    F- are not both present.  This is useful if
                    your data is not that complete.  It is much
                    better to obtain complete data however.

OVERALLSCALE       Do not do local scaling; just an overall
                    scale factor for F+, F- at each wavelength.
                    Use this if you already have scaled the data
                    and you don't want any more scaling done.


SWAP_ANO            Swap H K L -> -H -K -L as data are read in to
                    SOLVE in scale_native, scale_derivative, and
                    scale_mad. This is to correct for a detector or indexing
                    that swapped F+ for F-

The atom types recognized by SOLVE are:
H, H-1, He, Li, Li+1, Be, Be+2, B, C, Cv, N, O, O-1,
F, F-1, Ne, Na, Na+1, Mg, Mg+2, Al, Al+3, Si, Siv, Si+4,
P, S, Cl, Cl-1, Ar, K, K+1, Ca, Ca+2, Sc, Sc+3, Ti, Ti+2,
Ti+3, Ti+4, V, V+2, V+3, V+5, Cr, Cr+2, Cr+3, Mn, Mn+2, Mn+3,
```

Mn+4, Fe, Fe+2, Fe+3, Co, Co+2, Co+3, Ni, Ni+2, Ni+3, Cu,
Cu+1, Cu+2, Zn, Zn+2, Ga, Ga+3, Ge, Ge+4, As, Se, Br,
Br-1, Kr, Rb, Rb+1, Sr, Sr+2, Y, Y+3, Zr, Zr+4, Nb, Nb+3,
Nb+5, Mo, Mo+3, Mo+5, Mo+6, Tc, Ru, Ru+3, Ru+4, Rh, Rh+3,
Rh+4, Pd, Pd+2, Pd+4, Ag, Ag+1, Ag+2, Cd, Cd+2, In, In+3,
Sn, Sn+2, Sn+4, Sb, Sb+3, Sb+5, Te, I, I-1, Xe, Cs, Cs+1,
Ba, Ba+2, La, La+3, Ce, Ce+3, Ce+4, Pr, Pr+3, Pr+4, Nd,
Nd+3, Pm, Pm+3, Sm, Sm+3, Eu, Eu+2, Eu+3, Gd, Gd+3, Tb,
Tb+3, Dy, Dy+3, Ho, Ho+3, Er, Er+3, Tm, Tm+3, Yb, Yb+2,
Yb+3, Lu, Lu+3, Hf, Hf+4, Ta, Ta+5, W, W+6, Re, Os, Os+4,
Ir, Ir+3, Ir+4, Pt, Pt+2, Pt+4, Au, Au+1, Au+3, Hg, Hg+1,
Hg+2, Tl, Tl+1, Tl+3, Pb, Pb+2, Pb+4, Bi, Bi+3, Bi+5, Po,
At, Rn, Fr, Ra, Ra+2, Ac, Ac+3, Th, Th+4, Pa, U, U+3, U+4,
U+6, Np, Np+3, Np+4, Np+6, Pu, Pu+3, Pu+4, Pu+6, Am, Cm, Bk, Cf

```
newatomtype xxxx    define scattering properties of atom xxxx not recognized
                    by SOLVE
                    HINT: to get the aval, bval, cval values from Int tables
                    for atoms recognized by SOLVE, type mad_atom [atomname] and
                    SOLVE will list them for you.
aval a1 a2 a3 a4    4 real numbers (a1,a2,a3,a4)  from International Tables
for
                     the most recently defined newatomtype
bval b1 b2 b3 b4     b values for newatomtype
cval c              c value for newatomtype
fprimv xx           f' value for newatomtype
fprprv xx           f" value for newatomtype

                    For CLUSTERS, follow the instructions in
                    all_keywords.


mad_atom  xxxx   name of the anomalously scattering atom is xxxx.  If SOLVE does
                 not recognize this atom, first input its scattering parameters
                 with newatomtype
fprimv_mad       f' value for anomalously scattering atom
                 at this wavelength (must be input after each wavelength)
fprprv_mad       f" value for this wavelength

FIXSCATTFACTORS     Fix scattering factors at their input values.
                    This is a good idea if you have a reasonable
                    idea of the f' and f" values.  [this is the
                    default]

REFSCATTFACTORS     refine scattering factors f' and f".  If you
                    refine them, be sure to look at their new
                    values at the end of the routine MADMRG and
                    verify that they are reasonable.

derivative n        begin input of information for derivative/wavelength n
                    This command is used to start entering information on a
```

```
                    derivative.  If you want to modify something after you've
                    gone on to another derivative then you need to use the
                    command GOTODERIV
lambda n            identical to derivative n

nsolsite_deriv      Maximum number of sites for this derivative
                    (overrides nsolsite)

cutoff_deriv 200 3.5   resolution limits for this derivative/wavelength only

RES_PHASE 2.8   high-resolution limit for phasing only
SN_MIN    Set RES_PHASE so that signal-to-noise is bigger than this
SN_RATIO_MIN    Set RES_PHASE so that signal-to-noise is bigger than this
                    ratio times the value at low-resolution

INPHASE             include this wavelength/derivative in phasing.
NOINPHASE           do not this include this derivative/wavelength in phasing

INANO               include anomalous differences for this wavelength/deriv

noanorefine        use anomalous differences in phasing but not
                   refinement for this derivative.
                   (this is usually the best option for MIR
                   unless your anomalous differences are really
                   big, as from a synchrotron MIR dataset at an
                   absorption edge).  Note: you still have to specify
                   for each derivative "inano" to include anomalous
                   differences for that derivative.

anorefine          For this derivative with "inano" specified, use
                   anomalous differences in both refinement and phasing.
                   This is best for MAD data. (This is the default also).
                   Applies to current derivative/wavelength


ATOMNAME XXX      Start reading in information about a new site with atomname
                     "XXX" at the current wavelength/deriv.

OCCUPANCY xx       occupancy of current atom
BVALUE xx          B-value of current atom
XYZ  xx yy zz      fractional coordinates of current atom
REFINEALL          refine x,y,z,occ and b for this atom
REFINENONE         don't refine anything for this atom

Note: Flags for refinement of a heavy atom do not apply when the keyword
SOLVE is used (only with HEAVY)


PDB_XYZ_IN        PDB file with orthogonal A coordinates
                 of all heavy-atoms for this derivative/wavelength
                 NOTE: you cannot use XYZ/OCC/BVALUE/REFINEMENT parameters along with
pdb_xyz_in.
```

```
NRES   n            # of residues in asymmetric unit  [default=100]

NANOMALOUS n        # of anomalously scattering atoms in asymmetric unit.
                    Used to estimate how big the Fa values might be.
                    Also used to set max # of heavy atoms if nsolsite is not set

nobayes        do not use Bayesian correlated phasing in SOLVE.

ntopfour xx         Number of Fourier peaks to pick from a map

ntopderiv xx        Number of Fourier peaks to be tested for
                    inclusion in the model

nsolsite xx         Maximum number of sites in a derivative unless overridden by
                    nsolsite_deriv

nseedtest xx        Number of seeds per derivative to try (before
                    sorting)

nseedsolve xx       Number of seeds (total) to try after
                    sorting them

ntopsolve xx        Number of solutions to print out at the end
                    and number of solutions to keep track of at
                    any one time

addsolve           Add on to solution that is input
                    [default=off]
checksolve         Compare all solutions to input solution
                    [default=off]

analyze_solve      Analyze input solution without doing anything
                    else [default=off]

[no]delete          do [not] check out all solutions by testing
                    all one-site deletions [default=delete]

[no]inverse         do [not] check out all solutions by testing
                    their inverses (does not apply if a solution
                    is centrosymmetric or if anomalous differences
                    are not used). [default=inverse]

SCORING_TABLE (8 values)  Scoring table (usually generated by SOLVE) consisting
                      of mean and standard deviation of scores for trial
                      solutions for Pattersons, Cross-fouriers, Native Fourier
                      maps, and mean figure of merit.  This keyword is useful
                      when you are running SOLVE after modifying the script
                      file it writes out at the end.

QUICK                 once a plausible solution is found, don't keep looking,
                      just add on sites to it and check it at the end. [default]
```

THOROUGH	keep looking anyways until a limit set by ntopsolve, nseedsolve, etc is reached.
NTOL_SITE	a site within ntol_site grid units of an existing site is considered to be a duplicate and is ignored. [default=8]
NTOL_SOLN	a heavy-atom solution for which every site matches another solution within ntol_soln grid units is considered to be a duplicate and is ignored. [default=2]
ACCEPTANCE xx	the weighting function for scoring patterson and free-difference fourier peak heights is adjusted so that a new site with height relative to the previous average height of ACCEPTANCE or higher will generally give a solution with a higher score than the solution without this site. [default =0.2]
SN_MIN xx	Identify working resolution as the point where signal-to-noise in the data goes down to about XX. Default =0.5
SN_RATIO_MIN xx	Identify working resolution as the point where signal-to-noise in the data goes down to about XX times its maximum value. Larger of value of S/N obtained by SN_MIN and SN_RATIO_MIN used. Default = 0.1
NO_SIM	Do not use Sim weighing with heavy-atom structure factors in SAD

```
# solve.com  -- take 1 mad dataset and solve it
# version that reads ccp4 mtz data file
#
setenv CCP4_OPEN UNKNOWN
solve <<EOD
logfile solve.logfile            ! write out most information to this file.
resolution 20  3.0               ! you need resolution.  Cell params read
                                 ! from the mtz file.

! get the data from ccp4 file mad_fbar-cad.mtz:
! NOTE:  do not put in FP=FP SIGFP=SIGFP because SOLVE will assume it is MIR!
! Also you can EITHER use labin/hklin  OR readccp4_unmerged but not both.

labin FPH1=FPH1 SIGFPH1=SIGFPH1 DPH1=DPH1 SIGDPH1=SIGDPH1
labin FPH2=FPH2 SIGFPH2=SIGFPH2 DPH2=DPH2 SIGDPH2=SIGDPH2
labin FPH3=FPH3 SIGFPH3=SIGFPH3 DPH3=DPH3 SIGDPH3=SIGDPH3
hklin mad_fbar-cad.mtz
! scaled data is all ready

mad_atom se
lambda 1
label set 1 with 2 se atoms, lambda 1
wavelength .9782                       ! wavelength value
fprimv_mad  -10                        ! f' value at this wavelength
fprprv_mad  3                          ! f doubleprime value
ATOMNAME Se                            ! we're about to enter data on an
atom

lambda 2
wavelength 0.977865
fprimv_mad  -7.5
fprprv_mad  5

lambda 3
wavelength 0.8856
fprimv_mad  -2
fprprv_mad  3.5
nres 80
nanomalous 2
!
analyze_mad
solve
!--------all done----------
EOD
```

```
#  command file to solve a 2-deriv MIR dataset using ccp4 mtz input
#  It looks just like one_mir_dataset.html except it has LABIN and HKLIN
#  and it has no rawnativefile or rawderivfile and no scale_native or
#  scale_derivative


# CCP4_OPEN environmental variable set to UNKNOWN so file overwriting will work
setenv CCP4_OPEN UNKNOWN

solve <<EOD
!   ccp4 mtz file input
!   solve a 2-deriv MIR dataset
logfile mir.logfile            ! write out most information to this file.
                               ! summary info will be written to solve.prt
resolution 20 3                ! you need resolution.  space group and cell
                               ! dimensions read from mtz file.
!  get the mtz file information and read it in:
labin FP=FP SIGFP=SIGFP FPH1=FPH1 SIGFPH1=SIGFPH1 DPH1=DPH1 SIGDPH1=SIGDPH1
labin FPH2=FPH2 SIGFPH2=SIGFPH2 DPH2=DPH2 SIGDPH2=SIGDPH2
hklin mir_fbar-cad.mtz
!  now we're ready with scaled data

derivative 1                   ! about to enter information on derivative #1
label deriv 1 HG               ! a label for this deriv
atomname hg
inano                          ! You need to tell it if anomalous diffs are
                               ! to be used
noanorefine                     ! use anomalous differences in phasing
                                ! but not refinement (best option for MIR)
nsolsite_deriv 2                ! max 2 sites this deriv

derivative 2
label deriv 2 also hg
atomname hg
inano
noanorefine                     ! use anomalous differences in phasing
                                ! but not refinement (best option for MIR)

acceptance 0.35                 ! accept a new site if it has a
                                ! peak height about 1/3 of avg or more
nsolsite 2                      ! number of sites per deriv
                                ! (use nsolsite_deriv to set individual
values)
ANALYZE_MIR                     ! analyze this mir data and set up for SOLVE
SOLVE
EOD
```

Contents	Index

What to do next after you've run SOLVE

After you have run SOLVE you should first:

- Have a look at how to interpret your output

If SOLVE successfully solved your structure you might:

- look at your solve.ezd map directly with Alwyn Jones' "O" program (www.imsb.au.dk/~mok/o/) or your equivalent solve.ccp4_map with CCP4.\
- go on to density modification and automated model-building with RESOLVE
- Check out how to export your output to other packages

If SOLVE got most of the heavy atom solution you can:

- analyze variations on your solution with ANALYZE_SOLVE
- try and add more sites using ADDSOLVE

and if SOLVE was not able to find a good solution you should:

- See if you should rerun SOLVE with different control parameters
- If you have a MAD dataset...try SOLVE with just the peak wavelength and the SAD script.

ANALYZE_SOLVE: Analyzing a known solution with SOLVE
ADDSOLVE: Getting more heavy atom sites if you have a partial solution.

You can use ANALYZE_SOLVE or ADDSOLVE to input some sites you have obtained already. Here is the easiest way to do it:

- In your solve.com file, under "lambda 1" or "derivative 1" specify
  - atomname xx ! xx is your atom
  - xyz x y z ! coordinates of this atom
  - .. more xyz values
- anywhere ***before*** ANALYZE_MAD or ANALYZE_MIR or SAD specify:
  - ANALYZE_SOLVE (refine input sites) or ADDSOLVE (start from input sites)
- SOLVE will refine these sites and either just phase with them (ANALYZE_SOLVE) or go on

and find more sites (ADDSOLVE)
- See the sample solve.com file for ANALYZE_SOLVE and the one for ADDSOLVE for examples.

There is another way to do the same thing that is more complicated, but does not require re-running scaling and analysis. If you have already run SOLVE, you can also use SOLVE to analyze a MAD or MIR solution that you have already obtained. Start with your solve_mad.script file or solve_mir.script file that was written out during automated SOLVE operation by the routine ANALYZE_MAD or ANALYZE_MIR and:

- edit it to include the heavy atom sites that you want to check. For MAD data, you only need to specify the heavy atom sites for the one wavelength designated as the standard wavelength (this is indicated in your solve_mad.script file by the lines "This is the "standard" wavelength from MADMRG" and "Put your heavy atom parameters here"), they are copied automatically to the others. For more info on the keywords in the solve_mad.script and solve_mir.script files see the writeups for solve_mad.script (MAD data) or solve_mir.script (MIR data)
- copy the 1-line scoring table from your phases-hl.script output file into your solve_mad.script or solve_mir.script file. The keyword is scoring_table.
- Then specify the following keywords:

  ```
  ANALYZE_SOLVE
  SOLVE
  ```

- then move all your ouput files from previous runs to somewhere safe so that SOLVE does not overwrite them
- then run solve using the edited solve_mad.script or solve_mir.script file as input

The program will read in these heavy atom parameters and will refine them. It will do a phasing calculation and will analyze the derivatives relative to the pattersons and with cross-difference Fouriers. This analysis is the same as you would get at the end of SOLVE ordinarily. (SOLVE will also write out the files "phases-hl.export; .drg; .script as it does when running originally. Be sure you move your original ones out of the way or use the keywords "EXPORTFILE", "PHASEFILE", and "NEWSCRIPTFILE" to define new filenames for these output files so SOLVE does not overwrite them.)

SOLVE can be used to generate additional sites if you have a partial heavy atom solution. You do this in exactly the same way as described above for ANALYZE_SOLVE except you substitute ADDSOLVE for ANALYZE_SOLVE. Then run SOLVE. It will read in your solution and generate other solutions that are based on it and then it will analyze them for you and write out the best ones. This is a good thing to do once SOLVE has found a good, but not complete solution for you. You may need to increase the number of sites allowed in a solution (NSOLSITE) to allow more sites to be found. You may also want

to reduce the value of "ACCEPTANCE" so as to include weak sites.

If you use ADDSOLVE, then you might also wish to specify the keyword "CHECKSOLVE". If you do this, then the analysis routine will compare each solution you obtain to the one you input at the beginning. This way you can see how things are going.

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
#
#  command file to use addsolve on a SAD dataset
#
#
solve<<EOD > solve.log
logfile solve.logfile

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym


readformatted        ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged            ! premerged/ unmerged
read_intensities     ! read_intensities/read_amplitudes
fixscattfactors      ! fixscattfactors/refscattfactors

addsolve        ! look for more sites then refine and phase

mad_atom se

lambda 1
label SAD data for Pt
rawmadfile lam1.intensities
fprprv_mad 5.0
xyz   0.9770      0.2439      1.1071   ! put sites here
```

```
xyz   1.1877     0.2625     0.9833
xyz   1.1669     0.2943     0.9205


nsolsite_deriv 4            ! 4 atoms max
SAD                         ! solve an SAD dataset
EOD
#
#
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
#
#  command file to use analyze_solve on a SAD dataset
#
#
solve<<EOD > solve.log
logfile solve.logfile

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym


readformatted        ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged            ! premerged/ unmerged
read_intensities     ! read_intensities/read_amplitudes
fixscattfactors      ! fixscattfactors/refscattfactors

analyze_solve        ! do not look for sites...just refine and phase

mad_atom se

lambda 1
label SAD data for Pt
rawmadfile lam1.intensities
fprprv_mad 5.0
xyz   0.9770      0.2439      1.1071   ! put sites here
```

```
xyz   1.1877     0.2625     0.9833
xyz   1.1669     0.2943     0.9205

refinenone    !  put this and the next line here to fix xyz
refineoccb    !


nsolsite_deriv 4            ! 4 atoms max
SAD                         ! solve an SAD dataset
EOD
#
```

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
#
#  generate SAS dataset (ano scatterer in native)
#
solve  <<EOD

! generate Se SAD dataset

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym

logfile generate.logfile
solvefile generate.prt
coordinatefile coords.pdb
percent_error  0.5
iranseed 31933

mad_atom Pt
lambda 1
label 0.9797 A Se SAD dataset
wavelength 0.9797
fprimv_mad  -8              ! f' value at this wavelength
fprprv_mad  10
ATOMNAME Pt
xyz  0.44 0.16 0.38
occ 1.0
bvalue 20
ATOMNAME Pt
xyz  0.14 0.36 0.28
occ 1.0
bvalue 20
generate_mad
! now lam_1.intensities = SAS intensity data at 0.9797 A
end
EOD
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
#
#  command file to solve an SAD dataset (Pt 1-wavelength)
#
#
solve<<EOD > solve.log
logfile solve.logfile

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym


readformatted       ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged           ! premerged/ unmerged
read_intensities    ! read_intensities/read_amplitudes
fixscattfactors     ! fixscattfactors/refscattfactors

mad_atom se

lambda 1
label SAD data for Pt
rawmadfile lam1.intensities
fprprv_mad 5.0

nsolsite_deriv 2            ! 2 atoms max
SAD                         ! solve an SAD dataset
```

```
EOD
#
#  Now do density modification and build a model:
resolve<<EOD>resolve.log
!solvent_content 0.40
seq_file seq.dat        ! sequence file
EOD
#  All done; your output phases are in resolve.mtz and
#   your model is in resolve.pdb
#
```

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*****************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*****************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 6)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: SOLVE 06-Apr-05

Space group number is: 5
Space group name from file name is: c2
Rescaling standard dataset to put it on approximate absolute scale. NRES =
200; <F**2> expected = 196000.0
; <F**2> observed in lowest resolution shell = 489643.2
... Scale factor = 0.4002915

*** Analysis of this scaled MAD data set ***

Fbar,sigma,Delano,sigma for 1 wavelengths written to:
mad_fbar.scl

F+,sigma,F-,sigma for 1 wavelengths written to:
mad_fpfm.scl

** Completeness of Fbar data at each wavelength: **

Completeness of dataset 1 (F > 2.000000 * sigma)
SAD data for Pt

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	235	98.3
2	4.500	317	316	99.7
3	4.200	120	119	99.2
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	142	99.3
8	3.300	172	171	99.4
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1814	99.6

** R-factors for F-bar data dispersive differences **

Anomalous differences lambda 1 (f" = 5.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	235	307.724	0.079	0.998	27.96	13.15	2.13
2	4.500	314	301.800	0.073	1.005	25.56	11.90	2.15
3	4.200	119	289.654	0.074	1.001	24.07	11.72	2.05
4	3.975	120	251.377	0.080	0.998	23.64	10.10	2.34
5	3.750	155	249.422	0.079	0.998	23.61	10.06	2.35
6	3.600	111	225.244	0.088	0.999	23.11	9.07	2.55
7	3.450	142	192.784	0.101	0.999	23.12	7.76	2.98
8	3.300	171	190.314	0.089	1.000	19.87	7.94	2.50
9	3.150	189	186.536	0.084	1.000	18.17	7.34	2.48
10	3.000	253	177.901	0.083	1.001	17.32	7.09	2.44
Total:		1809	240.774	0.081	1.000	22.94	10.00	2.37

Recommended resolution cut-off = 3.00

SAD analysis: defining pseudo-native based on Fbar for the SAD data
and pseudo-derivative based on Fbar and del anom for the SAD data

Script file suitable for running SOLVE written to:
solve_mir.script

**** SOLVE: Solutions to MIR or SIR datasets ******

Derivatives considered: 1 (NSET)
Cross-vectors tested in HASSP: 20 (ICRMAX, DEFAULT=20)
HASSP solutions saved per deriv: 30 (NTOPHASSP, DEFAULT=30)

```
Fourier peaks saved per map:        30  (NTOPFOUR, DEFAULT=10)
Sites per derivative:               4  (NSOLSITE, DEFAULT=20)
Derivative solutions per seed:       5  (NTOPDERIV, DEFAULT=5)
Seeds per derivative tested:        10  (NSEEDTEST,DEFAULT=10)
Sorted seeds to use                  5  (NSEEDSOLVE, DEFAULT=5)
Number of final solutions saved:     5  (NTOPSOLVE, DEFAULT=5)




 Sites per derivative vary with derivative.
Derivative   Max sites
    1            2


Solutions obtained will be compared to input solution (ICHECKSOLVE)
Correlated phasing used (CORRELPHASE)
Standard difference fouriers will be calculated for derivative           1

Datafile with           4 columns of data:
Title:mad_fbar.scl Fnat,sig,(fbar,sig,delano,sig)n
Data: Fbar              SAD data for Pt
Data: Sig of Fbar        SAD data for Pt
Data: Del Ano (F+ - F-)   SAD data for Pt
Data: sig of Del Ano      SAD data for Pt

Fnat,sigma taken from columns          1          2
Fder,sig,Delano,sig deriv           1 from cols:          1          2
         3          4

Check solution to be compared to all solutions found:

Derivative           1:
 Site    X       Y      Z
   1   0.140   0.360   0.280
   2   0.440   0.160   0.380

*********************************************************

ANALYZE_SOLVE:  analysis of top          1 solutions



*************************************************************


Solution          1

********************** Analysis of this solution *************

 ******    Analysis of non-randomness of native Fourier map ******

A.  Maps with distinct solvent regions havea high standard deviation
of local r.m.s. electron density.
```

For this map the SD of this local r.m.s. is 0.2269217

B. Maps with distinct solvent regions also have a high correlation
of local r.m.s. electron density with
density at neighboring locations. Typical
values for poor maps in this
structure solution are 3.6363978E-02 +/- 2.8206760E-02
For this map the correlation of r.m.s. density
in neighboring boxes is 0.1060753
The correlation coefficient is used here in scoring.

Skew of the map is: 0.1197042

****** Analysis of derivative solutions with the difference Patterson ******
and with cross-validation difference Fouriers

Derivative # 1

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.141	0.361	0.281	76.859
2	0.438	0.167	0.385	69.500

Evaluation of this test soln with 2 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.281	0.000	-0.562	10586.4	11814.6	2

Cross-vectors for sites 2 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.297	-0.194	0.104	8196.26	5341.68	1
2	-0.578	-0.194	-0.667	7865.75	5341.68	1

Cross-vectors for sites 2 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.875	0.000	-0.771	8177.69	9660.44	2

Overall quality of this Patterson soln = 10019.1
Overall quality of the fit to patterson = 1.90927
Avg normalized peak height = 4480.66

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

```
 Site    x       y       z       occ       B     -- PEAK  HEIGHT --

   1   0.140   0.360   0.280   1.230  25.260            8.96
   2   0.440   0.162   0.381   1.040  16.937            8.92

Summary of scoring for this solution:
                        -- over many solutions--    -- this solution --
Criteria                    MEAN         SD        VALUE       Z-SCORE
Pattersons:                  3.61        1.74        7.47        2.21
Cross-validation Fourier:    4.91        5.00        14.9        2.00
NatFourier CCx100:           3.64        2.82        10.6        2.47
Mean figure of meritx100:   0.000E+00    5.82        49.1        8.44
Correction for Z-scores:                                        -3.33

Overall Z-score value:                                           11.8


****** Overall analysis of phasing for solution           1************


HEAVY:  Refine heavy atom parameters

File title: SOLVE     06-Apr-05
CRYSTALLOGRAPHIC PARAMETERS

    A =  76.00     B =  28.00     C =  42.00     alpha =  90.00     beta = 103.00     gamma =  90.00

PHASES CALCULATED EVERY   5 DEGREES
RESIDUALS CALCULATED ON EXTRA ZEROTH CYCLE ONLY
SIGMAS FROM data FILE WILL BE USED
STATISTICS WILL BE PRINTED ON ZEROTH CYCLE, SHIFTS ON LAST
PHASING WILL BE DONE TAKING INTO ACCOUNT THE CORRELATIONS AMONG DERIVATIVES
THE GROUPS OF DERIVATIVES WITH CORRELATIONS WILL BE UPDATED
THE BETA VALUES FOR EACH DERIV  WILL BE SET TO 1.0
PHASE-AVERAGED RESIDUALS WILL BE USED FOR PHASING

TYPE OF REFINEMENT SELECTED:
UNPHASED ORIGIN-REMOVED  PATTERSON REFINEMENT ONLY

Bayesian Correlated Phasing will be used

RESOLUTION LIMITS IN ANGSTROMS:      3.000    20.000
MINIMUM RATIO OF FDER TO RMS LACK-OF-CLOSURE FOR  INCLUSION IN REFINEMENT OR PHASING=      0.000
MINIMUM NATIVE F:      0.000
MINIMUM FIGURE OF MERIT FOR PHASED REFINEMENT:      0.000
MINIMUM ALLOWED ISOTROPIC B:      0.000
PARAMETER SHIFTS GREATER THAN     0.0000 TIMES SIGMA  WILL BE SCALED BY     0.5000
MINIMUM RATIO OF FNAT/SIGMA OR FDER/SIGMA TO INCLUDE:      1.000

NUMBER OF REFINEMENT CYCLES IS      2
DERIVATIVES REFINED DURING THESE CYCLES ARE :  0  0
TYPE OF OUTPUT SELECTED IS: +10 COLUMNS OF HENDRICKSON-LATTMAN COEFFICIENTS
1 INPUT data FILE WITH   4 COLUMNS IS:
mad_fbar.scl
```

```
COLUMN  0 : mad_fbar.scl Fnat,sig,(fbar,sig,delano,sig)n
COLUMN  1 : Fbar              SAD data for Pt
COLUMN  2 : Sig of Fbar       SAD data for Pt
COLUMN  3 : Del Ano (F+ - F-)   SAD data for Pt
COLUMN  4 : sig of Del Ano      SAD data for Pt


data COLUMNS FOR NATIVE F AND SIGMA:     1    2
data COLUMNS FOR BEST AND MOST PROB PHASES AND FIGURE OF MERIT:     0    0    0
OVERALL SCALE FACTOR FOR ALL data =     1.000
SCALE FACTOR FOR NATIVE SIGMAS =     1.000


DERIVATIVE INFORMATION FOR  1 COMPOUNDS


COMPOUND 1     SAD data for Pt
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)    1    2    3    4
THIS DERIVATIVE WILL BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
ONLY ANO DIFFERENCES WILL BE USED IN REFINEMENT AND PHASING FOR THIS DERIVATIVE.
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

Sim weighting on heavy-atoms used assuming FO/F" =    6.478219
CARRYING OUT STANDARD REFINEMENT
Total of           2 cycles will be done
Derivs refined will be           0          0

SUMMARY OF RESULTS ON FINAL CYCLE:

NUMBER OF REFLECTIONS READ =    1814
NUMBER OF F .GT. FMIN      =    1814
NUMBER OF F IN RES. LIMITS =    1814
NUMBER OF F .GT. MIN <M>   =    1798
NUMBER OF F USED TO REFINE =       0



FIGURE OF MERIT <       0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1.0
# OF REFLECTIONS       221   152   138   160   155   143   188   191   278   187

FIGURE OF MERIT WITH RESOLUTION
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
N:             1813    104    150    202    218    247    277    296    319
MEAN FIG MERIT:   0.52   0.46   0.51   0.51   0.50   0.53   0.53   0.55   0.54
```

```
COMPOUND  1          SAD data for Pt
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

ANOM DIFFS:      1555.    72.   115.   171.   189.   211.   244.   263.   290.
RMS OBS DIFF:     26.2   36.4   31.9   30.1   28.5   27.3   25.3   21.1   19.8
RMS CALC DIFF:    23.1   33.0   28.5   26.1   24.6   23.6   22.3   18.6   18.1
RMS RESIDUAL:     11.1   14.0   12.5   14.0   13.3   11.6   10.1    8.8    8.0
RATIO ISO/ANO:    0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:            0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
ANOMALOUS LOC:          0.0    5.3    7.1    5.8    5.8    5.5    3.8    3.5
RMS FPH :             431.6  319.1  343.3  338.2  284.9  240.7  223.1  202.9
RMS FH  :               0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS SIGMA:             10.8    8.0    8.6    8.5    7.1    6.0    5.6    5.1


PARAMETER SHIFTS FOR DERIV  1 : SAD data for Pt

                    SCALE FACTOR     OVERALL B
CURRENT VALUES:         1.0000         0.0000



                SITE  ATOM      OCCUP     X       Y       Z        B


CURRENT VALUES:     1    Se      1.2302  0.1398  0.3600  0.2805   25.2597

CURRENT VALUES:     2    Se      1.0403  0.4400  0.1619  0.3808   16.9368



*****************************************************************
*****************************************************************


*** Summary of solutions and their relationships
 to each other and to check solution ***


-------------------------------------------------------------
solution #          1 with overall quality =   11.79113
Derivative           1 with           2 sites.
Overall scale =    1.000000     and  overall b of   0.0000000E+00
 0.1397693      0.3600000      0.2804537       1.230202       25.25968
 0.4400216      0.1619363      0.3808324       1.040293       16.93677
Best match of solution 1 -> solution 2:

-------- solution 1 --------    -------------solution 2 ------
site     x        y        z    site     x        y        z     DIST (A)
Derivative  1
```

1	0.140	0.360	0.280	1	0.140	0.360	0.280	0.03
2	0.440	0.162	0.381	2	0.440	0.160	0.380	0.06

Comparison of this solution with check solution:
Number of sites in this solution matching check= 2
... and number not matching = 0
by derivative, this is...

Deriv	nsame	ndifferent
1	2	0

All sites in this solution are contained in check soln

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
#
#
#  generate 1 mad dataset
#
solve <<EOD

!  GENERATE 1 MAD DATASET:

@solve.setup
percent_error 0.5
coordinatefile coords.pdb
iranseed -199753
logfile generate.logfile
solvefile generate.prt

mad_atom se                          ! define the scattering factors...
!
lambda 1
label set 1 with 2 se atoms, lambda 1
wavelength 0.9782            ! wavelength value
fprimv_mad  -10             ! f' value at this wavelength
fprprv_mad  3
ATOMNAME Se
xyz  0.44 0.16 0.38
occ 1.5
bvalue 20.
ATOMNAME Se
xyz  0.23 0.45 0.165
occ 1.5
bvalue 20.
!
lambda 2
wavelength 0.977865
fprimv_mad  -7.5
fprprv_mad  5
```

```
lambda 3
wavelength 0.8856
fprimv_mad  -2
fprprv_mad  3.5
!
GENERATE_MAD                    ! generate the MAD dataset now.
!
EOD
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
#   solve a MAD dataset
#
solve<<EOD > solve.log
logfile solve.logfile

resolution 20 3.0
cell 60 60 50 90 90 90
symfile /usr/local/lib/solve/p21212.sym


readformatted       ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged           ! premerged/ unmerged
read_intensities    ! read_intensities/read_amplitudes
refscattfactors     ! fixscattfactors/refscattfactors

mad_atom se     ! the anomalously-scattering atom
lambda 1
label set 1 with 2 se atoms, lambda 1
wavelength .9782                    ! wavelength value
fprimv_mad  -9                      ! f' value at this wavelength
fprprv_mad  3.5                     ! f doubleprime value
rawmadfile lam1.intensities         ! datafile for this wavelength
lambda 2
wavelength 0.977865
fprimv_mad  -8
fprprv_mad  4.5
```

```
rawmadfile lam2.intensities

lambda 3
wavelength 0.8856
fprimv_mad  -2.5
fprprv_mad  3.0
rawmadfile lam3.intensities
nres 80                    ! number of residues in au
nanomalous 2               ! number of anomalously-scattering atoms in au
!
scale_mad
analyze_mad
solve
EOD
#
# Now run Resolve to do density modification and build a model
#
resolve << EOD > resolve.log
!solvent_content 0.70       !    solvent fraction
seq_file seq.dat            !    sequence file
EOD
#
#  That's it! Now resolve.mtz has your updated phases
#  and resolve.pdb has your model
#
```

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*****************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*****************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 6)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: SOLVE 06-Apr-05

Space group number is: 18
Space group name from file name is: p21212
Rescaling standard dataset to put it on approximate absolute scale. NRES = 80; <F**2> expected = 39200.00
; <F**2> observed in lowest resolution shell = 321198.7
... Scale factor = 0.1220428

*** Analysis of this scaled MAD data set ***

Fbar,sigma,Delano,sigma for 3 wavelengths written to:
mad_fbar.scl

F+,sigma,F-,sigma for 3 wavelengths written to:
mad_fpfm.scl

** Completeness of Fbar data at each wavelength: **

Completeness of dataset 1 (F > 2.000000 * sigma)
set 1 with 2 se atoms, lambda 1

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	515	514	99.8
2	4.500	682	682	100.0
3	4.200	263	263	100.0
4	3.975	257	257	100.0
5	3.750	305	305	100.0
6	3.600	267	267	100.0
7	3.450	307	307	100.0
8	3.300	354	354	100.0
9	3.150	416	416	100.0
10	3.000	511	511	100.0
total		3877	3876	100.0

Completeness of dataset 2 (F > 2.000000 * sigma)
set 2

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	515	513	99.6
2	4.500	682	682	100.0
3	4.200	263	263	100.0
4	3.975	257	257	100.0
5	3.750	305	304	99.7
6	3.600	267	267	100.0
7	3.450	307	307	100.0
8	3.300	354	353	99.7
9	3.150	416	416	100.0
10	3.000	511	511	100.0
total		3877	3873	99.9

Completeness of dataset 3 (F > 2.000000 * sigma)
set 3

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	515	515	100.0
2	4.500	682	682	100.0
3	4.200	263	263	100.0
4	3.975	257	255	99.2
5	3.750	305	305	100.0
6	3.600	267	267	100.0
7	3.450	307	306	99.7

8	3.300	354	354	100.0
9	3.150	416	415	99.8
10	3.000	511	507	99.2
total		3877	3869	99.8

** R-factors for F-bar data dispersive differences **

Dispersive differences lambda 2 - lambda 1
(Delta f-prime = 1.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	507	127.465	0.028	1.000	2.42	4.07	0.59
2	4.500	672	108.986	0.027	1.001	2.01	3.21	0.63
3	4.200	261	96.516	0.027	1.000	1.48	2.83	0.52
4	3.975	254	88.763	0.030	1.000	2.07	2.61	0.79
5	3.750	302	83.108	0.029	1.000	1.80	2.45	0.74
6	3.600	264	73.858	0.027	1.000	1.36	2.17	0.63
7	3.450	305	69.579	0.030	1.000	1.63	2.04	0.80
8	3.300	350	67.112	0.029	1.000	1.42	1.93	0.73
9	3.150	410	62.170	0.030	1.001	1.52	1.84	0.83
10	3.000	507	60.430	0.033	1.000	1.85	1.77	1.04
Total:		3832	86.387	0.029	1.000	1.84	2.69	0.73

Recommended resolution cut-off = 3.00

Dispersive differences lambda 3 - lambda 1
(Delta f-prime = 6.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	514	130.889	0.060	1.000	8.72	4.26	2.05
2	4.500	678	110.458	0.058	1.001	7.20	3.27	2.20
3	4.200	263	97.259	0.056	1.001	6.01	2.83	2.12
4	3.975	254	89.277	0.060	1.000	6.05	2.61	2.32
5	3.750	304	83.442	0.062	1.000	5.90	2.43	2.43
6	3.600	266	74.648	0.064	0.999	5.53	2.20	2.52
7	3.450	306	70.124	0.069	0.999	5.55	2.04	2.71
8	3.300	352	67.457	0.065	0.999	4.99	1.95	2.56
9	3.150	414	62.938	0.067	1.000	4.99	1.86	2.68
10	3.000	507	60.673	0.070	0.999	5.03	1.78	2.83
Total:		3858	87.523	0.062	1.000	6.30	2.75	2.43

Recommended resolution cut-off = 3.00

Dispersive differences lambda 3 - lambda 2
(Delta f-prime = 5.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	511	129.510	0.043	0.999	5.67	4.22	1.34
2	4.500	675	109.336	0.042	1.000	4.97	3.26	1.53
3	4.200	262	97.004	0.043	1.001	4.48	2.81	1.59
4	3.975	254	89.282	0.042	1.000	3.78	2.63	1.44
5	3.750	304	83.837	0.046	1.000	4.10	2.45	1.67
6	3.600	263	74.451	0.044	0.999	3.45	2.20	1.57
7	3.450	305	69.758	0.051	0.999	3.86	2.03	1.90
8	3.300	351	67.259	0.048	0.999	3.43	1.96	1.76
9	3.150	414	62.912	0.050	0.999	3.46	1.85	1.87
10	3.000	504	60.419	0.050	1.000	3.30	1.77	1.86
Total:		3843	87.047	0.045	1.000	4.25	2.74	1.65

Recommended resolution cut-off = 3.00

Anomalous differences lambda 1 (f" = 3.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	507	128.084	0.040	1.000	3.75	5.62	0.67
2	4.500	671	109.328	0.043	1.000	4.07	4.50	0.90
3	4.200	257	95.242	0.046	1.000	4.28	3.86	1.11
4	3.975	252	88.207	0.045	1.000	3.50	3.62	0.97
5	3.750	300	82.340	0.051	1.000	4.15	3.34	1.24
6	3.600	262	73.371	0.048	0.999	3.43	3.00	1.14
7	3.450	302	69.115	0.050	1.000	3.34	2.80	1.19
8	3.300	349	67.715	0.049	1.000	3.32	2.73	1.21
9	3.150	405	61.307	0.053	1.001	3.29	2.49	1.32
10	3.000	505	60.529	0.054	1.001	3.32	2.50	1.33
Total:		3810	86.307	0.046	1.000	3.68	3.73	1.09

Recommended resolution cut-off = 3.00

Anomalous differences lambda 2 (f" = 4.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	503	124.687	0.053	1.000	6.97	5.31	1.31
2	4.500	666	108.572	0.059	1.000	7.03	4.46	1.57
3	4.200	256	95.904	0.066	1.000	7.10	3.90	1.82
4	3.975	250	88.554	0.062	1.000	5.85	3.63	1.61
5	3.750	298	82.528	0.071	1.000	6.79	3.35	2.03
6	3.600	262	73.455	0.071	1.000	6.17	3.00	2.06
7	3.450	301	69.000	0.074	1.000	6.04	2.80	2.15
8	3.300	345	67.542	0.071	1.001	5.62	2.72	2.06
9	3.150	407	61.927	0.079	1.001	5.69	2.52	2.26
10	3.000	504	60.379	0.074	1.001	5.15	2.48	2.08
Total:		3792	85.776	0.065	1.000	6.31	3.67	1.86

Recommended resolution cut-off = 3.00

Anomalous differences lambda 3 (f" = 3.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	506	127.124	0.042	1.000	4.50	5.48	0.82
2	4.500	668	108.143	0.046	1.000	4.70	4.45	1.06
3	4.200	259	95.535	0.054	1.000	5.39	3.90	1.38
4	3.975	252	88.221	0.051	1.000	4.70	3.63	1.29
5	3.750	302	82.993	0.056	1.000	5.14	3.37	1.53
6	3.600	263	73.650	0.056	1.000	4.50	2.99	1.51
7	3.450	301	69.191	0.053	1.000	3.70	2.82	1.31
8	3.300	348	67.135	0.057	1.000	3.94	2.72	1.45
9	3.150	408	62.047	0.058	1.000	3.97	2.50	1.59
10	3.000	500	59.542	0.057	1.000	3.61	2.45	1.48
Total:		3807	85.960	0.051	1.000	4.40	3.69	1.30

Recommended resolution cut-off = 3.00

ANALYZE_MAD: Run MADMRG and MADBST on MAD data to get ready for SOLVE

Correlation of anomalous differences at different wavelengths.
(You should probably cut your data off at the resolution where
this drops below about 0.3. A good dataset has correlation
between peak and remote of at least 0.7 overall. Data with

correlations below about 0.5 probably are not contributing much.)

DMIN	CORRELATION FOR WAVELENGTH PAIRS 1 VS 2	1 VS 3	2 VS 3
6.00	0.67	0.60	0.72
4.50	0.67	0.59	0.73
4.20	0.71	0.66	0.81
3.98	0.73	0.65	0.75
3.75	0.75	0.69	0.79
3.60	0.69	0.72	0.74
3.45	0.75	0.72	0.75
3.30	0.81	0.73	0.80
3.15	0.75	0.69	0.79
3.00	0.69	0.61	0.73
ALL	0.71	0.64	0.75

Final refined values of fprime and fdoubleprime

Form factors at lambda = 0.9782
f-prime = -9.40
f" = 3.17

Form factors at lambda = 0.9779
f-prime = -7.41
f" = 5.13

Form factors at lambda = 0.8856
f-prime = -3.43
f" = 3.55

Fa Patterson from MADBST to be written to:
patterson.patt

Script file suitable for running SOLVE written to:
solve_mad.script

Datafile for SOLVE with MADMRG-compressed dataset
("Fnat",sig,"Fder",sig,"Delano",sig,iso diffs, ano diffs,
<Fhcos>, <Fhsin> from MADBST) is:
solve.data

**** SOLVE: Solutions to MIR or SIR datasets ******

Derivatives considered: 3 (NSET)

```
Cross-vectors tested in HASSP:      20  (ICRMAX, DEFAULT=20)
HASSP solutions saved per deriv:    30  (NTOPHASSP, DEFAULT=30)
Fourier peaks saved per map:        30  (NTOPFOUR, DEFAULT=10)
Sites per derivative:               2  (NSOLSITE, DEFAULT=20)
Derivative solutions per seed:      5  (NTOPDERIV, DEFAULT=5)
Seeds per derivative tested:        10  (NSEEDTEST,DEFAULT=10)
Sorted seeds to use                 5  (NSEEDSOLVE, DEFAULT=5)
Number of final solutions saved:    5  (NTOPSOLVE, DEFAULT=5)


Solutions obtained will be compared to input solution (ICHECKSOLVE)
Correlated phasing used (CORRELPHASE)
Patterson map for derivative           2 will be read directly from:
patterson.patt

For derivative           2 the heavy atom
 structure factor components parallel to and
perpendicular to the native structure factor will be read from columns
         9 and          10

Datafile with          10 columns of data:
Title:MADMRG output (cols 1 to  8) and  MADBST fh cos,sin theta (c
Data: madmrg: MOCK FNAT
Data: madmrg: MOCK sig FNAT
Data: madmrg: MOCK FDER
Data: madmrg: MOCK sig FDER
Data: madmrg: MOCK DEL ANO
Data: madmrg: MOCK sig DEL ANO
Data: madmrg: Del iso for Patterson
Data: madmrg: Sigma of del iso for Patterson
Data: <Fa cos theta> = Fa component along Fo weighted by fom
Data: <Fa sin theta> = weighted Fa component perpendicular to Fo

Fnat,sigma taken from columns          1          2
Fder,sig,Delano,sig deriv           2 from cols:           3          4
         5          6

Check solution to be compared to all solutions found:

Derivative           1:
 Site    X       Y      Z

Derivative           2:
 Site    X       Y      Z
   1   0.440   0.160   0.380
   2   0.230   0.450   0.165

Derivative           3:
 Site    X       Y      Z

*********************************************************

ANALYZE_SOLVE:  analysis of top          1 solutions
```

****

Solution 1

********** Analysis of this solution **********

****** Analysis of non-randomness of native Fourier map ******

A. Maps with distinct solvent regions havea high standard deviation
of local r.m.s. electron density.
For this map the SD of this local r.m.s. is 0.4349099

B. Maps with distinct solvent regions also have a high correlation
of local r.m.s. electron density with
density at neighboring locations. Typical
values for poor maps in this
structure solution are 0.1916740 +/- 0.1049155
For this map the correlation of r.m.s. density
in neighboring boxes is 0.4667333
The correlation coefficient is used here in scoring.

Skew of the map is: 0.7527540

****** Analysis of derivative solutions with the difference Patterson ******
and with cross-validation difference Fouriers

Derivative # 2

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.438	0.160	0.382	80.612
2	0.229	0.451	0.167	80.552

Evaluation of this test soln with 2 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.875	-0.319	0.000	10439.7	12996.5	2
2	-0.375	0.500	-0.764	11707.2	12996.5	2
3	0.500	0.181	-0.764	11883.2	12996.5	2

Cross-vectors for sites 2 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

1	-0.208	0.292	-0.215	11662.7	6493.43	1
2	-0.667	-0.611	-0.215	11140.0	6493.43	1
3	-0.167	0.792	-0.549	11392.1	6493.43	1
4	0.292	-0.111	-0.549	11677.4	6493.43	1

Cross-vectors for sites 2 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.458	-0.903	0.000	11677.5	12977.3	2
2	0.042	0.500	-0.333	11344.2	12977.3	2
3	0.500	-0.403	-0.333	10936.5	12977.3	2

Overall quality of this Patterson soln = 19766.4
Overall quality of the fit to patterson = 3.26333
Avg normalized peak height = 5959.79

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.440	0.159	0.380	0.603	26.069	20.25
2	0.230	0.450	0.165	0.549	20.622	19.75

Summary of scoring for this solution:

	-- over many solutions--		-- this solution --	
Criteria	MEAN	SD	VALUE	Z-SCORE
Pattersons:	4.98	2.23	9.93	2.22
Cross-validation Fourier:	28.2	10.1	33.3	0.506
NatFourier CCx100:	19.2	10.5	46.7	2.62
Mean figure of meritx100:	0.000E+00	9.20	80.4	8.74
Correction for Z-scores:			-4.97	

Overall Z-score value: 9.11

****** Overall analysis of phasing for solution 1**********

*** Re-estimation of scattering factors by refinement of occupancies using ***
dispersive and anomalous differences.

Estimation of scattering factors at each wavelength
by refinement of occupancies relative to those found from the
initial refinement carried out with data from MADMRG.

Refining iso occupancies for iso diffs lambda 2 - lambda 1

Results of refinement:
Ratio of occupancies to standard refinement: 1.249 +/- 0.048
Delta f-prime based on input f-prime values: 1.986
New estimate of delta f-prime: 2.481 +/- 0.095

with sign of: 1. and Z of 39.7

Refining iso occupancies for iso diffs lambda 3 - lambda 1

Results of refinement:
Ratio of occupancies to standard refinement: 1.051 +/- 0.040
Delta f-prime based on input f-prime values: 5.973
New estimate of delta f-prime: 6.278 +/- 0.236
with sign of: 1. and Z of 55.6

Refining iso occupancies for iso diffs lambda 3 - lambda 2

Results of refinement:
Ratio of occupancies to standard refinement: 1.045 +/- 0.006
Delta f-prime based on input f-prime values: 3.987
New estimate of delta f-prime: 4.166 +/- 0.026
with sign of: 1. and Z of 51.5

Refining ano occupancies for ano diffs lambda 1

Results of refinement:
Ratio of occupancies to standard refinement: 0.899 +/- 0.024
f" value based on input values: 3.167
New estimate of f": 2.849 +/- 0.075

Refining ano occupancies for ano diffs lambda 2

Results of refinement:
Ratio of occupancies to standard refinement: 0.881 +/- 0.034
f" value based on input values: 5.134
New estimate of f": 4.522 +/- 0.174

Refining ano occupancies for ano diffs lambda 3

Results of refinement:
Ratio of occupancies to standard refinement: 0.919 +/- 0.039
f" value based on input values: 3.547
New estimate of f": 3.261 +/- 0.137

Fitting f-prime values.

Restraints:

Lambda	Target f-prime	final f-prime	weight
1	-9.400	-9.811	0.001

2	-7.414	-7.478	0.001
3	-3.427	-3.427	0.001

Delta-fprime targets:

Lambda i	j	target delta-fprime	final delta-fprime	wgt
2	1	2.481	2.333	39.74
3	1	6.278	6.384	55.62
3	2	4.166	4.051	51.52

Residual for restraints: 0.13144E-01
Residual for targets: 1.4718

Final refined values of f-prime and f"

Wavelength	------- f-prime --------		--------f"----------	
	last refinement	Refined	last refinement	Refined
1	-9.400	-9.811	3.167	2.849
2	-7.414	-7.478	5.134	4.522
3	-3.427	-3.427	3.547	3.261

*** Done with re-estimation of scattering factors ***

HEAVY: Refine heavy atom parameters

File title: SOLVE 06-Apr-05
CRYSTALLOGRAPHIC PARAMETERS

A = 60.00 B = 60.00 C = 50.00 alpha = 90.00 beta = 90.00 gamma = 90.00

PHASES CALCULATED EVERY 5 DEGREES
RESIDUALS CALCULATED ON EXTRA ZEROTH CYCLE ONLY
SIGMAS FROM data FILE WILL BE USED
STATISTICS WILL BE PRINTED ON ZEROTH CYCLE, SHIFTS ON LAST
PHASING WILL BE DONE TAKING INTO ACCOUNT THE CORRELATIONS AMONG DERIVATIVES
THE GROUPS OF DERIVATIVES WITH CORRELATIONS WILL BE UPDATED
THE BETA VALUES FOR EACH DERIV WILL BE SET TO 1.0
PHASE-AVERAGED RESIDUALS WILL BE USED FOR PHASING

TYPE OF REFINEMENT SELECTED:
UNPHASED ORIGIN-REMOVED PATTERSON REFINEMENT ONLY

Bayesian correlated MAD phasing will be used with wavelength 2
as the reference wavelength.

```
RESOLUTION LIMITS IN ANGSTROMS:      3.000    20.000
MINIMUM RATIO OF FDER TO RMS LACK-OF-CLOSURE FOR  INCLUSION IN REFINEMENT OR PHASING=      0.000
MINIMUM NATIVE F:      0.000
MINIMUM FIGURE OF MERIT FOR PHASED REFINEMENT:      0.000
MINIMUM ALLOWED ISOTROPIC B:      0.000
PARAMETER SHIFTS GREATER THAN     0.0000 TIMES SIGMA  WILL BE SCALED BY     0.5000
MINIMUM RATIO OF FNAT/SIGMA OR FDER/SIGMA TO INCLUDE:      1.000

NUMBER OF REFINEMENT CYCLES IS      2
DERIVATIVES REFINED DURING THESE CYCLES ARE :  0  0
TYPE OF OUTPUT SELECTED IS: +10 COLUMNS OF HENDRICKSON-LATTMAN COEFFICIENTS
1 INPUT data FILE WITH  12 COLUMNS IS:
mad_fpfm.scl
COLUMN  0 : mad_fpfm.scl Fnat,sig,(F+,sig,F-,sig)n
COLUMN  1 : F from I_TO_F       set 1 with 2 se atoms, lambda 1
COLUMN  2 : SIGMA of F          set 1 with 2 se atoms, lambda 1
COLUMN  3 : F from I_TO_F       set 1 with 2 se atoms, lambda 1
COLUMN  4 : SIGMA of F          set 1 with 2 se atoms, lambda 1
COLUMN  5 : F from I_TO_F       set 2
COLUMN  6 : SIGMA of F          set 2
COLUMN  7 : F from I_TO_F       set 2
COLUMN  8 : SIGMA of F          set 2
COLUMN  9 : F from I_TO_F       set 3
COLUMN 10 : SIGMA of F          set 3
COLUMN 11 : F from I_TO_F       set 3
COLUMN 12 : SIGMA of F          set 3


DERIVATIVE INFORMATION FOR  3 COMPOUNDS


COMPOUND 1     TEST REFINEMENT LAMBDA 3  (ANO ONLY)
COLUMNS FOR F+, SIGMA, F-, SIGMA     1    2    3    4
THIS DERIVATIVE WILL NOT BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
ONLY ANO DIFFERENCES WILL BE USED IN REFINEMENT AND PHASING FOR THIS DERIVATIVE.
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
NO PARAMETERS REFINED FOR ATOM LAM1 WITH ZERO OCCUPANCY

COMPOUND 2     set 2
COLUMNS FOR F+, SIGMA, F-, SIGMA     5    6    7    8
THIS DERIVATIVE WILL BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 3     set 3
COLUMNS FOR F+, SIGMA, F-, SIGMA     9   10   11   12
THIS DERIVATIVE WILL NOT BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
```

```
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
NO PARAMETERS REFINED FOR ATOM LAM3 WITH ZERO OCCUPANCY
CARRYING OUT STANDARD REFINEMENT
Total of           2 cycles will be done
Derivs refined will be           0          0

SUMMARY OF RESULTS ON FINAL CYCLE:

NUMBER OF REFLECTIONS READ =    3877
NUMBER OF F .GT. FMIN      =    3877
NUMBER OF F IN RES. LIMITS =    3877
NUMBER OF F .GT. MIN <M>   =    3854
NUMBER OF F USED TO REFINE =       0



FIGURE OF MERIT <       0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1.0
# OF REFLECTIONS        91    81    83    93    97   101   141   241   478  2467

FIGURE OF MERIT WITH RESOLUTION
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
N:             3873    231    344    416    469    538    598    627    650
MEAN FIG MERIT:   0.84   0.79   0.87   0.83   0.84   0.83   0.85   0.85   0.83

RMS ANOMALOUS FH/E  [f" PART OF FH / RMS ANO ERROR]:

LAMBDA:  1        1.2    0.9    1.3    1.1    1.2    1.3    1.4    1.4    1.2
LAMBDA:  2        1.8    1.6    1.9    1.8    1.8    1.9    1.8    1.9    1.7
LAMBDA:  3        1.3    1.1    1.4    1.3    1.4    1.4    1.4    1.5    1.3

RMS DISPERSIVE FH/E  [Delta-f-prime PART OF FH / RMS DISPERSIVE ERROR]:

L1 VS L2:         0.7    0.6    0.7    0.7    0.7    0.8    0.8    0.9    0.8
L1 VS L3:         1.7    1.5    1.8    1.7    1.6    1.7    1.8    1.9    1.8
L2 VS L3:         1.2    1.1    1.2    1.1    1.0    1.2    1.3    1.4    1.2

RMS ANOMALOUS FH  [f" PART OF FH] AS % of F:

LAMBDA:  1        3.1    2.4    3.2    2.9    2.9    3.3    3.5    3.6    3.6
LAMBDA:  2        5.0    4.0    5.1    4.6    4.6    5.2    5.5    5.6    5.7
LAMBDA:  3        3.6    2.8    3.7    3.3    3.4    3.7    3.9    4.1    4.1

RMS DISPERSIVE FH  [Delta-f-prime PART OF FH] AS % of F:

L1 VS L2:         2.5    1.9    2.7    2.4    2.4    2.7    2.8    2.9    3.0
L1 VS L3:         6.9    5.3    7.3    6.6    6.5    7.4    7.8    8.0    8.1
L2 VS L3:         4.5    3.6    4.6    4.3    4.1    4.7    4.9    5.1    5.2

RMS ANOMALOUS ERRORS [ CALC - OBS VALUE OF (F+ - F-)/2], IN % OF RMS F:
LAMBDA:  1        2.5    2.5    2.6    2.5    2.5    2.5    2.5    2.5    2.9
LAMBDA:  2        2.8    2.5    2.6    2.6    2.5    2.7    3.0    3.0    3.4
LAMBDA:  3        2.7    2.5    2.7    2.5    2.5    2.6    2.8    2.8    3.1
```

```
RMS DISPERSIVE ERRORS [ CALC - OBS VALUE OF (F(i) - F(j))], IN % OF RMS F:
L1 VS L2:       3.5   3.4   3.6   3.5   3.3   3.5   3.5   3.4   3.6
L1 VS L3:       4.0   3.6   4.1   3.8   4.0   4.3   4.4   4.3   4.5
L2 VS L3:       3.8   3.3   3.9   3.8   4.0   3.8   3.9   3.7   4.1

CORRELATED ANOMALOUS ERRORS BY WAVELENGTH (%):

LAMBDA:  1       0.8   0.0   0.7   0.6   0.7   0.9   1.2   1.2   1.2
LAMBDA:  2       1.4   0.0   1.1   1.0   1.2   1.4   1.9   2.0   1.8
LAMBDA:  3       1.0   0.0   0.8   0.7   0.9   1.0   1.4   1.4   1.3

RMS F BY WAVELENGTH:

LAMBDA:  1     102.3  187.9  123.7  128.7  117.6   97.2   84.0   76.5   69.5
LAMBDA:  2     100.4  172.7  125.0  124.5  117.9   96.6   83.6   76.7   69.0
LAMBDA:  3     101.6  180.5  123.8  127.6  117.1   98.3   84.5   77.1   69.6


PARAMETER SHIFTS FOR DERIV  2 : set 2

                    SCALE FACTOR     OVERALL B
CURRENT VALUES:         1.0000         0.0000



                SITE  ATOM      OCCUP     X       Y       Z        B


CURRENT VALUES:     1    Se      0.6032  0.4396  0.1591  0.3797   26.0691

CURRENT VALUES:     2    Se      0.5493  0.2302  0.4500  0.1646   20.6222



*************************************************************
*************************************************************


*** Summary of solutions and their relationships
 to each other and to check solution ***


-------------------------------------------------------
solution #          1 with overall quality =   9.114912
Derivative           2 with           2 sites.
Overall scale =    1.000000    and  overall b of   0.0000000E+00
 0.4396450      0.1591296      0.3797142      0.6032023       26.06913
 0.2302192      0.4500228      0.1646223      0.5493416       20.62219
Best match of solution 1 -> solution 2:

-------- solution 1 --------    -------------solution 2 ------
site     x       y       z    site     x       y       z     DIST (A)
Derivative  2
 1    0.440    0.159    0.380    1    0.440    0.160    0.380    0.06
```

```
 2    0.230    0.450    0.165    2    0.230    0.450    0.165    0.02
Comparison of this solution with check solution:
Number of sites in this solution matching check=          2
... and number not matching =          0
by derivative, this is...
Deriv        nsame        ndifferent
        1        0        0
        2        2        0
        3        0        0
All sites in this solution are contained in check soln
```

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
#
#  command file to generate a 2-deriv MIR dataset
#
solve << EOD
resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym

coordinatefile coords.pdb
percent_error 10
iranseed -124093
OVERWRITE                          ! overwrite duplicate file names without
                              !     asking
logfile generate.logfile
solvefile generate.prt
!
deriv 1                        ! enter parameters for deriv 1
!
inano                          ! use anom diffs
atom hg
occ 0.8
bvalue 40.
xyz -0.620932 -0.0765346 -0.637333
atom hg
occ 0.4
bvalue 25.
xyz -0.315098 -0.512727 -0.664318
!
deriv 2
inano
atom hg
occ  0.8
bvalue 30.
xyz 0.620932 0.5765346 0.337333
atom hg
occ 0.6
```

```
bvalue 25.
xyz 0.515098 0.212727 0.364318

GENERATE_MIR                    ! generate the MIR dataset now.
!
EOD
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
#  solve.com   solve an MIR problem
#
solve<<EOD > solve.log
logfile solve.logfile

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym


readformatted       ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged           ! premerged/ unmerged
read_intensities    ! read_intensities/read_amplitudes
fixscattfactors     ! fixscattfactors/refscattfactors

rawnativefile native.intensities ! native data H K L Iobs Sigma usually
                    ! NOTE: all datafiles must be in the same format

derivative 1                ! about to enter information on derivative #1
label deriv 1 HG            ! a label for this deriv
atom hg

rawderivfile der1.intensities  !  derivative data
inano           ! You need to tell it if anomalous diffs are to be used
noanorefine     ! use anomalous differences in phasing
                ! but not refinement (best option for MIR)
```

```
nsolsite_deriv 2        ! max 2 sites this deriv

derivative 2
label deriv 2 also hg
atom hg

inano
noanorefine                    ! use anomalous differences in phasing
                           ! but not refinement (best option for MIR)
rawderivfile der2.intensities       ! the derivative data is in this file


acceptance 0.35                ! accept a new site if it has a
                           ! peak height about 1/3 of avg or more
nsolsite 2                     ! number of sites per deriv
                           ! (use nsolsite_deriv to set individual values)
SCALE_NATIVE                   ! scale the native dataset
SCALE_MIR                    ! scale the derivs to the native
ANALYZE_MIR                    ! analyze this mir data and set up for SOLVE
SOLVE
EOD
#
# Now run Resolve to do density modification and build a model
#
resolve << EOD > resolve.log
!solvent_content 0.40       !    solvent fraction
seq_file seq.dat           !   sequence file
EOD
#
#  That's it! Now resolve.mtz has your updated phases
#  and resolve.pdb has your model
#
```

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*********************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*********************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 6)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: SOLVE 08-Apr-05

Space group number is: 5
Space group name from file name is: c2
Rescaling standard dataset to put it on approximate absolute scale. NRES = 200; <F**2> expected = 196000.0
; <F**2> observed in lowest resolution shell = 440464.3
... Scale factor = 0.4449850
Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0

7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

SCALE_MIR for dataset 1
Scale derivatives to previously-scaled native.

Default of "fp_or_fm" (use either F+ or
F- if available) will be used as this flag was not set

Analysis of this MIR dataset.

Fnative, sigma, and (Fbar,sigma, delano,sig) for 2
derivatives written to:
mir_fbar.scl

Fnative, sigma, and (F+,sigma,F-,sig) for 2 derivatives written to:
mir_fpfm.scl

** Completeness of native data (F > 2.000000 * sigma)

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0

total 1822 1822 100.0

*** Analysis of this scaled deriv data set ***

** Completeness of Fbar data for each derivative: **

Derivative 1
deriv 1 HG ! a label for this deriv

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	0	0.0
2	4.500	317	0	0.0
3	4.200	120	0	0.0
4	3.975	120	17	14.2
5	3.750	155	154	99.4
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0

total 1822 1042 57.2

Derivative 2
deriv 2 also hg

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	238	99.6
2	4.500	317	317	100.0
3	4.200	120	119	99.2

4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	253	99.6
total		1822	1819	99.8

** R-factors for F-bar data isomorphous differences **

isomorphous differences derivs 1 - native

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	0	0.000	0.000	0.000	0.00	0.00	0.00
2	4.500	0	0.000	0.000	0.000	0.00	0.00	0.00
3	4.200	0	0.000	0.000	0.000	0.00	0.00	0.00
4	3.975	17	195.016	0.205	0.983	39.51	32.33	1.22
5	3.750	153	231.026	0.190	0.994	43.75	36.75	1.19
6	3.600	111	209.183	0.198	0.992	41.67	32.39	1.29
7	3.450	143	170.781	0.252	0.999	43.65	27.58	1.58
8	3.300	172	180.269	0.218	0.998	38.64	28.79	1.34
9	3.150	190	172.200	0.206	0.998	36.13	26.52	1.36
10	3.000	254	158.512	0.199	0.999	31.17	25.13	1.24
Total:		1040	182.971	0.208	0.997	38.44	29.19	1.32

Recommended resolution cut-off = 3.00

isomorphous differences derivs 2 - native

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N

1	6.000	238	314.223	0.265	0.999	89.08	51.96	1.71
2	4.500	316	280.927	0.224	1.000	66.42	44.28	1.50
3	4.200	119	269.513	0.210	0.996	57.77	42.41	1.36
4	3.975	120	241.903	0.240	0.998	61.43	38.12	1.61
5	3.750	154	234.726	0.238	1.002	58.53	37.16	1.57
6	3.600	112	210.333	0.236	0.994	51.43	32.32	1.59
7	3.450	143	170.781	0.238	1.000	42.63	27.27	1.56
8	3.300	171	178.917	0.248	0.997	47.01	28.79	1.63
9	3.150	189	172.456	0.232	0.997	42.39	26.81	1.58
10	3.000	252	157.994	0.257	1.003	43.14	25.17	1.71
Total:		1814	227.006	0.240	0.999	59.41	37.30	1.59

Recommended resolution cut-off = 3.00

** R-factors for anomalous differences **

anomalous differences deriv 1

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	0	0.000	0.000	0.000	0.00	0.00	0.00
2	4.500	0	0.000	0.000	0.000	0.00	0.00	0.00
3	4.200	0	0.000	0.000	0.000	0.00	0.00	0.00
4	3.975	16	172.767	0.130	1.010	0.00	32.65	0.00
5	3.750	148	228.107	0.128	1.006	0.00	40.23	0.00
6	3.600	108	199.183	0.128	1.005	0.00	35.38	0.00
7	3.450	140	172.896	0.148	1.003	16.44	29.56	0.56
8	3.300	168	176.979	0.133	1.001	0.00	31.63	0.00
9	3.150	186	163.771	0.141	1.002	11.42	28.92	0.39
10	3.000	246	152.065	0.132	1.000	0.00	26.93	0.00
Total:		1012	177.711	0.134	1.003	6.39	31.69	0.15

Recommended resolution cut-off = 3.00

anomalous differences deriv 2

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	233	303.943	0.141	1.002	17.34	57.37	0.30
2	4.500	308	274.102	0.130	1.005	0.00	49.65	0.00
3	4.200	114	253.899	0.108	1.001	0.00	45.51	0.00
4	3.975	119	244.065	0.121	1.002	0.00	43.92	0.00
5	3.750	149	227.131	0.131	0.999	0.00	40.83	0.00
6	3.600	111	204.313	0.128	1.000	0.00	36.31	0.00
7	3.450	142	170.840	0.140	0.999	4.05	31.01	0.13
8	3.300	164	173.636	0.129	1.001	0.00	31.28	0.00
9	3.150	186	164.302	0.120	0.999	0.00	30.04	0.00
10	3.000	247	155.606	0.146	1.000	7.54	28.43	0.27
Total:		1773	220.802	0.131	1.001	0.00	41.48	0.09

Recommended resolution cut-off = 4.50

Script file suitable for running SOLVE written to:
solve_mir.script

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
#
#  generate MIR dataset with ano scatterer in native
#
solve  <<EOD

! generate fe-native + MIR dataset

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym

logfile generate.logfile
solvefile generate.prt
coordinatefile coords.pdb
percent_error  1
iranseed 31933

! native dataset treated as 1-wavelength Fe data
!
mad_atom fe
lambda 1
label 1.5418 A Fe-native dataset
wavelength 1.5418
fprimv_mad  -1              ! f' value at this wavelength
fprprv_mad  3
ATOMNAME fe
xyz  0.44 0.16 0.38
occ 1.0
bvalue 20
generate_mad
! now lam_1.intensities = native intensity data at 1.5418 A
end
EOD

solve  <<EOD
```

```
! generate fe-native + MIR dataset
@solve.setup
logfile generate_mir.logfile
solvefile generate_mir.prt
coordinatefile coords.pdb
percent_error 4
iranseed 184231

deriv 1
inano
atom hg
occ 1.0
bvalue 31.
xyz 0.15 0.25 0.35

deriv 2
inano
atomname hg
occ 1.0
bvalue 25
xyz 0.51 0.31 0.145
GENERATE_MIR                    ! generate the MIR dataset now.
!  now der1.intensities, der2.intensities have deriv data
!  native.intensities has native data WITHOUT anom data (don't use)
EOD
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
#  command file to solve a 2-deriv MIR dataset + anom diffs in native
#
solve<<EOD > solve.log
logfile solve.logfile

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym


readformatted       ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged           ! premerged/ unmerged
read_intensities    ! read_intensities/read_amplitudes
fixscattfactors     ! fixscattfactors/refscattfactors

rawnativefile lam1.intensities ! native data

! now read in native data again as a pseudo-deriv 1:
! put in scattering factors for pseudo-atom fel1 with only ano
! scattering

newatomtype FEL1
aval 0 0 0 0
bval 0 0 0 0
cval 0.001
fprimv 0
```

```
fprprv 3

derivative 1
label native data with ano diffs, treated as deriv 1
nsolsite_deriv 1              ! just 1 site
rawderivfile lam1.intensities
inano
anoonly                       ! no isomorphous diffs, just ano
atomname fel1                 ! use our pseudo-atom with no iso scattering

! now read in the regular 2 derivs:
derivative 2                  ! about to enter information on derivative #1
label deriv 1 HG              ! a label for this deriv
nsolsite_deriv 1              ! one site in this deriv
rawderivfile der1.intensities  ! data from generate.com
inano                         ! You need to tell it if anomalous diffs are
                              ! to be used
noanorefine                   ! do not use ano diffs for refinement
atomname Hg                   ! tell SOLVE what the heavy atom is

derivative 3
label deriv 2 also hg
rawderivfile der2.intensities       ! the derivative data is in this file
nsolsite_deriv 2
inano
noanorefine                   ! do not use ano diffs for refinement
atomname hg
acceptance 0.35               ! accept a new site if it has a
                              ! peak height about 1/3 of avg or more
SCALE_NATIVE                  ! scale the native dataset
SCALE_MIR                     ! scale the derivs to the native
ANALYZE_MIR                   ! analyze this mir data and set up for SOLVE
SOLVE
EOD
#
# Now run Resolve to do density modification and build a model
#
resolve << EOD > resolve.log
!solvent_content 0.4       !    solvent fraction
EOD
#
#  That's it! Now resolve.mtz has your updated phases
#  and resolve.pdb has your model
```

#

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*****************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*****************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 6)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: SOLVE 06-Apr-05

Space group number is: 5
Space group name from file name is: c2
Rescaling standard dataset to put it on approximate absolute scale. NRES =
200; <F**2> expected = 196000.0
; <F**2> observed in lowest resolution shell = 430905.9
... Scale factor = 0.4548557
Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

SCALE_MIR for dataset 1
Scale derivatives to previously-scaled native.

Default of "fp_or_fm" (use either F+ or

F- if available) will be used as this flag was not set

Analysis of this MIR dataset.

Fnative, sigma, and (Fbar,sigma, delano,sig) for 3
derivatives written to:
mir_fbar.scl

Fnative, sigma, and (F+,sigma,F-,sig) for 3 derivatives written to:
mir_fpfm.scl

** Completeness of native data (F > 2.000000 * sigma)

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

*** Analysis of this scaled deriv data set ***

** Completeness of Fbar data for each derivative: **

Derivative 1
native data with ano diffs, treated as deriv 1

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	234	97.9
2	4.500	317	314	99.1
3	4.200	120	118	98.3
4	3.975	120	117	97.5

5	3.750	155	154	99.4
6	3.600	112	111	99.1
7	3.450	143	141	98.6
8	3.300	172	170	98.8
9	3.150	190	189	99.5
10	3.000	254	249	98.0
total		1822	1797	98.6

Derivative 2
deriv 1 HG ! a label for this deriv

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

Derivative 3
deriv 2 also hg

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

** R-factors for F-bar data isomorphous differences **

isomorphous differences derivs 1 - native

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	234	308.792	0.001	1.000	0.00	1.32	0.00
2	4.500	313	286.401	0.001	1.000	0.00	1.15	0.00
3	4.200	118	272.982	0.001	1.000	0.00	1.10	0.00
4	3.975	117	237.600	0.001	1.000	0.00	0.98	0.00
5	3.750	154	240.469	0.001	1.000	0.00	0.98	0.00
6	3.600	111	210.582	0.001	1.000	0.00	0.84	0.00
7	3.450	140	177.880	0.001	1.000	0.00	0.71	0.00
8	3.300	170	179.320	0.001	1.000	0.00	0.74	0.00
9	3.150	188	172.708	0.001	1.000	0.00	0.69	0.00
10	3.000	249	159.973	0.001	1.000	0.00	0.64	0.00
Total:		1794	228.544	0.001	1.000	0.00	0.96	0.00

Recommended resolution cut-off = 3.00

isomorphous differences derivs 2 - native

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	238	316.171	0.234	1.003	89.05	21.26	4.19
2	4.500	317	290.436	0.214	1.012	75.05	18.12	4.14
3	4.200	120	277.707	0.209	1.015	69.64	18.02	3.86
4	3.975	120	242.145	0.221	1.005	63.91	15.58	4.10
5	3.750	155	241.042	0.192	1.003	57.14	15.68	3.64
6	3.600	112	212.302	0.218	1.002	57.23	13.62	4.20
7	3.450	142	179.511	0.236	1.002	49.81	11.53	4.32
8	3.300	172	182.340	0.231	1.003	50.41	12.03	4.19
9	3.150	190	174.773	0.202	1.000	42.54	11.02	3.86
10	3.000	254	164.417	0.201	1.001	39.86	10.47	3.81
Total:		1820	232.231	0.216	1.005	62.70	15.43	4.03

Recommended resolution cut-off = 3.00

isomorphous differences derivs 3 - native

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N

1	6.000	239	317.834	0.260	1.000	101.89	21.48	4.74
2	4.500	317	290.436	0.237	1.007	83.68	18.42	4.54
3	4.200	120	277.707	0.223	1.004	72.53	17.56	4.13
4	3.975	120	242.145	0.265	1.000	76.98	16.43	4.69
5	3.750	155	241.042	0.232	1.001	65.68	15.63	4.20
6	3.600	112	212.302	0.268	1.002	67.02	13.35	5.02
7	3.450	143	180.467	0.300	1.000	63.27	11.52	5.49
8	3.300	172	182.340	0.255	1.002	56.74	11.68	4.86
9	3.150	190	174.773	0.258	0.999	52.58	10.90	4.82
10	3.000	254	164.417	0.255	1.006	49.73	10.69	4.65

Total: 1822 232.541 0.252 1.003 72.19 15.53 4.70

Recommended resolution cut-off = 3.00

** R-factors for anomalous differences **

anomalous differences deriv 1

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	233	305.990	0.045	1.000	5.66	17.36	0.33
2	4.500	310	283.528	0.047	1.000	9.22	15.05	0.61
3	4.200	118	273.420	0.046	1.000	7.46	14.57	0.51
4	3.975	116	234.454	0.042	1.001	2.18	12.86	0.17
5	3.750	151	235.985	0.046	1.000	5.65	12.71	0.44
6	3.600	110	208.529	0.040	1.000	3.65	11.11	0.33
7	3.450	137	173.756	0.047	1.000	5.47	9.20	0.59
8	3.300	166	175.347	0.048	1.000	5.05	9.65	0.52
9	3.150	187	172.394	0.052	1.000	6.96	9.19	0.76
10	3.000	245	157.844	0.042	1.000	0.00	8.46	0.00

Total: 1773 226.195 0.046 1.000 6.04 12.63 0.43

Recommended resolution cut-off = 3.00

anomalous differences deriv 2

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	235	317.747	0.089	0.998	10.35	37.75	0.27
2	4.500	307	271.861	0.089	1.000	10.70	30.90	0.35

3	4.200	117	282.674	0.090	1.001	13.28	31.71	0.42
4	3.975	117	232.714	0.082	1.004	0.00	27.04	0.00
5	3.750	153	238.028	0.089	1.002	4.40	28.19	0.16
6	3.600	111	213.931	0.087	1.003	0.00	24.32	0.00
7	3.450	139	180.621	0.093	1.000	9.61	20.35	0.47
8	3.300	171	184.635	0.096	0.999	9.40	21.35	0.44
9	3.150	187	170.440	0.103	0.999	11.04	19.46	0.57
10	3.000	248	158.611	0.090	0.999	4.15	18.08	0.23

Total: 1785 227.722 0.091 1.000 8.55 27.05 0.31

Recommended resolution cut-off = 3.00

anomalous differences deriv 3

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	235	315.881	0.092	0.998	9.94	37.95	0.26
2	4.500	312	281.434	0.090	1.002	9.41	32.13	0.29
3	4.200	117	271.861	0.076	1.000	0.00	30.99	0.00
4	3.975	117	247.817	0.081	1.001	0.00	29.00	0.00
5	3.750	152	234.973	0.087	1.001	0.00	27.08	0.00
6	3.600	111	209.664	0.105	1.000	14.18	23.86	0.59
7	3.450	140	180.503	0.083	1.000	0.00	20.51	0.00
8	3.300	169	175.354	0.093	1.001	5.74	20.32	0.28
9	3.150	188	166.699	0.093	1.001	4.87	19.14	0.25
10	3.000	252	166.419	0.100	1.001	9.96	19.02	0.52

Total: 1793 228.674 0.090 1.001 6.38 27.28 0.25

Recommended resolution cut-off = 3.00

Script file suitable for running SOLVE written to:
solve_mir.script

**** SOLVE: Solutions to MIR or SIR datasets ******

Derivatives considered: 3 (NSET)
Cross-vectors tested in HASSP: 6 (ICRMAX, DEFAULT=20)
HASSP solutions saved per deriv: 30 (NTOPHASSP, DEFAULT=30)
Fourier peaks saved per map: 30 (NTOPFOUR, DEFAULT=10)
Sites per derivative: 5 (NSOLSITE, DEFAULT=20)
Derivative solutions per seed: 5 (NTOPDERIV, DEFAULT=5)
Seeds per derivative tested: 3 (NSEEDTEST,DEFAULT=10)
Sorted seeds to use 5 (NSEEDSOLVE, DEFAULT=5)
Number of final solutions saved: 5 (NTOPSOLVE, DEFAULT=5)

```
 Sites per derivative vary with derivative.
Derivative   Max sites
    1          1

    2          1

    3          2


Solutions obtained will be compared to input solution (ICHECKSOLVE)
Correlated phasing used (CORRELPHASE)
Standard difference fouriers will be calculated for derivative          1
Standard difference fouriers will be calculated for derivative          2
Standard difference fouriers will be calculated for derivative          3

Datafile with          14 columns of data:
Title:mir_fbar.scl Fnat,sig,(fbar,sig,delano,sig)n
Data: Native F data
Data: Native sigma of F data
Data: Fbar              native data with ano diffs, treated as d
Data: Sig of Fbar        native data with ano diffs, treated as d
Data: Del Ano (F+ - F-)   native data with ano diffs, treated as d
Data: sig of Del Ano      native data with ano diffs, treated as d
Data: Fbar              deriv 1 HG              ! a label for t
Data: Sig of Fbar        deriv 1 HG              ! a label for t
Data: Del Ano (F+ - F-)   deriv 1 HG              ! a label for t
Data: sig of Del Ano      deriv 1 HG              ! a label for t
Data: Fbar              deriv 2 also hg
Data: Sig of Fbar        deriv 2 also hg
Data: Del Ano (F+ - F-)   deriv 2 also hg
Data: sig of Del Ano      deriv 2 also hg

Fnat,sigma taken from columns          1          2
Fder,sig,Delano,sig deriv          1 from cols:          3          4
         5          6
Fder,sig,Delano,sig deriv          2 from cols:          7          8
         9         10
Fder,sig,Delano,sig deriv          3 from cols:         11         12
        13         14

Check solution to be compared to all solutions found:

Derivative          1:
 Site    X       Y       Z
   1   0.440   0.160   0.380

Derivative          2:
 Site    X       Y       Z
   1   0.150   0.250   0.350

Derivative          3:
 Site    X       Y       Z
```

```
  1   0.510   0.310   0.145

*********************************************************

ANALYZE_SOLVE:  analysis of top          1 solutions



*************************************************************


Solution          1

********************** Analysis of this solution *************

 ******    Analysis of non-randomness of native Fourier map ******

A.  Maps with distinct solvent regions havea high standard deviation
of local r.m.s. electron density.
For this map the SD of this local r.m.s. is   0.3133972


B.  Maps with distinct solvent regions also have a high correlation
of local r.m.s. electron density with
density at neighboring locations. Typical
values for poor maps in this
 structure solution are   0.1325388     +/-   7.8560203E-02
For this map the correlation of r.m.s. density
in neighboring boxes is   0.3252193
The correlation coefficient is used here in scoring.

Skew of the map is:   0.6867959

******  Analysis of derivative solutions with the difference Patterson ******
  and with cross-validation difference Fouriers


 ------------------------------------------------
     Derivative #          1



List of sites analyzed for compatibility with difference Patterson

  PEAK        X         Y         Z     OPTIMIZED
                                       RELATIVE OCCUPANCY
    1     0.443     0.167     0.385      44.174

Evaluation of this test soln with    1 sites after optimizing
occupancy of each site

Cross-vectors for sites  1 and  1 (excluding origin; 1000 = 1 sigma):
 #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
  1   -0.885   0.000  -0.771   3902.67     3902.67          2
```

Overall quality of this Patterson soln = 1379.80
Overall quality of the fit to patterson = 0.690534E-06
Avg normalized peak height = 975.668

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.443	0.160	0.381	1.254	60.000	13.41

Derivative # 2

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.151	0.250	0.354	98.637

Evaluation of this test soln with 1 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.302	0.000	-0.708	19458.5	19458.5	2

Overall quality of this Patterson soln = 6879.60
Overall quality of the fit to patterson = 0.138107E-05
Avg normalized peak height = 4864.61

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.151	0.254	0.350	0.593	29.336	27.89

Derivative # 3

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.510	0.319	0.146	102.387

Evaluation of this test soln with 1 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.021	0.000	-0.292	20966.1	20966.1	2

Overall quality of this Patterson soln = 7412.63
Overall quality of the fit to patterson = 0.110485E-04
Avg normalized peak height = 5241.52

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.510	0.313	0.145	0.653	25.943	27.67

Summary of scoring for this solution:

	-- over many solutions--		-- this solution --	
Criteria	MEAN	SD	VALUE	Z-SCORE
Pattersons:	2.15	1.84	6.99	2.63
Cross-validation Fourier:	0.000E+00	0.500	43.5	87.0
NatFourier CCx100:	13.3	7.86	32.5	2.45
Mean figure of meritx100:	0.000E+00	19.3	85.9	4.44
Correction for Z-scores:			-43.1	

Overall Z-score value: 53.4

****** Overall analysis of phasing for solution 1**********

HEAVY: Refine heavy atom parameters

File title: SOLVE 06-Apr-05
CRYSTALLOGRAPHIC PARAMETERS

A = 76.00 B = 28.00 C = 42.00 alpha = 90.00 beta = 103.00 gamma = 90.00

PHASES CALCULATED EVERY 5 DEGREES
RESIDUALS CALCULATED ON EXTRA ZEROTH CYCLE ONLY
SIGMAS FROM data FILE WILL BE USED
STATISTICS WILL BE PRINTED ON ZEROTH CYCLE, SHIFTS ON LAST
PHASING WILL BE DONE TAKING INTO ACCOUNT THE CORRELATIONS AMONG DERIVATIVES
THE GROUPS OF DERIVATIVES WITH CORRELATIONS WILL BE UPDATED
THE BETA VALUES FOR EACH DERIV WILL BE SET TO 1.0
PHASE-AVERAGED RESIDUALS WILL BE USED FOR PHASING

TYPE OF REFINEMENT SELECTED:
UNPHASED ORIGIN-REMOVED PATTERSON REFINEMENT ONLY

Bayesian Correlated Phasing will be used

```
 RESOLUTION LIMITS IN ANGSTROMS:      3.000    20.000
 MINIMUM RATIO OF FDER TO RMS LACK-OF-CLOSURE FOR  INCLUSION IN REFINEMENT OR PHASING=      0.000
 MINIMUM NATIVE F:      0.000
 MINIMUM FIGURE OF MERIT FOR PHASED REFINEMENT:      0.000
 MINIMUM ALLOWED ISOTROPIC B:      0.000
 PARAMETER SHIFTS GREATER THAN     0.0000 TIMES SIGMA  WILL BE SCALED BY     0.5000
 MINIMUM RATIO OF FNAT/SIGMA OR FDER/SIGMA TO INCLUDE:      1.000

 NUMBER OF REFINEMENT CYCLES IS      2
 DERIVATIVES REFINED DURING THESE CYCLES ARE :  0  0
 TYPE OF OUTPUT SELECTED IS: +10 COLUMNS OF HENDRICKSON-LATTMAN COEFFICIENTS
1 INPUT data FILE WITH  14 COLUMNS IS:
 mir_fbar.scl
 COLUMN  0 : mir_fbar.scl Fnat,sig,(fbar,sig,delano,sig)n
 COLUMN  1 : Native F data
 COLUMN  2 : Native sigma of F data
 COLUMN  3 : Fbar              native data with ano diffs, treated as deriv 1
 COLUMN  4 : Sig of Fbar        native data with ano diffs, treated as deriv 1
 COLUMN  5 : Del Ano (F+ - F-)   native data with ano diffs, treated as deriv 1
 COLUMN  6 : sig of Del Ano      native data with ano diffs, treated as deriv 1
 COLUMN  7 : Fbar              deriv 1 HG              ! a label for this deriv
 COLUMN  8 : Sig of Fbar        deriv 1 HG              ! a label for this deriv
 COLUMN  9 : Del Ano (F+ - F-)   deriv 1 HG              ! a label for this deriv
 COLUMN 10 : sig of Del Ano      deriv 1 HG              ! a label for this deriv
 COLUMN 11 : Fbar              deriv 2 also hg
 COLUMN 12 : Sig of Fbar        deriv 2 also hg
 COLUMN 13 : Del Ano (F+ - F-)   deriv 2 also hg
 COLUMN 14 : sig of Del Ano      deriv 2 also hg


 data COLUMNS FOR NATIVE F AND SIGMA:     1    2
 data COLUMNS FOR BEST AND MOST PROB PHASES AND FIGURE OF MERIT:     0    0    0
 OVERALL SCALE FACTOR FOR ALL data =     1.000
 SCALE FACTOR FOR NATIVE SIGMAS =     1.000


 DERIVATIVE INFORMATION FOR  3 COMPOUNDS


 COMPOUND 1     native data with ano diffs, treated as deriv 1
 COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)    3    4    5    6
 THIS DERIVATIVE WILL BE USED IN PHASING
 ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
 OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
 ONLY ANO DIFFERENCES WILL BE USED IN REFINEMENT AND PHASING FOR THIS DERIVATIVE.
 AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
 DIVIDED BY :     1.001*EXP(     0.000*(SIN theta/LAMBDA)**2 )
 THEN SIGMAS WILL BE MULTIPLIED BY      1.000

 COMPOUND 2     deriv 1 HG              ! a label for this deriv
 COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)    7    8    9   10
 THIS DERIVATIVE WILL BE USED IN PHASING
 ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
```

```
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     0.958*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 3     deriv 2 also hg
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)   11   12   13   14
THIS DERIVATIVE WILL BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     0.947*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
CARRYING OUT STANDARD REFINEMENT
Total of           2 cycles will be done
Derivs refined will be           0          0

SUMMARY OF RESULTS ON FINAL CYCLE:

NUMBER OF REFLECTIONS READ =    1822
NUMBER OF F .GT. FMIN      =    1822
NUMBER OF F IN RES. LIMITS =    1822
NUMBER OF F .GT. MIN <M>   =    1822
NUMBER OF F USED TO REFINE =       0



FIGURE OF MERIT <       0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1.0
# OF REFLECTIONS        14    20    16    36    46    51    83   155   286  1115

FIGURE OF MERIT WITH RESOLUTION
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
N:             1822    106    152    202    220    247    278    298    319
MEAN FIG MERIT:   0.86   0.85   0.88   0.86   0.86   0.86   0.88   0.86   0.83





COMPOUND  1          native data with ano diffs, treated as deriv 1
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09

ANOM DIFFS:      1537.    72.   114.   170.   185.   208.   241.   261.   286.
RMS OBS DIFF:     14.1   19.9   17.3   19.1   15.8   13.7   12.1   11.4    9.5
RMS CALC DIFF:     6.5   12.1   10.3    8.9    7.0    5.8    4.9    4.0    3.1
RMS RESIDUAL:     13.1   18.3   16.6   17.6   15.6   12.7   11.0   10.2    8.7
RATIO ISO/ANO:    0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:             0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
ANOMALOUS LOC:           0.0    6.7    5.7    0.0    0.0    0.0    0.0    0.0
```

```
RMS FPH :            451.5  304.1  334.5  324.1  276.3  224.4  209.2  185.8
RMS FH  :             0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS SIGMA:             15.9   10.7   11.8   11.5    9.8    7.9    7.4    6.6
```

```
COMPOUND  2         deriv 1 HG              ! a label for this deriv
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09

CENTRIC REFLNS:   263.    33.    36.    31.    31.    36.    34.    33.    29.
RMS HA F:        83.4  119.6  100.2   91.2   83.4   74.0   64.8   58.9   48.6
RMS RESIDUAL:     37.3   47.8   42.9   43.1   35.1   40.2   27.5   29.9   23.7
RMS(FH)/RMS(E):   2.23   2.50   2.34   2.12   2.38   1.84   2.36   1.97   2.05
CENTRIC R FACT:   0.42   0.32   0.42   0.47   0.36   0.49   0.42   0.52   0.47

ACENTRIC REFLN:  1559.    73.   116.   171.   189.   211.   244.   265.   290.
RMS DERIV FPH:   287.7  474.4  338.9  348.1  344.8  288.3  239.7  222.6  198.2
RMS SIGMA FPH:    14.3   23.5   16.9   17.2   17.2   14.4   12.0   11.1    9.9
RMS SIGMA FP:      6.9   11.4    7.8    8.4    8.2    7.0    5.7    5.3    4.7
RMS HA F:        74.6  112.8  104.3   91.6   81.1   72.7   65.4   57.0   52.5
RMS RESIDUAL:     26.0   35.9   31.8   30.8   28.1   26.2   24.2   22.9   18.7
RMS(FH)/RMS(E):   2.87   3.14   3.28   2.97   2.89   2.77   2.70   2.49   2.81

ANOM DIFFS:      1558.    73.   116.   171.   189.   211.   244.   265.   289.
RMS OBS DIFF:     32.1   48.7   42.6   41.0   36.7   30.3   26.3   25.5   22.0
RMS CALC DIFF:    11.3   16.7   14.7   13.9   11.5   11.3   10.0    9.2    8.6
RMS RESIDUAL:     29.9   45.2   40.8   38.8   34.7   28.6   24.2   23.2   19.8
RATIO ISO/ANO:    9.09   9.61   9.45   9.29   9.13   8.99   8.86   8.74   8.62

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:            39.8   39.7   38.3   30.8   36.1   24.3   27.6   22.0
ANOMALOUS LOC:           0.0   22.9   17.9    4.6    0.0    3.8    6.3    2.7
RMS FPH :            474.4  338.9  348.1  344.8  288.3  239.7  222.6  198.2
RMS FH  :            112.8  104.3   91.6   81.1   72.7   65.4   57.0   52.5
RMS SIGMA:             26.1   18.6   19.1   19.0   16.0   13.2   12.4   11.0
```

```
COMPOUND  3         deriv 2 also hg
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09

CENTRIC REFLNS:   263.    33.    36.    31.    31.    36.    34.    33.    29.
RMS HA F:        94.4  130.3  115.6  107.0   82.5   90.3   76.9   66.0   58.3
RMS RESIDUAL:     39.0   56.7   43.2   43.1   37.8   37.8   30.7   24.2   27.7
RMS(FH)/RMS(E):   2.42   2.30   2.67   2.48   2.19   2.39   2.51   2.73   2.11
```

```
CENTRIC R FACT:   0.40   0.42   0.37   0.38   0.37   0.49   0.34   0.41   0.44

ACENTRIC REFLN:  1559.    73.   116.   171.   189.   211.   244.   265.   290.
RMS DERIV FPH:   291.4  478.6  340.6  356.1  338.5  304.9  242.7  220.0  203.9
RMS SIGMA FPH:    14.5   23.9   16.9   17.7   16.9   15.3   12.1   11.0   10.2
RMS SIGMA FP:      6.9   11.4    7.8    8.4    8.2    7.0    5.7    5.3    4.7
RMS HA F:        85.9  129.0  111.7  106.8   92.3   83.6   78.3   68.5   60.7
RMS RESIDUAL:     29.5   45.2   34.8   36.1   30.1   32.3   26.0   22.5   23.1
RMS(FH)/RMS(E):   2.91   2.85   3.20   2.96   3.07   2.59   3.01   3.05   2.63

ANOM DIFFS:     1558.    73.   116.   171.   189.   210.   244.   265.   290.
RMS OBS DIFF:     32.1   52.2   39.9   37.5   38.8   30.2   26.9   24.9   23.4
RMS CALC DIFF:    12.8   18.4   15.5   16.1   13.6   12.2   11.8   10.8    9.3
RMS RESIDUAL:     28.8   45.6   37.6   32.6   35.2   27.6   23.9   22.3   21.3
RATIO ISO/ANO:    9.09   9.61   9.45   9.29   9.13   8.99   8.86   8.74   8.62

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:            49.5   40.1   38.7   32.5   33.8   28.1   21.5   26.1
ANOMALOUS LOC:           0.0   16.6    0.0    9.6    0.0    0.0    3.0    6.1
RMS FPH :              478.6  340.6  356.1  338.5  304.9  242.7  220.0  203.9
RMS FH  :              129.0  111.7  106.8   92.3   83.6   78.3   68.5   60.7
RMS SIGMA:              26.5   18.6   19.5   18.8   16.8   13.4   12.3   11.2

Analysis of correlated modeling and non-isomorphism errors
obtained using phased residuals.
The derivatives were grouped into 2 sets  where the members of a set
had some mutual correlation.


Set 1 contains derivatives  1

Set 2 contains derivatives  2 3

SUMMARY OF CORRELATED ERRORS AMONG DERIVATIVES

DERIVATIVE:           1
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
RMS errors correlated and uncorrelated with others in group:
    Correlated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  Uncorrelated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 2:         0.00   0.00   0.00   0.00   0.00   0.00   0.01   0.01   0.00
DERIV 3:         0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
Errors correlated and uncorrelated with others in group:
    Correlated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  Uncorrelated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
```

```
DERIV 2:        0.00   0.00   0.00   0.00   0.00   0.00   0.02   0.00   0.00
DERIV 3:        0.00   0.01   0.00   0.00   0.00   0.00   0.00   0.00   0.00

DERIVATIVE:          2
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
RMS errors correlated and uncorrelated with others in group:
    Correlated:   26.4   35.6   27.5   32.7   25.4   22.6   22.5   20.7   19.9
  Uncorrelated:   20.1   17.9   28.6   19.8   17.4   28.1    9.3   18.2    9.3

Correlation of errors with other derivs:
DERIV 1:        0.00   0.00   0.00   0.00   0.00   0.00   0.01   0.01   0.00
DERIV 3:        0.52   0.52   0.44   0.61   0.53   0.37   0.63   0.62   0.63

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
Errors correlated and uncorrelated with others in group:
    Correlated:   23.9   33.0   35.5   23.2   23.2   26.5   25.4   17.3   17.2
  Uncorrelated:   16.4   10.7    7.9   24.9   17.6   12.3   13.2   21.0   12.7

Correlation of errors with other derivs:
DERIV 1:        0.00   0.00   0.00   0.00   0.00   0.00   0.02   0.00   0.00
DERIV 3:        0.44   0.42   0.63   0.31   0.40   0.48   0.56   0.35   0.40

DERIVATIVE:          3
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
RMS errors correlated and uncorrelated with others in group:
    Correlated:   26.4   35.6   27.5   32.7   25.4   22.6   22.5   20.7   19.9
  Uncorrelated:   22.9   34.4   29.1   20.7   20.4   25.1   17.0    6.0   16.8

Correlation of errors with other derivs:
DERIV 1:        0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
DERIV 2:        0.52   0.52   0.44   0.61   0.53   0.37   0.63   0.62   0.63

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
Errors correlated and uncorrelated with others in group:
    Correlated:   23.9   33.0   35.5   23.2   23.2   26.5   25.4   17.3   17.2
  Uncorrelated:   25.5   39.9   21.7   36.0   23.6   28.4   18.6   20.2   22.7

Correlation of errors with other derivs:
DERIV 1:        0.00   0.01   0.00   0.00   0.00   0.00   0.00   0.00   0.00
DERIV 2:        0.44   0.42   0.63   0.31   0.40   0.48   0.56   0.35   0.40


PARAMETER SHIFTS FOR DERIV  1 : native data with ano diffs, treated as deriv 1

                  SCALE FACTOR     OVERALL B
CURRENT VALUES:        1.0007         0.0000



              SITE  ATOM       OCCUP     X       Y       Z        B
```

```
CURRENT VALUES:     1    FEL1     1.2540  0.4427  0.1600  0.3808   60.0000


PARAMETER SHIFTS FOR DERIV  2 : deriv 1 HG              ! a label for this deriv

                    SCALE FACTOR     OVERALL B
CURRENT VALUES:         0.9583         0.0000



                SITE  ATOM       OCCUP     X       Y       Z        B


CURRENT VALUES:     1    Hg       0.5931  0.1505  0.2545  0.3504   29.3356


PARAMETER SHIFTS FOR DERIV  3 : deriv 2 also hg

                    SCALE FACTOR     OVERALL B
CURRENT VALUES:         0.9467         0.0000



                SITE  ATOM       OCCUP     X       Y       Z        B


CURRENT VALUES:     1    Hg       0.6534  0.5102  0.3131  0.1455   25.9431



*************************************************************
*************************************************************


*** Summary of solutions and their relationships
 to each other and to check solution ***


-------------------------------------------------------
solution #          1 with overall quality =   53.40180
Derivative           1 with           1 sites.
Overall scale =    1.000737     and  overall b of   0.0000000E+00
 0.4426706      0.1600000      0.3808467       1.254002       60.00000
Derivative           2 with           1 sites.
Overall scale =   0.9582843     and  overall b of   0.0000000E+00
 0.1505051      0.2544911      0.3504136      0.5931091       29.33563
Derivative           3 with           1 sites.
Overall scale =   0.9466927     and  overall b of   0.0000000E+00
 0.5102358      0.3130870      0.1454696      0.6533505       25.94308
Best match of solution 1 -> solution 2:

-------- solution 1 --------    -------------solution 2 ------
```

```
site     x       y       z    site     x       y       z     DIST (A)
Derivative  1
 1    0.443    0.160    0.381    1    0.440    0.160    0.380    0.20
Derivative  2
 1    0.151    0.254    0.350    1    0.150    0.250    0.350    0.13
Derivative  3
 1    0.510    0.313    0.145    1    0.510    0.310    0.145    0.09
Comparison of this solution with check solution:
Number of sites in this solution matching check=          3
... and number not matching =          0
by derivative, this is...
Deriv          nsame          ndifferent
        1          1          0
        2          1          0
        3          1          0
All sites in this solution are contained in check soln
```

Contents	Index

COMBINE: Combining multiple MAD and MIR datasets

Combining multiple MAD or MIR datasets is easy with SOLVE. You simply read in one dataset, scale and analyze it, then you give SOLVE the keyword "NEW_DATASET" and go on and read in the next dataset, scale and analyze it. When all datasets are input, you tell SOLVE to go ahead and solve the combined datasets with the command "COMBINE_ALL_DATA", followed by "SOLVE" as usual.

When combining datasets, SOLVE first converts MAD datasets to pseudo-MIR + anomalous scattering (i.e., native + 1 derivative + anomalous differences). Then SOLVE uses the first native dataset as the native. It treats the derivatives associated with that native as derivatives in the usual way. SOLVE treats the native for the other datasets as "derivatives" without heavy atoms, and the derivatives for these datasets as ordinary derivatives.

When the COMBINE command is used, solve writes out individual script files (e.g., solve_mad_01.script) that could be used to analyze the individual datasets, then a final script file (solve_combine.script) that can be used to analyze them all together.

SOLVE takes into account any non-isomorphisms among the different datasets by using Bayesian correlated phasing. This phasing method automatically corrects for non-isomorphism that is shared among a group of derivatives (or in this case derivatives and pseudo-derivatives).

NOTE: Unfortunately COMBINE cannot be used with multiple mtz files. If you have mtz data and want to use COMBINE you will need to dump your data out to a flat file and read it in using "readformatted".

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
#
# generate 1 mad and 1 mir dataset to solve
#
solve <<EOD

!  GENERATE MAD  and MIR DATASETS:

! dataset 1:   MAD dataset
resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym

percent_error 1
coordinatefile coords.pdb
iranseed -199753
logfile generate_mad.logfile
solvefile generate_mad.prt

mad_atom se                          ! define the scattering factors...
!
lambda 1
label set 1 with 2 se atoms, lambda 1
wavelength 0.9782            ! wavelength value
fprimv_mad  -10              ! f' value at this wavelength
fprprv_mad  3
ATOMNAME Se
xyz  0.44 0.16 0.38
occ 1.0
bvalue 20.
ATOMNAME Se
xyz  0.23 0.45 0.165
occ 1.0
bvalue 20.
!
lambda 2
wavelength 0.977865
```

```
fprimv_mad  -7.5
fprprv_mad  5

lambda 3
wavelength 0.8856
fprimv_mad  -2
fprprv_mad  3.5
!
GENERATE_MAD                        ! generate the MAD dataset now.
!
EOD

solve <<EOD

! NOW DATASET 2 (MIR)
@solve.setup
! change cell slightly for non-isomorphism:
cell 77 27 43 90 102 90
percent_error 1
coordinatefile coords.pdb
iranseed -189753
logfile generate_mir.logfile
solvefile generate_mir.prt

!
derivative 1
label set 1 with 1  pt atom
atomname pt
xyz 0.18 0.53 0.77  ! coords
occ 1.0
bvalue 20.

!
GENERATE_MIR                        ! generate the Pt  MIR  dataset now.
!
EOD
# end of generating mad and mir  datasets
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
solve<<EOD > solve.log
!
!   Take mad and mir datasets that may or may not
!   be exactly isomorphous, combine them into one pseudo-mir dataset
!   and solve it
!
logfile solve.logfile  ! another log file

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym


readformatted        ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged            ! premerged/ unmerged
read_intensities     ! read_intensities/read_amplitudes
fixscattfactors      ! fixscattfactors/refscattfactors


mad_atom se   ! define the anomalously-scattering atom

lambda 1
label set 1 with 2 se atoms, lambda 1
wavelength .9782                  ! wavelength value
fprimv_mad  -10                   ! f' value at this wavelength
fprprv_mad  3                     ! f doubleprime
```

```
rawmadfile lam1.intensities          ! data file

lambda 2
wavelength 0.977865
fprimv_mad  -7.5
fprprv_mad  5
rawmadfile lam2.intensities

lambda 3
wavelength 0.8856
fprimv_mad  -2
fprprv_mad  3.5
rawmadfile lam3.intensities

nres 100              [approx # of residues in protein molecule]
nanomalous 2            [approx # of anomalously scattering atoms per protein]
SCALE_MAD               ! read in and localscale the data
ANALYZE_MAD             ! run MADMRG and MADBST and analyze all the Pattersons

!------------------------end of first dataset -------------

new_dataset

!----------------second dataset (MIR with Pt atoms) ----------

rawnativefile native.intensities
!
derivative 1
label set 1 with 1  pt atoms, deriv 1
rawderivfile der1.intensities
atomname pt

nsolsite 1
scale_native
scale_mir
analyze_mir

!---------------------------end of second dataset --------------

!  combine the datasets into one now...

combine
```

```
!------ go -------
solve
!--------all done----------

EOD
#
# Now run Resolve to do density modification
#
resolve << EOD > resolve.log
!solvent_content 0.4        !    solvent fraction
EOD
#
#  That's it! Now resolve.mtz has your updated phases
#  and resolve.pdb has your model
#
echo 'All done.'
```

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*****************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*****************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 6)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: SOLVE 06-Apr-05

Space group number is: 5
Space group name from file name is: c2
Rescaling standard dataset to put it on approximate absolute scale. NRES =
100; <F**2> expected = 98000.00
; <F**2> observed in lowest resolution shell = 429063.8
... Scale factor = 0.2284043

*** Analysis of this scaled MAD data set ***

Fbar,sigma,Delano,sigma for 3 wavelengths written to:
mad_fbar.scl

F+,sigma,F-,sigma for 3 wavelengths written to:
mad_fpfm.scl

** Completeness of Fbar data at each wavelength: **

Completeness of dataset 1 (F > 2.000000 * sigma)
set 1 with 2 se atoms, lambda 1

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	316	99.7
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1821	99.9

Completeness of dataset 2 (F > 2.000000 * sigma)
set 2

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	238	99.6
2	4.500	317	316	99.7
3	4.200	120	120	100.0
4	3.975	120	119	99.2
5	3.750	155	154	99.4
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1818	99.8

Completeness of dataset 3 (F > 2.000000 * sigma)
set 3

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	316	99.7
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0

8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1821	99.9

** R-factors for F-bar data dispersive differences **

Dispersive differences lambda 2 - lambda 1
(Delta f-prime = 2.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	233	216.346	0.030	1.000	0.00	9.51	0.00
2	4.500	314	204.303	0.033	1.000	2.95	8.36	0.35
3	4.200	119	196.437	0.031	1.001	0.00	8.14	0.00
4	3.975	118	168.250	0.031	1.001	1.39	7.10	0.20
5	3.750	153	169.673	0.032	1.000	1.17	7.02	0.17
6	3.600	111	151.201	0.029	0.999	0.00	6.16	0.00
7	3.450	142	129.803	0.031	1.000	0.00	5.34	0.00
8	3.300	170	128.771	0.032	1.000	0.00	5.60	0.00
9	3.150	189	124.560	0.031	1.000	0.00	5.14	0.00
10	3.000	250	115.008	0.031	1.000	0.36	4.73	0.08
Total:		1799	162.951	0.031	1.000	0.00	7.02	0.10

Recommended resolution cut-off = 3.45

Dispersive differences lambda 3 - lambda 1
(Delta f-prime = 8.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	239	225.298	0.051	1.000	10.22	10.12	1.01
2	4.500	314	204.612	0.051	0.999	10.14	8.37	1.21
3	4.200	120	198.044	0.050	1.001	9.23	8.18	1.13
4	3.975	120	170.529	0.053	1.000	8.94	7.35	1.22
5	3.750	154	171.697	0.045	0.999	7.07	7.09	1.00
6	3.600	112	152.178	0.052	1.000	7.52	6.27	1.20
7	3.450	143	130.124	0.061	1.000	8.18	5.44	1.50
8	3.300	171	129.692	0.048	1.000	5.35	5.66	0.95
9	3.150	190	125.220	0.057	1.000	7.21	5.09	1.42
10	3.000	254	117.322	0.055	1.000	6.17	4.86	1.27
Total:		1817	165.209	0.052	1.000	8.30	7.18	1.19

Recommended resolution cut-off = 3.00

Dispersive differences lambda 3 - lambda 2
(Delta f-prime = 5.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	237	222.268	0.041	1.000	5.04	10.14	0.50
2	4.500	313	204.076	0.041	0.999	6.02	8.37	0.72
3	4.200	119	196.034	0.035	0.999	3.04	8.23	0.37
4	3.975	115	165.487	0.039	1.000	3.10	7.00	0.44
5	3.750	152	169.293	0.037	0.999	3.46	7.06	0.49
6	3.600	112	152.259	0.038	1.000	3.86	6.19	0.62
7	3.450	142	130.330	0.049	1.000	5.73	5.38	1.06
8	3.300	171	128.483	0.039	1.001	3.37	5.52	0.61
9	3.150	190	125.312	0.042	1.001	4.30	5.21	0.83
10	3.000	253	116.076	0.042	1.000	3.60	4.83	0.74
Total:		1804	163.721	0.040	1.000	4.50	7.15	0.66

Recommended resolution cut-off = 3.00

Anomalous differences lambda 1 (f" = 3.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	233	217.075	0.047	1.002	3.40	13.11	0.26
2	4.500	309	202.511	0.050	1.002	6.32	11.46	0.55
3	4.200	118	194.140	0.047	1.001	4.58	10.99	0.42
4	3.975	117	166.611	0.047	1.001	3.74	9.71	0.39
5	3.750	151	168.301	0.046	1.000	3.07	9.65	0.32
6	3.600	107	145.595	0.042	1.001	0.00	8.28	0.00
7	3.450	139	126.844	0.051	1.000	4.62	7.07	0.65
8	3.300	167	124.470	0.055	1.000	5.20	7.15	0.73
9	3.150	187	124.637	0.054	1.000	4.89	7.08	0.69
10	3.000	249	113.907	0.049	1.000	2.96	6.51	0.45
Total:		1777	161.304	0.049	1.001	4.41	9.57	0.47

Recommended resolution cut-off = 3.00

Anomalous differences lambda 2 (f" = 5.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	227	209.685	0.055	1.003	9.38	12.52	0.75
2	4.500	307	202.214	0.053	1.001	7.38	11.50	0.64
3	4.200	114	190.467	0.049	1.000	4.91	10.90	0.45
4	3.975	117	169.812	0.051	1.000	5.79	9.92	0.58
5	3.750	150	165.794	0.058	0.999	7.47	9.42	0.79
6	3.600	111	150.582	0.062	1.000	9.20	8.54	1.08
7	3.450	138	128.009	0.061	1.000	7.39	7.15	1.03
8	3.300	166	124.845	0.065	1.002	6.87	7.25	0.95
9	3.150	181	120.746	0.056	1.001	5.36	6.80	0.79
10	3.000	250	113.803	0.070	1.002	7.85	6.46	1.22
Total:		1761	159.874	0.057	1.001	7.43	9.45	0.84

Recommended resolution cut-off = 3.00

Anomalous differences lambda 3 (f" = 3.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	228	208.951	0.046	1.003	2.90	12.45	0.23
2	4.500	308	202.227	0.047	1.000	4.31	11.54	0.37
3	4.200	116	190.676	0.051	1.001	5.40	10.82	0.50
4	3.975	114	161.262	0.049	1.000	4.24	9.27	0.46
5	3.750	151	170.100	0.046	1.000	2.31	9.71	0.24
6	3.600	110	148.607	0.053	1.001	5.27	8.43	0.62
7	3.450	137	125.523	0.054	1.001	5.22	6.99	0.75
8	3.300	167	124.397	0.056	1.002	5.07	7.17	0.71
9	3.150	186	123.744	0.055	1.001	5.56	6.94	0.80
10	3.000	248	114.240	0.056	1.001	4.82	6.54	0.74
Total:		1765	159.648	0.050	1.001	4.55	9.42	0.53

Recommended resolution cut-off = 3.00

ANALYZE_MAD: Run MADMRG and MADBST on MAD data to get ready for SOLVE

Correlation of anomalous differences at different wavelengths.
(You should probably cut your data off at the resolution where
this drops below about 0.3. A good dataset has correlation
between peak and remote of at least 0.7 overall. Data with

correlations below about 0.5 probably are not contributing much.)

DMIN	CORRELATION FOR WAVELENGTH PAIRS 1 VS 2	1 VS 3	2 VS 3
6.00	0.22	0.12	0.30
4.50	0.23	0.30	0.40
4.20	0.23	0.30	0.30
3.98	0.39	0.19	0.37
3.75	0.39	0.27	0.38
3.60	0.44	0.33	0.46
3.45	0.41	0.30	0.48
3.30	0.48	0.27	0.36
3.15	0.41	0.27	0.53
3.00	0.44	0.36	0.45
ALL	0.32	0.26	0.39

Final refined values of fprime and fdoubleprime

Form factors at lambda = 0.9782
f-prime = -10.00
f" = 3.00

Form factors at lambda = 0.9779
f-prime = -7.50
f" = 5.00

Form factors at lambda = 0.8856
f-prime = -2.00
f" = 3.50

Fa Patterson from MADBST to be written to:
patterson.patt

Script file suitable for running SOLVE written to:
solve_mad.script

Datafile for SOLVE with MADMRG-compressed dataset
("Fnat",sig,"Fder",sig,"Delano",sig,iso diffs, ano diffs,
<Fhcos>, <Fhsin> from MADBST) is:
solve.data

----------NEW DATASET BEGINS HERE----------

```
Rescaling standard dataset to put it on approximate absolute scale.  NRES =
       100;  <F**2> expected =    98000.00
;  <F**2> observed in lowest resolution shell =    432624.4
  ... Scale factor =   0.2265244
Reflections observed:

                Possible  Found     % complete
shell     dmin

   1     6.000       239       238      99.6
   2     4.500       317       317     100.0
   3     4.200       120       120     100.0
   4     3.975       120       120     100.0
   5     3.750       155       155     100.0
   6     3.600       112       112     100.0
   7     3.450       143       143     100.0
   8     3.300       172       172     100.0
   9     3.150       190       189      99.5
  10     3.000       254       212      83.5

total              1822      1778      97.6

SCALE_MIR for dataset           1
Scale derivatives to previously-scaled native.

Default of "fp_or_fm" ( use either F+ or
F- if available) will be used as this flag was not set

Analysis of this MIR dataset.

Fnative, sigma, and (Fbar,sigma, delano,sig) for           1
 derivatives written to:
mir_fbar.scl


Fnative, sigma, and (F+,sigma,F-,sig) for           1 derivatives written to:
mir_fpfm.scl



 ** Completeness of native data (F >    2.000000     * sigma)

Reflections observed:

                Possible  Found     % complete
shell     dmin

   1     6.000       239       238      99.6
   2     4.500       317       317     100.0
   3     4.200       120       120     100.0
   4     3.975       120       120     100.0
   5     3.750       155       155     100.0
   6     3.600       112       112     100.0
   7     3.450       143       143     100.0
```

8	3.300	172	172	100.0
9	3.150	190	189	99.5
10	3.000	254	212	83.5
total		1822	1778	97.6

*** Analysis of this scaled deriv data set ***

** Completeness of Fbar data for each derivative: **

Derivative 1
set 1 with 1 pt atoms, deriv 1

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	238	99.6
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	212	83.5
total		1822	1779	97.6

** R-factors for F-bar data isomorphous differences **

isomorphous differences derivs 1 - native

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	237	225.397	0.254	1.004	68.87	11.65	5.91
2	4.500	317	204.631	0.236	1.006	58.52	10.04	5.83
3	4.200	120	195.215	0.207	1.008	50.13	9.72	5.16
4	3.975	119	165.581	0.254	1.004	50.32	8.36	6.02
5	3.750	155	163.990	0.249	1.000	48.83	8.13	6.00
6	3.600	112	149.516	0.243	1.004	42.88	7.42	5.78
7	3.450	143	135.230	0.257	0.999	42.19	6.46	6.54
8	3.300	172	130.340	0.270	1.001	42.73	6.63	6.44
9	3.150	189	121.253	0.264	1.003	38.87	5.91	6.58

10	3.000	211	114.502	0.262	1.004	36.19	5.71	6.33
Total:		1775	164.741	0.248	1.003	50.46	8.47	6.08

Recommended resolution cut-off = 3.00

** R-factors for anomalous differences **

anomalous differences deriv 1

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	232	215.593	0.047	1.000	7.46	11.74	0.64
2	4.500	311	197.708	0.044	1.002	5.10	10.52	0.48
3	4.200	119	194.770	0.041	1.002	3.16	10.22	0.31
4	3.975	116	168.877	0.039	1.001	2.61	8.93	0.29
5	3.750	153	160.807	0.038	1.000	0.00	8.57	0.00
6	3.600	111	153.367	0.045	1.000	4.38	8.05	0.54
7	3.450	140	123.909	0.042	1.000	2.47	6.50	0.38
8	3.300	169	133.775	0.041	0.999	2.42	7.14	0.34
9	3.150	187	118.543	0.039	1.000	0.80	6.27	0.13
10	3.000	208	111.083	0.039	1.000	0.00	6.01	0.00
Total:		1746	161.012	0.042	1.001	3.85	8.82	0.32

Recommended resolution cut-off = 3.87

Script file suitable for running SOLVE written to:
solve_mir.script

Combining a total of 1 MIR and 1
MAD datasets to form a composite dataset

----------NEW DATASET BEGINS HERE----------

**** SOLVE: Solutions to MIR or SIR datasets ******

Derivatives considered: 3 (NSET)
Cross-vectors tested in HASSP: 6 (ICRMAX, DEFAULT=20)
HASSP solutions saved per deriv: 30 (NTOPHASSP, DEFAULT=30)
Fourier peaks saved per map: 30 (NTOPFOUR, DEFAULT=10)
Sites per derivative: 2 (NSOLSITE, DEFAULT=20)
Derivative solutions per seed: 5 (NTOPDERIV, DEFAULT=5)
Seeds per derivative tested: 3 (NSEEDTEST,DEFAULT=10)

```
Sorted seeds to use                 5  (NSEEDSOLVE, DEFAULT=5)
Number of final solutions saved:     5  (NTOPSOLVE, DEFAULT=5)




 Sites per derivative vary with derivative.
Derivative   Max sites
    1            2

    2           -1

    3            1


Solutions obtained will be compared to input solution (ICHECKSOLVE)
Correlated phasing used (CORRELPHASE)
Patterson map for derivative            1 will be read directly from:
patterson.patt

For derivative           1 the heavy atom
 structure factor components parallel to and
perpendicular to the native structure factor will be read from columns
         9 and          10
Standard difference fouriers will be calculated for derivative           2
Standard difference fouriers will be calculated for derivative           3
For derivative           3 the corresponding native data
will be read from columns          11 and          12
For derivative           3 the corresponding native dataset is "derivative"
         2

Datafile with          16 columns of data:
Title:solve.data          (cols 1 to 10) and mir_fbar.scl
Data: madmrg: MOCK FNAT
Data: madmrg: MOCK sig FNAT
Data: madmrg: MOCK FDER
Data: madmrg: MOCK sig FDER
Data: madmrg: MOCK DEL ANO
Data: madmrg: MOCK sig DEL ANO
Data: madmrg: Del iso for Patterson
Data: madmrg: Sigma of del iso for Patterson
Data: <Fa cos theta> = Fa component along Fo weighted by fom
Data: <Fa sin theta> = weighted Fa component perpendicular to Fo
Data: Native F data
Data: Native sigma of F data
Data: Fbar               set 1 with 1  pt atoms, deriv 1
Data: Sig of Fbar         set 1 with 1  pt atoms, deriv 1
Data: Del Ano (F+ - F-)   set 1 with 1  pt atoms, deriv 1
Data: sig of Del Ano      set 1 with 1  pt atoms, deriv 1

Fnat,sigma taken from columns          1          2
Fder,sig,Delano,sig deriv           1 from cols:          3          4
         5          6
Fder,sig,Delano,sig deriv           2 from cols:         11         12
```

```
        0         0
Fder,sig,Delano,sig deriv          3 from cols:         13         14
        15         16

Check solution to be compared to all solutions found:

Derivative          1:
 Site    X       Y      Z
   1   0.440   0.160   0.380
   2   0.230   0.450   0.165
   3   0.180   0.530   0.770

Derivative          2:
 Site    X       Y      Z

Derivative          3:
 Site    X       Y      Z
   1   0.180   0.530   0.770

*************************************************************

ANALYZE_SOLVE:  analysis of top          1 solutions



*************************************************************


Solution          1

********************** Analysis of this solution *************

 ******    Analysis of non-randomness of native Fourier map ******

A.  Maps with distinct solvent regions havea high standard deviation
of local r.m.s. electron density.
For this map the SD of this local r.m.s. is   0.2787053


B.  Maps with distinct solvent regions also have a high correlation
of local r.m.s. electron density with
density at neighboring locations. Typical
values for poor maps in this
 structure solution are   0.1283073     +/-   7.4281111E-02
For this map the correlation of r.m.s. density
in neighboring boxes is   0.2468866
The correlation coefficient is used here in scoring.

Skew of the map is:   0.5446171

******  Analysis of derivative solutions with the difference Patterson ******
  and with cross-validation difference Fouriers
```

```
-----------------------------------------------
    Derivative #          1



List of sites analyzed for compatibility with difference Patterson

 PEAK        X         Y         Z     OPTIMIZED
                                       RELATIVE OCCUPANCY
   1     0.438     0.167     0.385      80.505
   2     0.229     0.444     0.167      79.514

Evaluation of this test soln with    2 sites after optimizing
occupancy of each site

Cross-vectors for sites  1 and  1 (excluding origin; 1000 = 1 sigma):
 #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
  1   -0.875   0.000  -0.771   10791.8     12962.0         2

Cross-vectors for sites  2 and  1 (excluding origin; 1000 = 1 sigma):
 #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
  1   -0.208   0.278  -0.219   10796.0     6401.26         1
  2   -0.667   0.278  -0.552   10793.2     6401.26         1

Cross-vectors for sites  2 and  2 (excluding origin; 1000 = 1 sigma):
 #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
  1   -0.458   0.000  -0.333   10422.8     12645.0         2

Overall quality of this Patterson soln =  11667.1
Overall quality of the fit to patterson =  3.10646
Avg normalized peak height =  5217.70


Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

 Site    x       y       z       occ       B     -- PEAK  HEIGHT --

   1   0.439   0.160   0.381   0.490  18.823            16.54
   2   0.229   0.450   0.165   0.417  21.490            16.40

-----------------------------------------------
    Derivative #          2



Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

 Site    x       y       z       occ       B     -- PEAK  HEIGHT --


-----------------------------------------------
    Derivative #          3
```

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.318	0.000	0.229	112.715

Evaluation of this test soln with 1 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.635	0.000	-0.458	25409.5	25409.5	2

Overall quality of this Patterson soln = 8983.61
Overall quality of the fit to patterson = 0.248592E-04
Avg normalized peak height = 6352.37

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.320	0.000	0.230	0.499	25.316	22.48

Summary of scoring for this solution:

Criteria	-- over many solutions-- MEAN	SD	-- this solution -- VALUE	Z-SCORE
Pattersons:	4.30	3.30	12.6	2.52
Cross-validation Fourier:	17.5	0.500	42.0	49.0
NatFourier CCx100:	12.8	7.43	24.7	1.60
Mean figure of meritx100:	0.000E+00	18.9	74.1	3.92
Correction for Z-scores:			-23.9	

Overall Z-score value: 33.1

****** Overall analysis of phasing for solution 1**********

HEAVY: Refine heavy atom parameters

File title: SOLVE 06-Apr-05
CRYSTALLOGRAPHIC PARAMETERS

A = 76.00 B = 28.00 C = 42.00 alpha = 90.00 beta = 103.00 gamma = 90.00

PHASES CALCULATED EVERY 5 DEGREES
RESIDUALS CALCULATED ON EXTRA ZEROTH CYCLE ONLY
SIGMAS FROM data FILE WILL BE USED
STATISTICS WILL BE PRINTED ON ZEROTH CYCLE, SHIFTS ON LAST

```
PHASING WILL BE DONE TAKING INTO ACCOUNT THE CORRELATIONS AMONG DERIVATIVES
THE GROUPS OF DERIVATIVES WITH CORRELATIONS WILL BE UPDATED
THE BETA VALUES FOR EACH DERIV  WILL BE SET TO 1.0
PHASE-AVERAGED RESIDUALS WILL BE USED FOR PHASING

TYPE OF REFINEMENT SELECTED:
UNPHASED ORIGIN-REMOVED  PATTERSON REFINEMENT ONLY

Bayesian Correlated Phasing will be used

RESOLUTION LIMITS IN ANGSTROMS:      3.000    20.000
MINIMUM RATIO OF FDER TO RMS LACK-OF-CLOSURE FOR  INCLUSION IN REFINEMENT OR PHASING=      0.000
MINIMUM NATIVE F:      0.000
MINIMUM FIGURE OF MERIT FOR PHASED REFINEMENT:      0.000
MINIMUM ALLOWED ISOTROPIC B:      0.000
PARAMETER SHIFTS GREATER THAN     0.0000 TIMES SIGMA  WILL BE SCALED BY     0.5000
MINIMUM RATIO OF FNAT/SIGMA OR FDER/SIGMA TO INCLUDE:      1.000

NUMBER OF REFINEMENT CYCLES IS      2
DERIVATIVES REFINED DURING THESE CYCLES ARE :  0  0
TYPE OF OUTPUT SELECTED IS: +10 COLUMNS OF HENDRICKSON-LATTMAN COEFFICIENTS
1 INPUT data FILE WITH  16 COLUMNS IS:
combine.scl_1_2
COLUMN  0 : solve.data          (cols 1 to 10) and mir_fbar.scl        ,cols 1 to
COLUMN  1 : madmrg: MOCK FNAT
COLUMN  2 : madmrg: MOCK sig FNAT
COLUMN  3 : madmrg: MOCK FDER
COLUMN  4 : madmrg: MOCK sig FDER
COLUMN  5 : madmrg: MOCK DEL ANO
COLUMN  6 : madmrg: MOCK sig DEL ANO
COLUMN  7 : madmrg: Del iso for Patterson
COLUMN  8 : madmrg: Sigma of del iso for Patterson
COLUMN  9 : <Fa cos theta> = Fa component along Fo weighted by fom
COLUMN 10 : <Fa sin theta> = weighted Fa component perpendicular to Fo
COLUMN 11 : Native F data
COLUMN 12 : Native sigma of F data
COLUMN 13 : Fbar               set 1 with 1  pt atoms, deriv 1
COLUMN 14 : Sig of Fbar         set 1 with 1  pt atoms, deriv 1
COLUMN 15 : Del Ano (F+ - F-)   set 1 with 1  pt atoms, deriv 1
COLUMN 16 : sig of Del Ano      set 1 with 1  pt atoms, deriv 1


data COLUMNS FOR NATIVE F AND SIGMA:     1    2
data COLUMNS FOR BEST AND MOST PROB PHASES AND FIGURE OF MERIT:     0    0    0
OVERALL SCALE FACTOR FOR ALL data =     1.000
SCALE FACTOR FOR NATIVE SIGMAS =     1.000


DERIVATIVE INFORMATION FOR  3 COMPOUNDS


COMPOUND 1     set 1 with 1  pt atoms, deriv 1
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)    3    4    5    6
THIS DERIVATIVE WILL BE USED IN PHASING
```

```
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 2     Native from dataset # 2 (an MIR set) used as a deriv.
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)   11   12    0    0
THIS DERIVATIVE WILL BE USED IN PHASING
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 3     set 1 with 1  pt atoms, deriv 1
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)   13   14    0    0
THIS DERIVATIVE WILL BE USED IN PHASING
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     0.947*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
CARRYING OUT STANDARD REFINEMENT
Total of           2 cycles will be done
Derivs refined will be           0          0

SUMMARY OF RESULTS ON FINAL CYCLE:

NUMBER OF REFLECTIONS READ =    1822
NUMBER OF F .GT. FMIN      =    1817
NUMBER OF F IN RES. LIMITS =    1817
NUMBER OF F .GT. MIN <M>   =    1815
NUMBER OF F USED TO REFINE =       0



FIGURE OF MERIT <       0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1.0
# OF REFLECTIONS        40    56    57    62   107    97   152   259   394   592

FIGURE OF MERIT WITH RESOLUTION
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
N:             1816    105    152    202    219    247    278    295    318
MEAN FIG MERIT:   0.74   0.81   0.80   0.75   0.71   0.75   0.74   0.72   0.72





COMPOUND  1          set 1 with 1  pt atoms, deriv 1
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

CENTRIC REFLNS:   262.    32.    36.    31.    31.    36.    34.    33.    29.
RMS HA F:        33.7   46.6   39.1   39.4   29.2   33.9   27.6   21.6   23.8
RMS RESIDUAL:     30.2   38.9   25.4   15.2   25.0   19.6   55.6   18.6   18.2
```

```
RMS(FH)/RMS(E):   1.12   1.20   1.54   2.59   1.17   1.73   0.50   1.16   1.31
CENTRIC R FACT:   0.45   0.43   0.44   0.36   0.49   0.49   0.56   0.51   0.26

ACENTRIC REFLN:  1543.    72.   116.   170.   188.   211.   241.   256.   289.
RMS DERIV FPH:   194.0  318.1  221.4  236.9  234.6  197.9  159.2  146.7  136.0
RMS SIGMA FPH:    23.3   48.8   22.5   30.0   26.1   24.4   17.0   16.6   14.8
RMS SIGMA FP:     23.5   49.0   22.8   30.3   26.3   24.6   17.2   16.8   14.9
RMS HA F:         30.5   44.8   39.6   37.0   33.0   29.9   28.0   24.6   22.2
RMS RESIDUAL:     25.4   38.8   29.4   35.3   29.3   26.3   19.8   18.9   16.7
RMS(FH)/RMS(E):   1.20   1.15   1.34   1.05   1.12   1.14   1.42   1.30   1.33

ANOM DIFFS:      1543.    72.   116.   170.   188.   211.   241.   256.   289.
RMS OBS DIFF:     11.1   14.1   14.5   13.0   11.8   10.6   10.6    9.3    9.0
RMS CALC DIFF:     9.1   11.9   11.1   10.6    9.5    9.0    8.9    8.0    7.1
RMS RESIDUAL:      7.7   11.5    9.2    9.0    9.0    7.3    6.8    6.1    6.3
RATIO ISO/ANO:    4.65   5.15   4.99   4.83   4.68   4.56   4.44   4.33   4.23

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:            0.0    0.0    0.0    0.0    0.0   52.6   10.7   12.6
ANOMALOUS LOC:          0.0    0.0    0.0    0.0    0.0    0.0    0.0    1.7
RMS FPH :             318.1  221.4  236.9  234.6  197.9  159.2  146.7  136.0
RMS FH  :              44.8   39.6   37.0   33.0   29.9   28.0   24.6   22.2
RMS SIGMA:             69.1   32.0   42.6   37.0   34.7   24.2   23.7   21.0
```

```
COMPOUND  2          Native from dataset # 2 (an MIR set) used as a deriv.
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

CENTRIC REFLNS:   260.    31.    36.    31.    31.    36.    34.    32.    29.
RMS HA F:         0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS RESIDUAL:     68.8   86.5   57.1   83.2   70.2   73.1   58.0   66.3   47.5
RMS(FH)/RMS(E):   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
CENTRIC R FACT:   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00

ACENTRIC REFLN:  1512.    73.   116.   171.   188.   211.   244.   262.   247.
RMS DERIV FPH:   194.0  320.8  216.3  235.8  230.9  192.7  161.5  149.0  133.8
RMS SIGMA FPH:     6.9   11.3    7.6    8.4    8.1    6.8    5.7    5.2    4.7
RMS SIGMA FP:     23.6   48.7   22.8   30.2   26.3   24.6   17.1   16.7   14.6
RMS HA F:         0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS RESIDUAL:     60.6   59.8   52.3   71.2   76.5   60.3   53.9   55.1   54.7
RMS(FH)/RMS(E):   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:           78.2   53.4   78.1   66.8   69.8   56.2   65.2   46.4
RMS FPH :             320.8  216.3  235.8  230.9  192.7  161.5  149.0  133.8
RMS FH  :              0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS SIGMA:             50.0   24.0   31.4   27.5   25.6   18.1   17.5   15.4
```

```
COMPOUND  3         set 1 with 1  pt atoms, deriv 1
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

CENTRIC REFLNS:   261.    31.    36.    31.    31.    36.    34.    33.    29.
RMS HA F:        70.4   95.2   87.2   74.5   71.7   62.8   53.6   55.8   47.3
RMS RESIDUAL:     71.8   78.9   66.2   80.4   76.0   73.3   62.6   79.9   52.7
RMS(FH)/RMS(E):   0.98   1.21   1.32   0.93   0.94   0.86   0.86   0.70   0.90
CENTRIC R FACT:   0.57   0.49   0.57   0.58   0.61   0.54   0.60   0.61   0.57

ACENTRIC REFLN:  1512.    73.   116.   171.   188.   211.   244.   262.   247.
RMS DERIV FPH:   204.8  333.4  238.2  251.6  240.7  200.8  170.4  157.8  140.5
RMS SIGMA FPH:     5.1    8.5    5.9    6.3    6.0    5.1    4.3    3.9    3.5
RMS SIGMA FP:     23.6   48.7   22.8   30.2   26.3   24.6   17.1   16.7   14.6
RMS HA F:        64.7   91.9   86.8   78.4   68.7   64.2   56.5   52.6   46.3
RMS RESIDUAL:     59.4   57.5   51.9   70.3   75.1   59.8   52.1   53.0   54.3
RMS(FH)/RMS(E):   1.09   1.60   1.67   1.11   0.91   1.07   1.09   0.99   0.85

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:           70.1   63.2   75.4   73.0   70.1   61.1   79.1   51.7
RMS FPH :             333.4  238.2  251.6  240.7  200.8  170.4  157.8  140.5
RMS FH  :              91.9   86.8   78.4   68.7   64.2   56.5   52.6   46.3
RMS SIGMA:             49.5   23.5   30.9   27.0   25.2   17.6   17.2   15.0

Analysis of correlated modeling and non-isomorphism errors
obtained using phased residuals.
The derivatives were grouped into 2 sets  where the members of a set
had some mutual correlation.


Set 1 contains derivatives  1

Set 2 contains derivatives  2 3

SUMMARY OF CORRELATED ERRORS AMONG DERIVATIVES

DERIVATIVE:          1
CENTRIC REFLECTIONS:
DMIN:          ALL      9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
RMS errors correlated and uncorrelated with others in group:
    Correlated:    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  Uncorrelated:   19.8    0.0    0.0    0.0    0.0    0.0   52.6   10.7   12.6

Correlation of errors with other derivs:
DERIV 2:         0.14   0.36   0.00   0.09   0.17   0.11   0.00   0.35   0.12
DERIV 3:         0.13   0.31   0.00   0.09   0.25   0.09   0.00   0.25   0.15
```

```
ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
Errors correlated and uncorrelated with others in group:
    Correlated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  Uncorrelated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 2:         0.27   0.31   0.36   0.37   0.31   0.17   0.25   0.22   0.24
DERIV 3:         0.25   0.26   0.32   0.35   0.30   0.16   0.22   0.25   0.23

DERIVATIVE:          2
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
RMS errors correlated and uncorrelated with others in group:
    Correlated:  62.7   69.6   55.4   76.1   66.7   63.7   51.9   66.4   47.5
  Uncorrelated:  19.0   35.5    0.0   17.8    2.9   28.4   21.5    0.0    0.0

Correlation of errors with other derivs:
DERIV 1:         0.14   0.36   0.00   0.09   0.17   0.11   0.00   0.35   0.12
DERIV 3:         0.88   0.89   0.91   0.98   0.91   0.84   0.79   0.86   0.94

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
Errors correlated and uncorrelated with others in group:
    Correlated:  73.3   35.8   58.5   84.5   95.8   73.6   66.2   68.8   70.9
  Uncorrelated:  27.6   29.7   29.8   32.2   31.9   23.1   27.9   26.9   22.1

Correlation of errors with other derivs:
DERIV 1:         0.27   0.31   0.36   0.37   0.31   0.17   0.25   0.22   0.24
DERIV 3:         0.90   0.88   0.83   0.90   0.92   0.92   0.89   0.91   0.92

DERIVATIVE:          3
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
RMS errors correlated and uncorrelated with others in group:
    Correlated:  62.7   69.6   55.4   76.1   66.7   63.7   51.9   66.5   47.5
  Uncorrelated:  27.8    8.5   30.5    0.0   29.7   29.3   32.1   42.8   20.5

Correlation of errors with other derivs:
DERIV 1:         0.13   0.31   0.00   0.09   0.25   0.09   0.00   0.25   0.15
DERIV 2:         0.88   0.89   0.91   0.98   0.91   0.84   0.79   0.86   0.94

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
Errors correlated and uncorrelated with others in group:
    Correlated:  73.3   35.8   58.5   84.5   95.8   73.6   66.2   68.8   70.9
  Uncorrelated:  21.7   17.9   27.8   28.0   24.5   20.3   19.8   16.3   20.2

Correlation of errors with other derivs:
DERIV 1:         0.25   0.26   0.32   0.35   0.30   0.16   0.22   0.25   0.23
DERIV 2:         0.90   0.88   0.83   0.90   0.92   0.92   0.89   0.91   0.92


PARAMETER SHIFTS FOR DERIV  1 : set 1 with 1  pt atoms, deriv 1
```

```
                SCALE FACTOR     OVERALL B
CURRENT VALUES:        1.0000         0.0000


            SITE  ATOM      OCCUP     X      Y      Z        B


CURRENT VALUES:     1    Se      0.4903  0.4391  0.1600  0.3807   18.8234

CURRENT VALUES:     2    Se      0.4168  0.2291  0.4499  0.1651   21.4900


PARAMETER SHIFTS FOR DERIV  2 : Native from dataset # 2 (an MIR set) used as a deriv.

                SCALE FACTOR     OVERALL B
CURRENT VALUES:        1.0000         0.0000


            SITE  ATOM      OCCUP     X      Y      Z        B


CURRENT VALUES:     1    Se      0.0100  0.0000  0.0000  0.0000    0.0000


PARAMETER SHIFTS FOR DERIV  3 : set 1 with 1  pt atoms, deriv 1

                SCALE FACTOR     OVERALL B
CURRENT VALUES:        0.9468         0.0000


            SITE  ATOM      OCCUP     X      Y      Z        B


CURRENT VALUES:     1    Se      0.4992  0.3198  0.0000  0.2302   25.3165


*************************************************************
*************************************************************


*** Summary of solutions and their relationships
 to each other and to check solution ***


-------------------------------------------------------
solution #          1 with overall quality =   33.07686
Derivative           1 with           2 sites.
Overall scale =    1.000000    and  overall b of   0.0000000E+00
 0.4390541      0.1600000      0.3806592      0.4903179       18.82336
```

```
 0.2291234      0.4498835      0.1651258      0.4168475       21.49002
Derivative            3 with            1 sites.
Overall scale =   0.9468162     and  overall b of   0.0000000E+00
 0.3198311      0.0000000E+00  0.2302399      0.4991650       25.31649
Best match of solution 1 -> solution 2:

-------- solution 1 --------    -------------solution 2 ------
site     x        y        z    site     x        y        z     DIST (A)
Derivative  1
 1    0.439    0.160    0.381    1    0.440    0.160    0.380    0.08
 2    0.229    0.450    0.165    2    0.230    0.450    0.165    0.07
Derivative  3
 1    0.320    0.000    0.230    1    0.320    0.030    0.230    0.84
Comparison of this solution with check solution:
Number of sites in this solution matching check=           3
... and number not matching =           0
by derivative, this is...
Deriv          nsame          ndifferent
         1          2          0
         2          0          0
         3          1          0
All sites in this solution are contained in check soln
```

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
#
#  generate 2 mad datasets
#
solve <<EOD

!  GENERATE 2 MAD DATASETS:

! dataset 1

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym

percent_error 4
coordinatefile coords.pdb
iranseed -199753
logfile generate.logfile
solvefile generate.prt

mad_atom se                          ! define the scattering factors...
!
lambda 1
label set 1 with 2 se atoms, lambda 1
wavelength 0.9782            ! wavelength value
fprimv_mad  -10              ! f' value at this wavelength
fprprv_mad  3
ATOMNAME Se
xyz  0.44 0.16 0.38
occ 1.0
bvalue 20.
ATOMNAME Se
xyz  0.23 0.45 0.165
occ 1.0
bvalue 20.
!
lambda 2
```

```
wavelength 0.977865
fprimv_mad  -7.5
fprprv_mad  5

lambda 3
wavelength 0.8856
fprimv_mad  -2
fprprv_mad  3.5
!
GENERATE_MAD                        ! generate the MAD dataset now.
!
EOD
mv lam1.intensities lam1_se.intensities
mv lam2.intensities lam2_se.intensities
mv lam3.intensities lam3_se.intensities
mv lambda_1.fft dataset_1.fft

solve <<EOD

! NOW DATASET 2 (FE)
@solve.setup
percent_error 2
coordinatefile coords.pdb
iranseed -179753
logfile generate_2.logfile
solvefile generate_2.prt

mad_atom fe                         ! define the scattering factors...
!
lambda 1
label set 1 with 1  fe atoms, lambda 1
wavelength 1.74            ! wavelength value
fprimv_mad  -9             ! f' value at this wavelength
fprprv_mad  2.5             ! f doubleprime value at this wavelength
atomname fe
xyz 0.18 0.53 0.77  ! coords
occ 1.0
bvalue 20.

lambda 2
label Fe PK
wavelength 1.73647
fprimv_mad  -5
```

```
fprprv_mad  4.5

lambda 3
label Fe RM
wavelength 0.978
fprimv_mad  1
fprprv_mad  1.5
!
GENERATE_MAD                    ! generate the FE MAD dataset now.
!
EOD
mv lam1.intensities lam1_fe.intensities
mv lam2.intensities lam2_fe.intensities
mv lam3.intensities lam3_fe.intensities
mv lambda_1.fft dataset_2.fft
# end of generating 2 mad datasets
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
# solve.com  -- take 2 mad datasets that may or may not
#   be exactly isomorphous, combine them into one pseudo-mir dataset
# and solve it
#
solve<<EOD > solve.log
logfile solve.logfile

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym


readformatted        ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged            ! premerged/ unmerged
read_intensities     ! read_intensities/read_amplitudes
fixscattfactors      ! fixscattfactors/refscattfactors

!
!---------first MAD dataset (se atoms)---------
mad_atom se
lambda 1
label set 1 with 2 se atoms, lambda 1
wavelength .9782                    ! wavelength value
fprimv_mad  -10                     ! f' value at this wavelength
fprprv_mad  3                       ! f doubleprime value
rawmadfile lam1_se.intensities      ! datafile
```

```
ATOMNAME Se                          ! we're about to enter data on an atom
lambda 2
wavelength 0.977865
fprimv_mad  -7.5
fprprv_mad  5
rawmadfile lam2_se.intensities

lambda 3
wavelength 0.8856
fprimv_mad  -2
fprprv_mad  3.5
rawmadfile lam3_se.intensities
nres 80
nanomalous 2
!
scale_mad
analyze_mad

!------------------------end of first dataset -------------

new_dataset                  ! tell solve we're about to start a new one

!----------------second MAD dataset (fe atoms) ----------
mad_atom fe
!
lambda 1
label set 1 with 1  fe atoms, lambda 1
wavelength 1.74              ! wavelength value
fprimv_mad  -9               ! f' value at this wavelength
fprprv_mad  2.5              ! f doubleprime value at this wavelength
rawmadfile lam1_fe.intensities
atomname fe

lambda 2
label Fe PK
wavelength 1.73647
fprimv_mad  -5
fprprv_mad  4.5
rawmadfile lam2_fe.intensities

lambda 3
label Fe RM
wavelength 0.978
```

```
fprimv_mad  1
fprprv_mad  1.5
rawmadfile lam3_fe.intensities
nres 80
nanomalous 1
scale_mad
analyze_mad
!--------------------------end of second dataset --------------

!  combine the datasets into one now...

combine

!  solve it...
solve
!--------all done----------

EOD
#
# Now run Resolve to do density modification and build a model
#
resolve << EOD > resolve.log
!solvent_content 0.40        !    solvent fraction
EOD
#
#  That's it! Now resolve.mtz has your updated phases
#  and resolve.pdb has your model
```

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*****************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*****************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 6)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: SOLVE 06-Apr-05

Space group number is: 5
Space group name from file name is: c2
Rescaling standard dataset to put it on approximate absolute scale. NRES = 80; <F**2> expected = 78400.00
; <F**2> observed in lowest resolution shell = 432433.2
... Scale factor = 0.1812997

*** Analysis of this scaled MAD data set ***

Fbar,sigma,Delano,sigma for 3 wavelengths written to:
mad_fbar.scl

F+,sigma,F-,sigma for 3 wavelengths written to:
mad_fpfm.scl

** Completeness of Fbar data at each wavelength: **

Completeness of dataset 1 (F > 2.000000 * sigma)
set 1 with 2 se atoms, lambda 1

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	237	99.2
2	4.500	317	315	99.4
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1818	99.8

Completeness of dataset 2 (F > 2.000000 * sigma)
set 2

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	238	99.6
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	119	99.2
5	3.750	155	154	99.4
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	253	99.6
total		1822	1818	99.8

Completeness of dataset 3 (F > 2.000000 * sigma)
set 3

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	238	99.6
2	4.500	317	315	99.4
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0

8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1819	99.8

** R-factors for F-bar data dispersive differences **

Dispersive differences lambda 2 - lambda 1
(Delta f-prime = 2.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	233	192.039	0.053	0.997	0.00	14.89	0.00
2	4.500	314	182.586	0.060	1.002	6.42	13.09	0.49
3	4.200	118	175.552	0.056	1.003	3.11	12.56	0.25
4	3.975	116	150.026	0.055	1.002	1.03	11.06	0.09
5	3.750	153	151.332	0.059	1.002	4.53	10.98	0.41
6	3.600	111	135.142	0.047	0.999	0.00	9.62	0.00
7	3.450	142	115.315	0.054	1.001	0.00	8.34	0.00
8	3.300	169	113.266	0.056	1.000	0.75	8.54	0.09
9	3.150	189	110.951	0.054	1.001	0.00	7.89	0.00
10	3.000	250	103.016	0.053	1.001	0.00	7.39	0.00
Total:		1795	145.176	0.056	1.001	1.80	10.93	0.15

Recommended resolution cut-off = 3.75

Dispersive differences lambda 3 - lambda 1
(Delta f-prime = 8.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	234	195.008	0.067	0.999	0.00	17.63	0.00
2	4.500	310	179.300	0.070	1.001	6.90	14.46	0.48
3	4.200	120	176.643	0.071	1.003	6.26	14.59	0.43
4	3.975	120	151.557	0.074	1.002	7.56	13.15	0.57
5	3.750	153	152.318	0.060	1.000	0.00	12.80	0.00
6	3.600	111	134.361	0.072	1.000	4.78	11.19	0.43
7	3.450	142	114.752	0.077	1.000	5.92	9.67	0.61
8	3.300	170	114.174	0.070	0.999	3.19	9.92	0.32
9	3.150	190	111.306	0.081	1.000	7.36	9.04	0.81
10	3.000	252	104.085	0.075	1.000	4.87	8.64	0.56
Total:		1802	145.338	0.071	1.000	4.68	12.62	0.42

Recommended resolution cut-off = 3.00

Dispersive differences lambda 3 - lambda 2
(Delta f-prime = 5.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	235	195.883	0.062	1.002	0.00	17.80	0.00
2	4.500	313	181.394	0.061	0.998	0.00	14.87	0.00
3	4.200	118	172.821	0.056	1.000	0.00	14.15	0.00
4	3.975	115	147.979	0.059	1.000	0.00	12.21	0.00
5	3.750	152	151.221	0.060	0.999	0.00	12.46	0.00
6	3.600	112	135.314	0.057	1.001	0.00	10.93	0.00
7	3.450	142	117.095	0.070	1.000	4.25	9.58	0.44
8	3.300	170	114.758	0.062	1.000	0.00	9.94	0.00
9	3.150	190	111.384	0.066	1.001	3.59	8.93	0.40
10	3.000	251	101.874	0.061	0.999	0.00	8.15	0.00
Total:		1798	145.289	0.061	1.000	0.00	12.55	0.08

Recommended resolution cut-off = 3.00

Anomalous differences lambda 1 (f" = 3.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	229	190.512	0.081	1.001	0.00	21.05	0.00
2	4.500	306	179.141	0.086	1.002	9.16	18.39	0.50
3	4.200	117	171.992	0.083	1.001	7.47	17.65	0.42
4	3.975	114	144.091	0.076	1.001	0.00	15.38	0.00
5	3.750	149	147.864	0.080	1.000	3.10	15.39	0.20
6	3.600	108	130.305	0.073	1.001	0.00	13.49	0.00
7	3.450	137	111.783	0.085	1.000	5.39	11.29	0.48
8	3.300	165	110.242	0.088	1.000	5.20	11.63	0.45
9	3.150	185	109.106	0.091	1.000	6.86	11.22	0.61
10	3.000	245	100.255	0.077	1.000	0.00	10.39	0.00
Total:		1755	142.080	0.083	1.001	5.02	15.34	0.28

Recommended resolution cut-off = 3.00

Anomalous differences lambda 2 (f" = 5.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	222	180.184	0.087	1.004	0.00	21.29	0.00
2	4.500	301	176.235	0.080	1.002	0.00	19.97	0.00
3	4.200	114	167.398	0.071	0.999	0.00	19.13	0.00
4	3.975	118	151.609	0.081	0.999	0.00	17.62	0.00
5	3.750	149	145.763	0.088	0.999	0.00	16.48	0.00
6	3.600	109	133.239	0.094	0.999	5.73	15.18	0.38
7	3.450	138	114.444	0.090	1.000	4.11	12.72	0.32
8	3.300	165	109.644	0.095	1.002	1.58	12.63	0.12
9	3.150	182	107.695	0.083	1.001	0.00	12.12	0.00
10	3.000	247	100.684	0.105	1.002	7.60	11.44	0.66
Total:		1745	140.182	0.087	1.001	0.00	16.45	0.15

Recommended resolution cut-off = 3.00

Anomalous differences lambda 3 (f" = 3.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	226	184.079	0.078	1.002	0.00	21.82	0.00
2	4.500	303	177.051	0.081	1.000	0.00	20.25	0.00
3	4.200	115	168.951	0.087	1.003	0.00	19.20	0.00
4	3.975	113	142.748	0.085	1.000	0.00	16.32	0.00
5	3.750	148	148.788	0.076	1.002	0.00	16.78	0.00
6	3.600	108	130.329	0.086	1.002	0.00	14.82	0.00
7	3.450	134	109.136	0.087	1.001	2.08	12.05	0.17
8	3.300	166	109.848	0.092	1.002	4.87	12.60	0.39
9	3.150	187	110.207	0.092	1.000	6.22	12.32	0.50
10	3.000	246	101.345	0.088	1.001	1.44	11.60	0.12
Total:		1746	140.494	0.084	1.001	0.00	16.53	0.12

Recommended resolution cut-off = 3.00

ANALYZE_MAD: Run MADMRG and MADBST on MAD data to get ready for SOLVE

Correlation of anomalous differences at different wavelengths.
(You should probably cut your data off at the resolution where
this drops below about 0.3. A good dataset has correlation
between peak and remote of at least 0.7 overall. Data with

correlations below about 0.5 probably are not contributing much.)

DMIN	CORRELATION FOR WAVELENGTH PAIRS 1 VS 2	1 VS 3	2 VS 3
6.00	0.11	-0.06	0.05
4.50	0.15	0.18	0.22
4.20	0.00	0.10	0.08
3.98	0.14	-0.04	0.13
3.75	0.14	0.04	0.12
3.60	0.20	0.08	0.19
3.45	0.14	0.22	0.17
3.30	0.23	0.09	0.04
3.15	0.17	0.06	0.31
3.00	0.25	0.08	0.19
ALL	0.14	0.08	0.15

Final refined values of fprime and fdoubleprime

Form factors at lambda = 0.9782
f-prime = -10.00
f" = 3.00

Form factors at lambda = 0.9779
f-prime = -7.50
f" = 5.00

Form factors at lambda = 0.8856
f-prime = -2.00
f" = 3.50

Fa Patterson from MADBST to be written to:
patterson.patt

Script file suitable for running SOLVE written to:
solve_mad.script

Datafile for SOLVE with MADMRG-compressed dataset
("Fnat",sig,"Fder",sig,"Delano",sig,iso diffs, ano diffs,
<Fhcos>, <Fhsin> from MADBST) is:
solve.data

----------NEW DATASET BEGINS HERE----------

Rescaling standard dataset to put it on approximate absolute scale. NRES = 80; <F**2> expected = 78400.00
; <F**2> observed in lowest resolution shell = 430408.9
... Scale factor = 0.1821524

*** Analysis of this scaled MAD data set ***

Fbar,sigma,Delano,sigma for 3 wavelengths written to:
mad_fbar.scl_2

F+,sigma,F-,sigma for 3 wavelengths written to:
mad_fpfm.scl

** Completeness of Fbar data at each wavelength: **

Completeness of dataset 1 (F > 2.000000 * sigma)
set 1 with 1 fe atoms, lambda 1

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	316	99.7
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	171	99.4
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1820	99.9

Completeness of dataset 2 (F > 2.000000 * sigma)
Fe PK

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	238	99.6
2	4.500	317	317	100.0
3	4.200	120	119	99.2

4	3.975	120	120	100.0
5	3.750	155	154	99.4
6	3.600	112	111	99.1
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	253	99.6
total		1822	1817	99.7

Completeness of dataset 3 (F > 2.000000 * sigma)
Fe RM

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	316	99.7
3	4.200	120	120	100.0
4	3.975	120	119	99.2
5	3.750	155	154	99.4
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1819	99.8

** R-factors for F-bar data dispersive differences **

Dispersive differences lambda 2 - lambda 1
(Delta f-prime = 4.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	234	195.280	0.042	0.999	0.00	11.82	0.00
2	4.500	313	178.804	0.045	1.002	2.45	10.25	0.24
3	4.200	116	171.340	0.040	1.003	0.00	9.75	0.00
4	3.975	120	154.011	0.043	1.001	0.00	8.93	0.00
5	3.750	152	147.529	0.041	1.001	0.00	8.40	0.00
6	3.600	110	133.788	0.036	1.000	0.00	7.60	0.00
7	3.450	143	111.995	0.043	1.001	0.00	6.60	0.00
8	3.300	168	113.456	0.043	1.000	0.00	6.65	0.00
9	3.150	189	109.306	0.040	1.000	0.00	6.25	0.00
10	3.000	250	100.592	0.043	1.001	0.00	5.69	0.00

Total: 1795 143.756 0.042 1.001 0.00 8.59 0.04

Recommended resolution cut-off = 3.30

Dispersive differences lambda 3 - lambda 1
(Delta f-prime = 10.00000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	237	201.844	0.054	1.000	4.94	12.74	0.39
2	4.500	313	178.453	0.059	1.001	8.32	10.37	0.80
3	4.200	119	175.136	0.051	1.003	4.76	10.19	0.47
4	3.975	119	153.114	0.061	1.000	8.17	8.98	0.91
5	3.750	153	148.150	0.050	0.999	3.13	8.64	0.36
6	3.600	110	133.631	0.054	1.000	4.43	7.81	0.57
7	3.450	143	111.995	0.064	1.001	5.77	6.83	0.84
8	3.300	170	115.066	0.056	1.000	3.84	7.02	0.55
9	3.150	190	109.527	0.064	1.001	6.26	6.35	0.98
10	3.000	253	102.825	0.061	1.001	4.96	6.22	0.80

Total: 1807 145.281 0.057 1.001 5.89 8.95 0.68

Recommended resolution cut-off = 3.00

Dispersive differences lambda 3 - lambda 2
(Delta f-prime = 6.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	235	198.772	0.046	1.001	0.00	12.64	0.00
2	4.500	312	178.411	0.045	0.999	0.00	10.48	0.00
3	4.200	119	173.257	0.045	1.000	2.03	10.06	0.20
4	3.975	119	154.540	0.049	1.000	4.02	8.99	0.45
5	3.750	152	146.118	0.044	0.999	0.00	8.56	0.00
6	3.600	110	134.072	0.042	1.000	0.00	7.77	0.00
7	3.450	141	111.854	0.046	1.000	0.00	6.70	0.00
8	3.300	170	115.147	0.046	1.001	0.00	7.15	0.00
9	3.150	188	108.397	0.049	1.002	2.43	6.35	0.38
10	3.000	251	100.247	0.046	1.001	1.26	5.79	0.22

Total: 1797 144.256 0.046 1.000 0.00 8.90 0.11

Recommended resolution cut-off = 3.00

Anomalous differences lambda 1 (f" = 2.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	229	193.187	0.057	1.000	0.00	15.56	0.00
2	4.500	305	175.425	0.060	1.000	6.43	13.43	0.48
3	4.200	117	172.210	0.060	0.999	4.23	13.12	0.32
4	3.975	115	148.059	0.054	1.002	0.00	11.63	0.00
5	3.750	153	148.036	0.061	1.000	4.25	11.52	0.37
6	3.600	107	130.528	0.055	1.001	0.00	10.01	0.00
7	3.450	136	107.254	0.060	1.000	2.16	8.11	0.27
8	3.300	167	113.203	0.059	1.000	0.00	8.80	0.00
9	3.150	186	107.628	0.061	1.001	2.70	8.28	0.33
10	3.000	244	97.237	0.058	1.000	0.00	7.37	0.00
Total:		1759	141.428	0.059	1.000	2.50	11.31	0.19

Recommended resolution cut-off = 3.98

Anomalous differences lambda 2 (f" = 4.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	227	188.885	0.067	1.001	5.37	15.67	0.34
2	4.500	301	172.899	0.057	1.001	0.00	13.87	0.00
3	4.200	114	167.698	0.052	1.000	0.00	13.54	0.00
4	3.975	117	152.365	0.059	1.000	0.00	12.45	0.00
5	3.750	150	142.807	0.066	1.000	3.65	11.50	0.32
6	3.600	109	132.978	0.057	1.000	0.00	10.63	0.00
7	3.450	139	109.253	0.067	1.000	4.39	8.69	0.51
8	3.300	162	110.604	0.059	1.001	0.00	8.94	0.00
9	3.150	184	106.070	0.061	1.000	0.00	8.53	0.00
10	3.000	244	97.956	0.067	1.001	3.01	7.86	0.38
Total:		1747	139.841	0.061	1.001	0.00	11.64	0.17

Recommended resolution cut-off = 3.75

Anomalous differences lambda 3 (f" = 1.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	227	189.070	0.053	1.003	0.00	15.23	0.00
2	4.500	305	174.238	0.055	1.000	0.00	13.75	0.00
3	4.200	118	171.259	0.059	1.001	2.52	13.40	0.19
4	3.975	116	149.421	0.062	0.999	4.09	11.80	0.35
5	3.750	145	140.756	0.050	1.001	0.00	11.07	0.00
6	3.600	108	130.392	0.054	1.000	0.00	10.23	0.00
7	3.450	134	103.646	0.056	1.001	0.00	8.02	0.00
8	3.300	163	111.202	0.055	1.002	0.00	8.78	0.00
9	3.150	181	104.837	0.054	1.000	0.00	8.20	0.00
10	3.000	247	99.023	0.058	1.002	0.00	7.76	0.00
Total:		1744	139.646	0.055	1.001	0.00	11.36	0.04

Recommended resolution cut-off = 3.15

ANALYZE_MAD: Run MADMRG and MADBST on MAD data to get ready for SOLVE

Final refined values of fprime and fdoubleprime

Form factors at lambda = 1.7400
f-prime = -9.00
f" = 2.50

Form factors at lambda = 1.7365
f-prime = -5.00
f" = 4.50

Form factors at lambda = 0.9780
f-prime = 1.00
f" = 1.50

Fa Patterson from MADBST to be written to:
patterson.patt_2

Script file suitable for running SOLVE written to:
solve_mad.script

Datafile for SOLVE with MADMRG-compressed dataset
("Fnat",sig,"Fder",sig,"Delano",sig,iso diffs, ano diffs,
<Fhcos>, <Fhsin> from MADBST) is:
solve.data_2

Combining a total of 0 MIR and 2

```
 MAD datasets to form a composite dataset


----------NEW DATASET BEGINS HERE---------------

 ****   SOLVE:  Solutions to MIR or SIR datasets  ******

Derivatives considered:             3  (NSET)
Cross-vectors tested in HASSP:       6  (ICRMAX, DEFAULT=20)
HASSP solutions saved per deriv:    30  (NTOPHASSP, DEFAULT=30)
Fourier peaks saved per map:        30  (NTOPFOUR, DEFAULT=10)
Sites per derivative:               2  (NSOLSITE, DEFAULT=20)
Derivative solutions per seed:       5  (NTOPDERIV, DEFAULT=5)
Seeds per derivative tested:         3  (NSEEDTEST,DEFAULT=10)
Sorted seeds to use                  5  (NSEEDSOLVE, DEFAULT=5)
Number of final solutions saved:     5  (NTOPSOLVE, DEFAULT=5)




 Sites per derivative vary with derivative.
Derivative   Max sites
     1            2

     2           -1

     3            1


Solutions obtained will be compared to input solution (ICHECKSOLVE)
Correlated phasing used (CORRELPHASE)
Patterson map for derivative            1 will be read directly from:
patterson.patt

Patterson map for derivative            3 will be read directly from:
patterson.patt_2

For derivative           1 the heavy atom
 structure factor components parallel to and
perpendicular to the native structure factor will be read from columns
         9 and          10
Standard difference fouriers will be calculated for derivative           2
For derivative           3 the heavy atom
 structure factor components parallel to and
perpendicular to the native structure factor will be read from columns
        19 and          20
For derivative           3 the corresponding native data
will be read from columns          11 and          12
For derivative           3 the corresponding native dataset is "derivative"
         2

Datafile with          20 columns of data:
Title:solve.data          (cols 1 to 10) and solve.data_2
Data: madmrg: MOCK FNAT
```

```
Data: madmrg: MOCK sig FNAT
Data: madmrg: MOCK FDER
Data: madmrg: MOCK sig FDER
Data: madmrg: MOCK DEL ANO
Data: madmrg: MOCK sig DEL ANO
Data: madmrg: Del iso for Patterson
Data: madmrg: Sigma of del iso for Patterson
Data: <Fa cos theta> = Fa component along Fo weighted by fom
Data: <Fa sin theta> = weighted Fa component perpendicular to Fo
Data: madmrg: MOCK FNAT
Data: madmrg: MOCK sig FNAT
Data: madmrg: MOCK FDER
Data: madmrg: MOCK sig FDER
Data: madmrg: MOCK DEL ANO
Data: madmrg: MOCK sig DEL ANO
Data: madmrg: Del iso for Patterson
Data: madmrg: Sigma of del iso for Patterson
Data: <Fa cos theta> = Fa component along Fo weighted by fom
Data: <Fa sin theta> = weighted Fa component perpendicular to Fo

Fnat,sigma taken from columns        1        2
Fder,sig,Delano,sig deriv        1 from cols:        3        4
        5        6
Fder,sig,Delano,sig deriv        2 from cols:        11        12
        0        0
Fder,sig,Delano,sig deriv        3 from cols:        13        14
        15        16

Check solution to be compared to all solutions found:

Derivative        1:
 Site    X      Y      Z
   1   0.440   0.160   0.380
   2   0.230   0.450   0.165

Derivative        2:
 Site    X      Y      Z

Derivative        3:
 Site    X      Y      Z
   1   0.180   0.530   0.770

*********************************************************

ANALYZE_SOLVE:  analysis of top        1 solutions



*************************************************************


Solution        1

********************** Analysis of this solution *************
```

****** Analysis of non-randomness of native Fourier map ******

A. Maps with distinct solvent regions havea high standard deviation
of local r.m.s. electron density.
For this map the SD of this local r.m.s. is 0.2271466

B. Maps with distinct solvent regions also have a high correlation
of local r.m.s. electron density with
density at neighboring locations. Typical
values for poor maps in this
structure solution are 6.4755760E-02 +/- 3.4622770E-02
For this map the correlation of r.m.s. density
in neighboring boxes is 0.1684203
The correlation coefficient is used here in scoring.

Skew of the map is: 0.1915549

****** Analysis of derivative solutions with the difference Patterson ******
and with cross-validation difference Fouriers

Derivative # 1

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.438	0.167	0.385	68.084
2	0.229	0.458	0.167	62.690

Evaluation of this test soln with 2 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.875	0.000	-0.771	7757.40	9270.82	2

Cross-vectors for sites 2 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.208	0.292	-0.219	7339.48	4268.16	1
2	-0.667	0.292	-0.552	7571.71	4268.16	1

Cross-vectors for sites 2 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.458	0.000	-0.333	6151.03	7860.00	2

Overall quality of this Patterson soln = 8017.65
Overall quality of the fit to patterson = 2.26086
Avg normalized peak height = 3585.60

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.439	0.160	0.382	0.522	26.244	13.94
2	0.229	0.452	0.167	0.296	9.188	12.59

Derivative # 2

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --

Derivative # 3

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.318	0.000	0.229	89.156

Evaluation of this test soln with 1 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.635	0.000	-0.458	15897.5	15897.5	2

Overall quality of this Patterson soln = 5620.60
Overall quality of the fit to patterson = 0.690534E-06
Avg normalized peak height = 3974.37

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.320	0.000	0.229	0.491	27.986	18.82

Summary of scoring for this solution:

	-- over many solutions--		-- this solution --	
Criteria	MEAN	SD	VALUE	Z-SCORE

```
Pattersons:                    3.22        2.25        8.40        2.30
Cross-validation Fourier:     11.9        0.500        34.4        45.0
NatFourier CCx100:            6.48        3.46        16.8        2.99
Mean figure of meritx100:    0.000E+00    10.7        60.3        5.64
Correction for Z-scores:                                -23.0

Overall Z-score value:                                   32.9


****** Overall analysis of phasing for solution          1************


HEAVY:  Refine heavy atom parameters

File title: SOLVE     06-Apr-05
CRYSTALLOGRAPHIC PARAMETERS

    A =  76.00     B =  28.00     C =  42.00     alpha =  90.00     beta = 103.00     gamma =  90.00

PHASES CALCULATED EVERY   5 DEGREES
RESIDUALS CALCULATED ON EXTRA ZEROTH CYCLE ONLY
SIGMAS FROM data FILE WILL BE USED
STATISTICS WILL BE PRINTED ON ZEROTH CYCLE, SHIFTS ON LAST
PHASING WILL BE DONE TAKING INTO ACCOUNT THE CORRELATIONS AMONG DERIVATIVES
THE GROUPS OF DERIVATIVES WITH CORRELATIONS WILL BE UPDATED
THE BETA VALUES FOR EACH DERIV  WILL BE SET TO 1.0
PHASE-AVERAGED RESIDUALS WILL BE USED FOR PHASING

TYPE OF REFINEMENT SELECTED:
UNPHASED ORIGIN-REMOVED  PATTERSON REFINEMENT ONLY

Bayesian Correlated Phasing will be used

RESOLUTION LIMITS IN ANGSTROMS:      3.000    20.000
MINIMUM RATIO OF FDER TO RMS LACK-OF-CLOSURE FOR  INCLUSION IN REFINEMENT OR PHASING=      0.000
MINIMUM NATIVE F:      0.000
MINIMUM FIGURE OF MERIT FOR PHASED REFINEMENT:      0.000
MINIMUM ALLOWED ISOTROPIC B:      0.000
PARAMETER SHIFTS GREATER THAN     0.0000 TIMES SIGMA  WILL BE SCALED BY     0.5000
MINIMUM RATIO OF FNAT/SIGMA OR FDER/SIGMA TO INCLUDE:      1.000

NUMBER OF REFINEMENT CYCLES IS      2
DERIVATIVES REFINED DURING THESE CYCLES ARE :  0  0
TYPE OF OUTPUT SELECTED IS: +10 COLUMNS OF HENDRICKSON-LATTMAN COEFFICIENTS
1 INPUT data FILE WITH  20 COLUMNS IS:
combine.scl_1_2
COLUMN  0 : solve.data          (cols 1 to 10) and solve.data_2        ,cols 1 to 1
COLUMN  1 : madmrg: MOCK FNAT
COLUMN  2 : madmrg: MOCK sig FNAT
COLUMN  3 : madmrg: MOCK FDER
COLUMN  4 : madmrg: MOCK sig FDER
COLUMN  5 : madmrg: MOCK DEL ANO
COLUMN  6 : madmrg: MOCK sig DEL ANO
COLUMN  7 : madmrg: Del iso for Patterson
```

```
COLUMN  8 : madmrg: Sigma of del iso for Patterson
COLUMN  9 : <Fa cos theta> = Fa component along Fo weighted by fom
COLUMN 10 : <Fa sin theta> = weighted Fa component perpendicular to Fo
COLUMN 11 : madmrg: MOCK FNAT
COLUMN 12 : madmrg: MOCK sig FNAT
COLUMN 13 : madmrg: MOCK FDER
COLUMN 14 : madmrg: MOCK sig FDER
COLUMN 15 : madmrg: MOCK DEL ANO
COLUMN 16 : madmrg: MOCK sig DEL ANO
COLUMN 17 : madmrg: Del iso for Patterson
COLUMN 18 : madmrg: Sigma of del iso for Patterson
COLUMN 19 : <Fa cos theta> = Fa component along Fo weighted by fom
COLUMN 20 : <Fa sin theta> = weighted Fa component perpendicular to Fo


data COLUMNS FOR NATIVE F AND SIGMA:     1    2
data COLUMNS FOR BEST AND MOST PROB PHASES AND FIGURE OF MERIT:     0    0    0
OVERALL SCALE FACTOR FOR ALL data =     1.000
SCALE FACTOR FOR NATIVE SIGMAS =     1.000


DERIVATIVE INFORMATION FOR  3 COMPOUNDS


COMPOUND 1     set 1 with 1  fe atoms, lambda 1
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)    3    4    5    6
THIS DERIVATIVE WILL BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 2     Native from dataset # 2 (a MAD set) used as a deriv.
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)   11   12    0    0
THIS DERIVATIVE WILL BE USED IN PHASING
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 3     Fe RM
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)   13   14   15   16
THIS DERIVATIVE WILL BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
CARRYING OUT STANDARD REFINEMENT
Total of           2 cycles will be done
Derivs refined will be           0          0

SUMMARY OF RESULTS ON FINAL CYCLE:

NUMBER OF REFLECTIONS READ =    1822
```

```
NUMBER OF F .GT. FMIN      =    1784
NUMBER OF F IN RES. LIMITS =    1784
NUMBER OF F .GT. MIN <M>   =    1780
NUMBER OF F USED TO REFINE =       0



FIGURE OF MERIT <       0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1.0
# OF REFLECTIONS        92   128   164   125   147   173   161   190   209   394

FIGURE OF MERIT WITH RESOLUTION
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
N:             1783    102    147    197    216    244    273    290    314
MEAN FIG MERIT:   0.60   0.59   0.64   0.58   0.55   0.58   0.63   0.61   0.63





COMPOUND  1          set 1 with 1  fe atoms, lambda 1
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

CENTRIC REFLNS:   256.    32.    33.    31.    30.    36.    34.    32.    28.
RMS HA F:        30.9   43.2   35.4   36.5   26.4   31.2   25.1   19.1   21.6
RMS RESIDUAL:     34.0   50.5   42.4   28.1   40.3   33.5   21.5   25.5   14.2
RMS(FH)/RMS(E):   0.91   0.86   0.84   1.30   0.66   0.93   1.17   0.75   1.52
CENTRIC R FACT:   0.55   0.50   0.60   0.46   0.65   0.55   0.53   0.58   0.46

ACENTRIC REFLN:  1522.    70.   114.   165.   185.   208.   238.   256.   286.
RMS DERIV FPH:   171.5  282.5  199.4  206.4  208.7  174.5  141.6  129.3  120.7
RMS SIGMA FPH:    39.5   81.3   41.6   45.1   46.5   39.6   34.3   27.8   24.6
RMS SIGMA FP:     39.7   81.8   42.1   45.7   45.6   40.0   34.6   28.1   24.9
RMS HA F:        27.8   41.0   35.8   33.7   29.9   27.3   25.7   22.6   20.3
RMS RESIDUAL:     39.1   57.1   48.3   54.3   42.9   39.1   31.0   29.8   28.2
RMS(FH)/RMS(E):   0.71   0.72   0.74   0.62   0.70   0.70   0.83   0.76   0.72

ANOM DIFFS:      1522.    70.   114.   165.   185.   208.   238.   256.   286.
RMS OBS DIFF:     14.1   19.6   17.9   17.4   16.6   13.0   12.4   11.5   10.9
RMS CALC DIFF:     8.4   11.2    9.9    9.8    8.7    8.4    8.3    7.5    6.6
RMS RESIDUAL:     12.3   17.9   15.3   15.2   15.4   11.4   10.1    9.4    9.2
RATIO ISO/ANO:    4.65   5.15   4.99   4.83   4.68   4.56   4.44   4.33   4.23

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:             0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
ANOMALOUS LOC:           0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS FPH :             282.5  199.4  206.4  208.7  174.5  141.6  129.3  120.7
RMS FH  :              41.0   35.8   33.7   29.9   27.3   25.7   22.6   20.3
RMS SIGMA:            115.4   59.2   64.2   65.1   56.3   48.8   39.5   35.0
```

```
COMPOUND  2          Native from dataset # 2 (a MAD set) used as a deriv.
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

CENTRIC REFLNS:   251.    32.    33.    31.    28.    36.    34.    30.    27.
RMS HA F:         0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS RESIDUAL:     24.3   43.0   23.3   19.8   21.7   28.9   15.6   13.8    8.2
RMS(FH)/RMS(E):   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
CENTRIC R FACT:   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   0.99

ACENTRIC REFLN:  1518.    69.   114.   164.   186.   208.   237.   256.   284.
RMS DERIV FPH:   172.3  282.3  198.4  215.1  207.7  175.2  141.7  129.9  118.7
RMS SIGMA FPH:    16.6   32.1   20.0   21.3   18.3   15.7   13.7   12.3   10.0
RMS SIGMA FP:     39.5   80.1   42.1   45.9   45.5   40.0   34.6   27.7   24.6
RMS HA F:         0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS RESIDUAL:     22.3   28.6   23.4   30.1   25.6   22.3   18.9   18.5   17.7
RMS(FH)/RMS(E):   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:            0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS FPH :             282.3  198.4  215.1  207.7  175.2  141.7  129.9  118.7
RMS FH  :               0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS SIGMA:             86.3   46.6   50.6   49.1   43.0   37.2   30.3   26.6


COMPOUND  3          Fe RM
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

CENTRIC REFLNS:   256.    32.    33.    31.    30.    36.    34.    32.    28.
RMS HA F:         18.2   26.3   22.7   19.4   18.4   15.6   13.2   13.3   11.0
RMS RESIDUAL:     32.9   47.1   36.7   32.3   38.7   35.2   19.9   25.2   14.7
RMS(FH)/RMS(E):   0.55   0.56   0.62   0.60   0.48   0.44   0.66   0.53   0.75
CENTRIC R FACT:   0.63   0.55   0.71   0.60   0.71   0.62   0.62   0.65   0.59

ACENTRIC REFLN:  1521.    69.   114.   165.   186.   208.   238.   256.   285.
RMS DERIV FPH:   170.1  275.9  201.1  206.1  207.5  174.1  140.4  129.5  116.5
RMS SIGMA FPH:    16.3   31.8   19.8   21.0   18.0   15.5   13.5   12.1    9.8
RMS SIGMA FP:     39.4   80.1   42.1   45.8   45.5   40.0   34.6   27.7   24.6
RMS HA F:         16.4   25.6   23.5   20.6   17.6   16.0   13.7   12.6   10.9
RMS RESIDUAL:     35.9   50.1   44.1   50.6   39.1   35.5   28.3   27.5   27.4
RMS(FH)/RMS(E):   0.46   0.51   0.53   0.41   0.45   0.45   0.49   0.46   0.40

ANOM DIFFS:      1521.    69.   114.   165.   186.   208.   238.   256.   285.
RMS OBS DIFF:     11.4   15.2   14.5   15.3   13.9   10.7    9.3    8.4    8.1
RMS CALC DIFF:     5.7    8.2    7.7    6.7    6.2    5.7    5.0    4.6    4.2
```

```
RMS RESIDUAL:     10.5   13.7   13.2   14.8   13.1    9.7    8.2    7.7    7.3
RATIO ISO/ANO:    4.03   4.50   4.34   4.19   4.05   3.93   3.83   3.73   3.64

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:            0.0    0.0    0.0    0.0    0.0    0.0   13.6    0.0
ANOMALOUS LOC:          0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS FPH :             275.9  201.1  206.1  207.5  174.1  140.4  129.5  116.5
RMS FH  :              25.6   23.5   20.6   17.6   16.0   13.7   12.6   10.9
RMS SIGMA:             86.2   46.5   50.4   49.0   42.9   37.1   30.2   26.5

Analysis of correlated modeling and non-isomorphism errors
obtained using phased residuals.
The derivatives were grouped into 2 sets  where the members of a set
had some mutual correlation.


Set 1 contains derivatives  1 3

Set 2 contains derivatives  2

SUMMARY OF CORRELATED ERRORS AMONG DERIVATIVES

DERIVATIVE:          1
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
RMS errors correlated and uncorrelated with others in group:
    Correlated:   5.7    0.0    0.0    0.0    7.6    0.0    0.0   14.5    0.0
  Uncorrelated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 2:         0.24   0.18   0.39   0.10   0.36   0.31   0.17   0.55   0.23
DERIV 3:         0.40   0.30   0.64   0.20   0.68   0.41   0.31   0.85   0.49

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
Errors correlated and uncorrelated with others in group:
    Correlated:   6.4    0.0    4.2   15.0    0.0    0.0    0.0    0.0    8.9
  Uncorrelated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 2:         0.29   0.12   0.35   0.41   0.28   0.29   0.25   0.35   0.46
DERIV 3:         0.48   0.21   0.65   0.68   0.44   0.48   0.39   0.58   0.69

DERIVATIVE:          2
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
RMS errors correlated and uncorrelated with others in group:
    Correlated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  Uncorrelated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 1:         0.24   0.18   0.39   0.10   0.36   0.31   0.17   0.55   0.23
DERIV 3:         0.36   0.28   0.47   0.21   0.48   0.57   0.22   0.61   0.34
```

```
ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
Errors correlated and uncorrelated with others in group:
    Correlated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  Uncorrelated:   0.0    0.0    0.0    0.1    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 1:         0.29   0.12   0.35   0.41   0.28   0.29   0.25   0.35   0.46
DERIV 3:         0.39   0.18   0.43   0.55   0.36   0.38   0.32   0.49   0.62

DERIVATIVE:          3
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
RMS errors correlated and uncorrelated with others in group:
    Correlated:   5.7    0.0    0.0    0.0    7.6    0.0    0.0   14.5    0.0
  Uncorrelated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 1:         0.40   0.30   0.64   0.20   0.68   0.41   0.31   0.85   0.49
DERIV 2:         0.36   0.28   0.47   0.21   0.48   0.57   0.22   0.61   0.34

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
Errors correlated and uncorrelated with others in group:
    Correlated:   6.7    0.0    4.2   14.2    0.0    0.0    0.0    0.0   10.6
  Uncorrelated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 1:         0.48   0.21   0.65   0.68   0.44   0.48   0.39   0.58   0.69
DERIV 2:         0.39   0.18   0.43   0.55   0.36   0.38   0.32   0.49   0.62


PARAMETER SHIFTS FOR DERIV  1 : set 1 with 1  fe atoms, lambda 1

                 SCALE FACTOR     OVERALL B
CURRENT VALUES:        1.0000         0.0000



             SITE  ATOM      OCCUP     X       Y       Z        B


CURRENT VALUES:     1    Fe      0.5222  0.4391  0.1600  0.3825   26.2440

CURRENT VALUES:     2    Fe      0.2956  0.2289  0.4525  0.1668    9.1885


PARAMETER SHIFTS FOR DERIV  2 : Native from dataset # 2 (a MAD set) used as a deriv.

                 SCALE FACTOR     OVERALL B
CURRENT VALUES:        1.0000         0.0000
```

```
                SITE  ATOM      OCCUP     X       Y       Z        B

CURRENT VALUES:     1    Fe      0.0100  0.0000  0.0000  0.0000    0.0000

PARAMETER SHIFTS FOR DERIV  3 : Fe RM

                    SCALE FACTOR     OVERALL B
CURRENT VALUES:         1.0000         0.0000


                SITE  ATOM      OCCUP     X       Y       Z        B

CURRENT VALUES:     1    Fe      0.4912  0.3196  0.0000  0.2287   27.9862


*************************************************************
*************************************************************


*** Summary of solutions and their relationships
 to each other and to check solution ***


-------------------------------------------------------
solution #          1 with overall quality =   32.89179
Derivative           1 with           2 sites.
Overall scale =    1.000000     and  overall b of   0.0000000E+00
 0.4391191      0.1600000      0.3824790      0.5222168       26.24403
 0.2288599      0.4524674      0.1667664      0.2956249       9.188481
Derivative           3 with           1 sites.
Overall scale =    1.000000     and  overall b of   0.0000000E+00
 0.3196436      0.0000000E+00  0.2286830      0.4911909       27.98615
Best match of solution 1 -> solution 2:

-------- solution 1 --------    -------------solution 2 ------
site     x       y       z    site     x       y       z     DIST (A)
Derivative  1
 1    0.439    0.160    0.382    1    0.440    0.160    0.380    0.14
 2    0.229    0.452    0.167    2    0.230    0.450    0.165    0.14
Derivative  3
 1    0.320    0.000    0.229    1    0.320    0.030    0.230    0.84
Comparison of this solution with check solution:
Number of sites in this solution matching check=          3
... and number not matching =           0
by derivative, this is...
Deriv          nsame          ndifferent
         1          2          0
         2          0          0
```

3 1 0

All sites in this solution are contained in check soln

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
#
#  generate 2 mir datasets
#
solve <<EOD

!  GENERATE 2 MIR DATASETS:

! dataset 1
resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym

percent_error 4
coordinatefile coords.pdb
iranseed -199753
logfile generate.logfile
solvefile generate.prt

!
derivative 1
label set 1 with 1 hg atoms, derivative 1
cell_deriv 76 28 42 90 103 90
ATOMNAME hg
xyz  0.44 0.16 0.38
occ 1.0
bvalue 20.
derivative 2
label deriv 2 set 1, Iodine
cell_deriv 76 28 42 90 103 90
ATOMNAME I
xyz  0.23 0.45 0.165
occ 1.0
bvalue 20.
!
GENERATE_MIR                    ! generate the MAD dataset now.
!
```

```
EOD
mv der1.intensities der1_hg.intensities
mv der2.intensities der2_i.intensities
mv native.intensities native_1.intensities

solve <<EOD

! NOW DATASET 2
@solve.setup
cell 76 28 42 90 103 90
percent_error 4
coordinatefile coords.pdb
iranseed 532131
logfile generate_2.logfile
solvefile generate_2.prt

derivative 1
label set 2 with 1 Pt atoms, derivative 1
cell_deriv 76 28 42 90 103 90
ATOMNAME  pt
xyz 0.71241 0.315 0.2167
occ 1.0
bvalue 25
GENERATE_MIR                    ! generate the MAD dataset now.
!
EOD
mv der1.intensities der1_pt.intensities
mv native.intensities native_2.intensities

# end of generating 2 mir datasets
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
# solve.com  -- take 2 mir datasets that may or may not
#   be exactly isomorphous, combine them into one pseudo-mir dataset
# and solve it
#
solve<<EOD > solve.log
logfile solve.logfile

! solve.setup for test case
resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym


readformatted       ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged           ! premerged/ unmerged
read_intensities    ! read_intensities/read_amplitudes
fixscattfactors     ! fixscattfactors/refscattfactors

!
!---------first MIR dataset (hg, i atoms)---------
rawnativefile native_1.intensities
derivative 1
label set 1 with 1 hg atoms, derivative 1
rawderivfile der1_hg.intensities
ATOMNAME hg
derivative 2
```

```
label deriv 2 set 1, Iodine
rawderivfile der2_i.intensities
ATOMNAME I

!
scale_native
scale_mir
analyze_mir

!------------------------end of first dataset -------------

new_dataset                  ! tell solve we're about to start a new one

!----------------second MAD dataset (Pt atoms) ----------
rawnativefile native_2.intensities
derivative 1
label set 2 with 1 hg atoms, derivative 1
rawderivfile der1_pt.intensities
ATOMNAME pt
scale_native
scale_mir
analyze_mir

!  combine the datasets into one now...

combine

!  solve it...
solve
!--------all done----------

EOD
#
# Now run Resolve to do density modification and build a model
#
resolve << EOD > resolve.log
!solvent_content 0.40        !    solvent fraction
EOD
#
#  That's it! Now resolve.mtz has your updated phases
#  and resolve.pdb has your model
#
```

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*****************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*****************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 6)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: SOLVE 06-Apr-05

Space group number is: 5
Space group name from file name is: c2
Rescaling standard dataset to put it on approximate absolute scale. NRES =
200; <F**2> expected = 196000.0
; <F**2> observed in lowest resolution shell = 434796.6
... Scale factor = 0.4507855
Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

SCALE_MIR for dataset 1
Scale derivatives to previously-scaled native.

Default of "fp_or_fm" (use either F+ or

F- if available) will be used as this flag was not set

Analysis of this MIR dataset.

Fnative, sigma, and (Fbar,sigma, delano,sig) for 2
derivatives written to:
mir_fbar.scl

Fnative, sigma, and (F+,sigma,F-,sig) for 2 derivatives written to:
mir_fpfm.scl

** Completeness of native data (F > 2.000000 * sigma)

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

*** Analysis of this scaled deriv data set ***

** Completeness of Fbar data for each derivative: **

Derivative 1
set 1 with 1 hg atoms, derivative 1

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0

5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

Derivative 2
deriv 2 set 1, Iodine

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

** R-factors for F-bar data isomorphous differences **

isomorphous differences derivs 1 - native

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	238	318.119	0.262	0.998	97.76	33.49	2.92
2	4.500	316	282.789	0.244	0.992	79.73	28.13	2.83
3	4.200	120	271.873	0.239	0.994	73.24	26.94	2.72
4	3.975	120	242.412	0.271	1.001	75.33	24.10	3.13
5	3.750	155	237.813	0.263	0.998	73.20	23.66	3.09
6	3.600	112	211.470	0.260	1.003	65.30	20.61	3.17
7	3.450	143	173.661	0.319	1.002	64.80	17.56	3.69
8	3.300	171	181.597	0.260	1.002	57.22	18.26	3.13
9	3.150	190	173.271	0.261	1.002	53.18	16.94	3.14
10	3.000	254	160.220	0.280	0.999	52.78	16.00	3.30
Total:		1819	229.157	0.262	0.998	71.59	23.78	3.09

Recommended resolution cut-off = 3.00

isomorphous differences derivs 2 - native

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	239	319.147	0.206	0.996	73.61	33.36	2.21
2	4.500	317	283.738	0.176	0.996	54.95	28.23	1.95
3	4.200	120	271.873	0.174	1.000	49.26	26.92	1.83
4	3.975	120	242.412	0.186	1.001	50.14	24.03	2.09
5	3.750	155	237.813	0.191	1.003	48.32	23.73	2.04
6	3.600	112	211.470	0.205	0.997	49.37	20.72	2.38
7	3.450	143	173.661	0.211	1.001	42.17	17.46	2.42
8	3.300	172	181.495	0.200	0.999	41.53	18.23	2.28
9	3.150	190	173.271	0.201	0.995	38.06	16.95	2.25
10	3.000	254	160.220	0.207	1.000	37.10	16.13	2.30
Total:		1822	229.500	0.194	0.998	50.55	23.79	2.17

Recommended resolution cut-off = 3.00

** R-factors for anomalous differences **

anomalous differences deriv 1

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	232	298.842	0.080	1.000	0.00	33.81	0.00
2	4.500	310	279.582	0.085	0.999	16.55	28.86	0.57
3	4.200	119	272.658	0.082	1.001	10.80	28.47	0.38
4	3.975	116	228.191	0.073	1.002	0.00	24.63	0.00
5	3.750	152	236.769	0.078	1.000	5.51	25.00	0.22
6	3.600	109	201.679	0.076	1.002	0.00	21.20	0.00
7	3.450	139	178.782	0.084	1.000	8.48	18.74	0.45
8	3.300	165	169.962	0.077	1.000	0.00	18.45	0.00
9	3.150	185	166.515	0.085	1.000	8.75	17.35	0.50
10	3.000	248	154.622	0.077	1.000	0.00	16.23	0.00
Total:		1775	222.499	0.080	1.000	7.71	24.44	0.23

Recommended resolution cut-off = 3.98

anomalous differences deriv 2

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	231	306.066	0.079	1.005	0.00	36.56	0.00
2	4.500	308	278.099	0.074	1.003	0.00	31.41	0.00
3	4.200	116	260.479	0.064	1.000	0.00	29.35	0.00
4	3.975	118	231.729	0.073	1.000	0.00	27.04	0.00
5	3.750	151	230.631	0.074	0.999	0.00	26.36	0.00
6	3.600	111	210.349	0.085	0.998	0.00	23.58	0.00
7	3.450	142	180.231	0.078	1.000	0.00	19.92	0.00
8	3.300	168	175.206	0.076	1.002	0.00	20.03	0.00
9	3.150	186	168.052	0.072	1.002	0.00	18.82	0.00
10	3.000	250	157.443	0.084	1.003	0.00	18.07	0.00
Total:		1781	223.321	0.076	1.002	0.00	26.35	0.00

Recommended resolution cut-off = 3.00

Script file suitable for running SOLVE written to:
solve_mir.script

----------NEW DATASET BEGINS HERE----------

Rescaling standard dataset to put it on approximate absolute scale. NRES =
200; <F**2> expected = 196000.0
; <F**2> observed in lowest resolution shell = 434796.6
... Scale factor = 0.4507855
Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

SCALE_MIR for dataset 2
Scale derivatives to previously-scaled native.

Default of "fp_or_fm" (use either F+ or
F- if available) will be used as this flag was not set

Analysis of this MIR dataset.

Fnative, sigma, and (Fbar,sigma, delano,sig) for 1
derivatives written to:
mir_fbar.scl_2

Fnative, sigma, and (F+,sigma,F-,sig) for 1 derivatives written to:
mir_fpfm.scl

** Completeness of native data (F > 2.000000 * sigma)

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

*** Analysis of this scaled deriv data set ***

** Completeness of Fbar data for each derivative: **

Derivative 1
set 2 with 1 hg atoms, derivative 1

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	238	99.6
2	4.500	317	317	100.0
3	4.200	120	119	99.2

4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1820	99.9

** R-factors for F-bar data isomorphous differences **

isomorphous differences derivs 1 - native

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	238	316.542	0.263	0.993	94.68	33.05	2.86
2	4.500	317	283.738	0.224	0.998	73.15	28.32	2.58
3	4.200	119	269.863	0.196	0.997	60.05	26.86	2.24
4	3.975	120	242.412	0.236	0.999	66.07	24.24	2.73
5	3.750	155	237.813	0.229	0.999	63.89	23.66	2.70
6	3.600	112	211.470	0.252	1.003	61.37	20.73	2.96
7	3.450	143	173.661	0.265	1.007	53.15	17.57	3.02
8	3.300	172	181.495	0.273	1.008	58.35	18.11	3.22
9	3.150	190	173.271	0.250	1.007	50.26	16.76	3.00
10	3.000	254	160.220	0.229	1.010	42.01	16.02	2.62
Total:		1820	228.955	0.240	1.002	65.28	23.73	2.78

Recommended resolution cut-off = 3.00

** R-factors for anomalous differences **

anomalous differences deriv 1

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	230	298.218	0.080	0.999	1.66	33.45	0.05
2	4.500	307	275.956	0.082	1.001	12.69	28.26	0.45
3	4.200	117	265.947	0.082	1.001	11.68	27.70	0.42
4	3.975	116	236.494	0.073	1.002	0.00	24.92	0.00
5	3.750	151	229.215	0.079	1.001	7.52	24.33	0.31
6	3.600	109	209.158	0.075	1.002	2.11	21.43	0.10
7	3.450	139	177.057	0.082	1.001	6.88	18.12	0.38

```
  8    3.300   169     174.454     0.084   1.000   7.03  18.21   0.39
  9    3.150   188     162.691     0.088   1.001   9.78  16.92   0.58
 10    3.000   246     154.940     0.076   1.000   0.00  15.96   0.00

Total:         1772     221.254     0.080   1.001   7.47  24.02   0.27

Recommended resolution cut-off =    3.00


Script file suitable for running SOLVE written to:
solve_mir.script

-------------------------------------------------

Combining a total of           2 MIR and          0
 MAD datasets to form a composite dataset


----------NEW DATASET BEGINS HERE---------------

 ****   SOLVE:  Solutions to MIR or SIR datasets  ******

Derivatives considered:              4  (NSET)
Cross-vectors tested in HASSP:       5  (ICRMAX, DEFAULT=20)
HASSP solutions saved per deriv:    30  (NTOPHASSP, DEFAULT=30)
Fourier peaks saved per map:        30  (NTOPFOUR, DEFAULT=10)
Sites per derivative:               5  (NSOLSITE, DEFAULT=20)
Derivative solutions per seed:       5  (NTOPDERIV, DEFAULT=5)
Seeds per derivative tested:         2  (NSEEDTEST,DEFAULT=10)
Sorted seeds to use                  5  (NSEEDSOLVE, DEFAULT=5)
Number of final solutions saved:     5  (NTOPSOLVE, DEFAULT=5)




 Sites per derivative vary with derivative.
Derivative   Max sites
    1           5

    2           5

    3          -1

    4           5


Solutions obtained will be compared to input solution (ICHECKSOLVE)
Correlated phasing used (CORRELPHASE)
Standard difference fouriers will be calculated for derivative          1
Standard difference fouriers will be calculated for derivative          2
Standard difference fouriers will be calculated for derivative          3
Standard difference fouriers will be calculated for derivative          4
For derivative           4 the corresponding native data
will be read from columns          11 and          12
```

```
For derivative          4 the corresponding native dataset is "derivative"
        3

Datafile with          16 columns of data:
Title:mir_fbar.scl        (cols 1 to 10) and mir_fbar.scl_2
Data: Native F data
Data: Native sigma of F data
Data: Fbar              set 1 with 1 hg atoms, derivative 1
Data: Sig of Fbar        set 1 with 1 hg atoms, derivative 1
Data: Del Ano (F+ - F-)   set 1 with 1 hg atoms, derivative 1
Data: sig of Del Ano      set 1 with 1 hg atoms, derivative 1
Data: Fbar              deriv 2 set 1, Iodine
Data: Sig of Fbar        deriv 2 set 1, Iodine
Data: Del Ano (F+ - F-)   deriv 2 set 1, Iodine
Data: sig of Del Ano      deriv 2 set 1, Iodine
Data: Native F data
Data: Native sigma of F data
Data: Fbar              set 2 with 1 hg atoms, derivative 1
Data: Sig of Fbar        set 2 with 1 hg atoms, derivative 1
Data: Del Ano (F+ - F-)   set 2 with 1 hg atoms, derivative 1
Data: sig of Del Ano      set 2 with 1 hg atoms, derivative 1

Fnat,sigma taken from columns         1         2
Fder,sig,Delano,sig deriv          1 from cols:         3         4
        5         6
Fder,sig,Delano,sig deriv          2 from cols:         7         8
        9        10
Fder,sig,Delano,sig deriv          3 from cols:        11        12
        0         0
Fder,sig,Delano,sig deriv          4 from cols:        13        14
       15        16

Check solution to be compared to all solutions found:

Derivative          1:
 Site    X      Y      Z
   1   0.440   0.160   0.380

Derivative          2:
 Site    X      Y      Z
   1   0.230   0.450   0.165

Derivative          3:
 Site    X      Y      Z

Derivative          4:
 Site    X      Y      Z
   1   0.712   0.315   0.217

*********************************************************

ANALYZE_SOLVE:  analysis of top          1 solutions
```

****

Solution 1

********** Analysis of this solution **********

****** Analysis of non-randomness of native Fourier map ******

A. Maps with distinct solvent regions havea high standard deviation
of local r.m.s. electron density.
For this map the SD of this local r.m.s. is 0.3435877

B. Maps with distinct solvent regions also have a high correlation
of local r.m.s. electron density with
density at neighboring locations. Typical
values for poor maps in this
structure solution are 0.2706311 +/- 4.8051257E-02
For this map the correlation of r.m.s. density
in neighboring boxes is 0.3589191
The correlation coefficient is used here in scoring.

Skew of the map is: 0.7876190

****** Analysis of derivative solutions with the difference Patterson ******
and with cross-validation difference Fouriers

Derivative # 1

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.438	0.167	0.385	99.551

Evaluation of this test soln with 1 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.875	0.000	-0.771	19820.8	19820.8	2

Overall quality of this Patterson soln = 7007.73
Overall quality of the fit to patterson = 0.138107E-05
Avg normalized peak height = 4955.21

Cross-validation fouriers calculated with all heavy atoms in

all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.440	0.160	0.380	0.652	20.946	30.70

Derivative # 2

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.229	0.444	0.167	102.535

Evaluation of this test soln with 1 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.458	0.000	-0.333	21027.1	21027.1	2

Overall quality of this Patterson soln = 7434.20
Overall quality of the fit to patterson = 0.234782E-04
Avg normalized peak height = 5256.77

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.230	0.451	0.165	0.670	21.331	34.79

Derivative # 3

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --

Derivative # 4

List of sites analyzed for compatibility with difference Patterson

```
 PEAK        X         Y         Z     OPTIMIZED
                                      RELATIVE OCCUPANCY
    1     0.714     0.319     0.219     106.411

Evaluation of this test soln with    1 sites after optimizing
occupancy of each site

Cross-vectors for sites  1 and  1 (excluding origin; 1000 = 1 sigma):
 #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
  1   -1.427   0.000  -0.438   22646.4     22646.4         2

Overall quality of this Patterson soln =  8006.72
Overall quality of the fit to patterson = 0.290024E-04
Avg normalized peak height =  5661.61


Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

 Site    x       y       z       occ       B     -- PEAK  HEIGHT --

   1   0.712   0.314   0.217   0.646  23.616            31.07

Summary of scoring for this solution:
                        -- over many solutions--    -- this solution --
Criteria                     MEAN          SD         VALUE       Z-SCORE
Pattersons:                   5.29        0.500         12.0         13.5
Cross-validation Fourier:    0.000E+00    0.500         73.2         146.
NatFourier CCx100:            27.1         4.81         35.9         1.84
Mean figure of meritx100:    0.000E+00     5.00         91.7         18.3
Correction for Z-scores:                                 -68.9

Overall Z-score value:                                    111.


****** Overall analysis of phasing for solution           1************


HEAVY:  Refine heavy atom parameters

File title: SOLVE     06-Apr-05
CRYSTALLOGRAPHIC PARAMETERS

    A =  76.00     B =  28.00     C =  42.00     alpha =  90.00     beta = 103.00     gamma =  90.00

PHASES CALCULATED EVERY   5 DEGREES
RESIDUALS CALCULATED ON EXTRA ZEROTH CYCLE ONLY
SIGMAS FROM data FILE WILL BE USED
STATISTICS WILL BE PRINTED ON ZEROTH CYCLE, SHIFTS ON LAST
PHASING WILL BE DONE TAKING INTO ACCOUNT THE CORRELATIONS AMONG DERIVATIVES
THE GROUPS OF DERIVATIVES WITH CORRELATIONS WILL BE UPDATED
THE BETA VALUES FOR EACH DERIV  WILL BE SET TO 1.0
PHASE-AVERAGED RESIDUALS WILL BE USED FOR PHASING
```

```
TYPE OF REFINEMENT SELECTED:
UNPHASED ORIGIN-REMOVED  PATTERSON REFINEMENT ONLY

Bayesian Correlated Phasing will be used

RESOLUTION LIMITS IN ANGSTROMS:      3.000    20.000
MINIMUM RATIO OF FDER TO RMS LACK-OF-CLOSURE FOR  INCLUSION IN REFINEMENT OR PHASING=      0.000
MINIMUM NATIVE F:      0.000
MINIMUM FIGURE OF MERIT FOR PHASED REFINEMENT:      0.000
MINIMUM ALLOWED ISOTROPIC B:      0.000
PARAMETER SHIFTS GREATER THAN     0.0000 TIMES SIGMA  WILL BE SCALED BY     0.5000
MINIMUM RATIO OF FNAT/SIGMA OR FDER/SIGMA TO INCLUDE:      1.000

NUMBER OF REFINEMENT CYCLES IS      2
DERIVATIVES REFINED DURING THESE CYCLES ARE :  0  0
TYPE OF OUTPUT SELECTED IS: +10 COLUMNS OF HENDRICKSON-LATTMAN COEFFICIENTS
1 INPUT data FILE WITH  16 COLUMNS IS:
combine.scl_1_2
COLUMN  0 : mir_fbar.scl        (cols 1 to 10) and mir_fbar.scl_2      ,cols 1 to
COLUMN  1 : Native F data
COLUMN  2 : Native sigma of F data
COLUMN  3 : Fbar              set 1 with 1 hg atoms, derivative 1
COLUMN  4 : Sig of Fbar        set 1 with 1 hg atoms, derivative 1
COLUMN  5 : Del Ano (F+ - F-)   set 1 with 1 hg atoms, derivative 1
COLUMN  6 : sig of Del Ano      set 1 with 1 hg atoms, derivative 1
COLUMN  7 : Fbar              deriv 2 set 1, Iodine
COLUMN  8 : Sig of Fbar        deriv 2 set 1, Iodine
COLUMN  9 : Del Ano (F+ - F-)   deriv 2 set 1, Iodine
COLUMN 10 : sig of Del Ano      deriv 2 set 1, Iodine
COLUMN 11 : Native F data
COLUMN 12 : Native sigma of F data
COLUMN 13 : Fbar              set 2 with 1 hg atoms, derivative 1
COLUMN 14 : Sig of Fbar        set 2 with 1 hg atoms, derivative 1
COLUMN 15 : Del Ano (F+ - F-)   set 2 with 1 hg atoms, derivative 1
COLUMN 16 : sig of Del Ano      set 2 with 1 hg atoms, derivative 1


data COLUMNS FOR NATIVE F AND SIGMA:     1    2
data COLUMNS FOR BEST AND MOST PROB PHASES AND FIGURE OF MERIT:     0    0    0
OVERALL SCALE FACTOR FOR ALL data =     1.000
SCALE FACTOR FOR NATIVE SIGMAS =     1.000


DERIVATIVE INFORMATION FOR  4 COMPOUNDS


COMPOUND 1     set 1 with 1 hg atoms, derivative 1
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)    3    4    0    0
THIS DERIVATIVE WILL BE USED IN PHASING
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     0.932*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
```

```
COMPOUND 2     deriv 2 set 1, Iodine
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)    7    8    0    0
THIS DERIVATIVE WILL BE USED IN PHASING
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     0.968*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 3     Native from dataset # 2 (an MIR set) used as a deriv.
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)   11   12    0    0
THIS DERIVATIVE WILL BE USED IN PHASING
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 4     set 2 with 1 hg atoms, derivative 1
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)   13   14    0    0
THIS DERIVATIVE WILL BE USED IN PHASING
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     0.939*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
CARRYING OUT STANDARD REFINEMENT
Total of           2 cycles will be done
Derivs refined will be           0          0

SUMMARY OF RESULTS ON FINAL CYCLE:

NUMBER OF REFLECTIONS READ =    1822
NUMBER OF F .GT. FMIN      =    1822
NUMBER OF F IN RES. LIMITS =    1822
NUMBER OF F .GT. MIN <M>   =    1822
NUMBER OF F USED TO REFINE =       0



FIGURE OF MERIT <       0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1.0
# OF REFLECTIONS         4     6    11     8    29    36    38    82   201  1407

FIGURE OF MERIT WITH RESOLUTION
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
N:              1822    106    152    202    220    247    278    298    319
MEAN FIG MERIT:   0.92   0.92   0.90   0.92   0.90   0.93   0.91   0.92   0.92




COMPOUND  1          set 1 with 1 hg atoms, derivative 1
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
```

```
CENTRIC REFLNS:   263.    33.    36.    31.    31.    36.    34.    33.    29.
RMS HA F:        99.7  133.4  114.4  113.8   89.5   99.8   85.0   68.5   71.2
RMS RESIDUAL:    30.6   45.3   33.2   32.2   25.2   37.2   17.2   23.6   17.5
RMS(FH)/RMS(E):  3.26   2.95   3.45   3.54   3.55   2.69   4.93   2.90   4.08
CENTRIC R FACT:  0.31   0.33   0.37   0.19   0.41   0.32   0.25   0.32   0.32

ACENTRIC REFLN:  1559.    73.   116.   171.   189.   211.   244.   265.   290.
RMS DERIV FPH:   290.8  495.3  334.4  356.2  342.9  293.3  240.5  223.2  199.6
RMS SIGMA FPH:    14.6   25.4   16.7   17.6   17.2   14.7   12.0   11.2    9.9
RMS SIGMA FP:     19.2   32.3   21.8   23.3   22.9   19.5   15.9   14.9   13.2
RMS HA F:        91.9  130.0  113.2  113.0   97.8   90.4   85.5   76.7   69.0
RMS RESIDUAL:    28.3   61.2   33.0   33.4   28.8   24.5   26.2   20.5   18.3
RMS(FH)/RMS(E):  3.24   2.12   3.43   3.38   3.39   3.69   3.26   3.74   3.77

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:          14.9   22.3   14.4    0.0   27.9    0.0   16.4   11.7
RMS FPH :            495.3  334.4  356.2  342.9  293.3  240.5  223.2  199.6
RMS FH  :            130.0  113.2  113.0   97.8   90.4   85.5   76.7   69.0
RMS SIGMA:            41.1   27.5   29.2   28.6   24.4   19.9   18.7   16.5
```

```
COMPOUND  2         deriv 2 set 1, Iodine
DMIN:         TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09

CENTRIC REFLNS:   263.    33.    36.    31.    31.    36.    34.    33.    29.
RMS HA F:        69.8   93.3   83.8   76.9   70.3   61.1   60.9   49.6   47.2
RMS RESIDUAL:    24.0   37.7   32.4   25.2   24.6   19.5   15.2    9.2   12.6
RMS(FH)/RMS(E):  2.91   2.48   2.59   3.06   2.85   3.13   4.01   5.40   3.75
CENTRIC R FACT:  0.28   0.36   0.31   0.28   0.31   0.23   0.27   0.17   0.20

ACENTRIC REFLN:  1559.    73.   116.   171.   189.   211.   244.   265.   290.
RMS DERIV FPH:   280.3  471.2  319.6  342.2  333.7  282.7  233.4  215.2  194.1
RMS SIGMA FPH:    14.0   24.1   15.8   17.0   16.9   14.0   11.6   10.7    9.7
RMS SIGMA FP:     19.2   32.3   21.8   23.3   22.9   19.5   15.9   14.9   13.2
RMS HA F:        64.0   93.1   83.7   75.8   68.5   63.5   56.6   54.1   47.5
RMS RESIDUAL:    24.5   42.3   30.2   29.6   27.6   24.2   19.5   20.1   17.2
RMS(FH)/RMS(E):  2.61   2.20   2.77   2.56   2.49   2.63   2.90   2.69   2.75

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:           0.0   21.2    0.0    0.0    0.0    0.0    0.0    0.0
RMS FPH :            471.2  319.6  342.2  333.7  282.7  233.4  215.2  194.1
RMS FH  :            93.1   83.7   75.8   68.5   63.5   56.6   54.1   47.5
RMS SIGMA:            40.2   27.0   28.8   28.4   24.0   19.7   18.4   16.4
```

```
COMPOUND  3          Native from dataset # 2 (an MIR set) used as a deriv.
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09

CENTRIC REFLNS:   263.    33.    36.    31.    31.    36.    34.    33.    29.
RMS HA F:         0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS RESIDUAL:     0.0    0.1    0.0    0.0    0.1    0.0    0.0    0.1    0.1
RMS(FH)/RMS(E):   0.82   0.69   0.89   0.00   0.54   0.00   0.86   0.68   0.76
CENTRIC R FACT:  10.52   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00

ACENTRIC REFLN:  1559.    73.   116.   171.   189.   211.   244.   265.   290.
RMS DERIV FPH:   271.2  454.3  308.7  327.7  324.2  278.2  222.2  212.5  185.8
RMS SIGMA FPH:    19.2   32.3   21.8   23.3   22.9   19.5   15.9   14.9   13.2
RMS SIGMA FP:     19.2   32.3   21.8   23.3   22.9   19.5   15.9   14.9   13.2
RMS HA F:         0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS RESIDUAL:     0.1    0.1    0.1    0.1    0.1    0.1    0.0    0.0    0.0
RMS(FH)/RMS(E):   0.62   0.45   0.60   0.62   0.39   0.53   0.97   1.03   0.92

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:            0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS FPH :             454.3  308.7  327.7  324.2  278.2  222.2  212.5  185.8
RMS FH  :               0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS SIGMA:             45.6   30.9   33.0   32.3   27.6   22.4   21.1   18.7



COMPOUND  4          set 2 with 1 hg atoms, derivative 1
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09

CENTRIC REFLNS:   263.    33.    36.    31.    31.    36.    34.    33.    29.
RMS HA F:         93.2  124.7  111.2  103.8   90.5   86.8   72.1   75.6   59.8
RMS RESIDUAL:     30.6   52.5   36.6   35.8   22.6   30.7   12.4   16.1   10.9
RMS(FH)/RMS(E):   3.05   2.37   3.04   2.90   4.01   2.83   5.80   4.68   5.46
CENTRIC R FACT:   0.29   0.36   0.31   0.26   0.31   0.28   0.25   0.19   0.34

ACENTRIC REFLN:  1557.    72.   116.   171.   188.   211.   244.   265.   290.
RMS DERIV FPH:   287.6  480.7  328.9  351.3  337.0  297.7  242.4  217.9  199.2
RMS SIGMA FPH:    14.4   24.0   16.4   17.6   16.8   14.9   12.1   10.9    9.9
RMS SIGMA FP:     19.1   31.4   21.8   23.3   22.8   19.5   15.9   14.9   13.2
RMS HA F:         85.0  122.1  111.8  101.0   93.2   83.9   76.3   69.8   62.9
RMS RESIDUAL:     28.1   58.7   31.5   35.3   30.7   25.1   24.7   19.8   17.8
RMS(FH)/RMS(E):   3.02   2.08   3.55   2.86   3.04   3.35   3.09   3.53   3.53

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:           31.5   27.5   18.1    0.0   18.5    0.0    0.0    0.0
RMS FPH :             480.7  328.9  351.3  337.0  297.7  242.4  217.9  199.2
```

```
RMS FH  :            122.1  111.8  101.0   93.2   83.9   76.3   69.8   62.9
RMS SIGMA:            39.5   27.3   29.2   28.4   24.6   19.9   18.5   16.5

Analysis of correlated modeling and non-isomorphism errors
obtained using phased residuals.
The derivatives were grouped into 3 sets  where the members of a set
had some mutual correlation.


Set 1 contains derivatives  1 4

Set 2 contains derivatives  2

Set 3 contains derivatives  3

SUMMARY OF CORRELATED ERRORS AMONG DERIVATIVES

DERIVATIVE:          1
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
RMS errors correlated and uncorrelated with others in group:
    Correlated:   16.4   25.2   25.7   19.8    0.0   15.8    0.0   11.9    2.9
  Uncorrelated:   10.1    0.0    0.0    0.0    0.0   22.9    0.0   11.2   11.4

Correlation of errors with other derivs:
DERIV 2:         0.48   0.53   0.70   0.35   0.50   0.38   0.45   0.19   0.40
DERIV 3:         0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
DERIV 4:         0.70   0.72   0.86   0.80   0.63   0.56   0.39   0.81   0.55

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
Errors correlated and uncorrelated with others in group:
    Correlated:   17.9   53.4   22.0   20.5   12.7    4.8   19.2    7.1    8.8
  Uncorrelated:   10.7   34.5   10.9    6.9    0.0    0.0   13.5    8.1    5.0

Correlation of errors with other derivs:
DERIV 2:         0.34   0.49   0.30   0.27   0.32   0.32   0.39   0.36   0.25
DERIV 3:         0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
DERIV 4:         0.68   0.69   0.71   0.65   0.70   0.66   0.69   0.63   0.68

DERIVATIVE:          2
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
RMS errors correlated and uncorrelated with others in group:
    Correlated:    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  Uncorrelated:    7.9    0.0   21.2    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 1:         0.48   0.53   0.70   0.35   0.50   0.38   0.45   0.19   0.40
DERIV 3:         0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
DERIV 4:         0.44   0.43   0.69   0.25   0.54   0.32   0.26   0.23   0.55

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.09
```

```
Errors correlated and uncorrelated with others in group:
    Correlated:   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  Uncorrelated:   8.6  17.1  18.2   8.0   0.0   0.0   0.0  11.0   6.6

Correlation of errors with other derivs:
DERIV 1:        0.34  0.49  0.30  0.27  0.32  0.32  0.39  0.36  0.25
DERIV 3:        0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
DERIV 4:        0.32  0.40  0.26  0.23  0.35  0.34  0.36  0.33  0.26

DERIVATIVE:         3
CENTRIC REFLECTIONS:
DMIN:         ALL    9.91  6.56  5.22  4.46  3.96  3.60  3.32  3.09
RMS errors correlated and uncorrelated with others in group:
    Correlated:   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  Uncorrelated:   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0

Correlation of errors with other derivs:
DERIV 1:        0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
DERIV 2:        0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
DERIV 4:        0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00

ACENTRIC REFLECTIONS:
DMIN:         ALL    9.91  6.56  5.22  4.46  3.96  3.60  3.32  3.09
Errors correlated and uncorrelated with others in group:
    Correlated:   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
  Uncorrelated:   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0

Correlation of errors with other derivs:
DERIV 1:        0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
DERIV 2:        0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
DERIV 4:        0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00

DERIVATIVE:         4
CENTRIC REFLECTIONS:
DMIN:         ALL    9.91  6.56  5.22  4.46  3.96  3.60  3.32  3.09
RMS errors correlated and uncorrelated with others in group:
    Correlated:  16.4  25.2  25.7  19.8   0.0  15.8   0.0  11.9   2.9
  Uncorrelated:   8.4  18.9   9.8   0.0   0.0   9.6   0.0   0.0   0.0

Correlation of errors with other derivs:
DERIV 1:        0.70  0.72  0.86  0.80  0.63  0.56  0.39  0.81  0.55
DERIV 2:        0.44  0.43  0.69  0.25  0.54  0.32  0.26  0.23  0.55
DERIV 3:        0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00

ACENTRIC REFLECTIONS:
DMIN:         ALL    9.91  6.56  5.22  4.46  3.96  3.60  3.32  3.09
Errors correlated and uncorrelated with others in group:
    Correlated:  18.0  54.5  22.0  20.5  12.9   4.8  19.2   7.1   8.8
  Uncorrelated:   9.1  27.0   0.0  17.6   7.5   0.0   5.2   5.2   0.0

Correlation of errors with other derivs:
DERIV 1:        0.68  0.69  0.71  0.65  0.70  0.66  0.69  0.63  0.68
DERIV 2:        0.32  0.40  0.26  0.23  0.35  0.34  0.36  0.33  0.26
DERIV 3:        0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00
```

```
PARAMETER SHIFTS FOR DERIV  1 : set 1 with 1 hg atoms, derivative 1

                 SCALE FACTOR     OVERALL B
CURRENT VALUES:        0.9317         0.0000



             SITE  ATOM      OCCUP     X      Y      Z        B


CURRENT VALUES:     1    Hg      0.6521  0.4399  0.1600  0.3804   20.9460


PARAMETER SHIFTS FOR DERIV  2 : deriv 2 set 1, Iodine

                 SCALE FACTOR     OVERALL B
CURRENT VALUES:        0.9675         0.0000



             SITE  ATOM      OCCUP     X      Y      Z        B


CURRENT VALUES:     1    I       0.6699  0.2303  0.4511  0.1652   21.3307


PARAMETER SHIFTS FOR DERIV  3 : Native from dataset # 2 (an MIR set) used as a deriv.

                 SCALE FACTOR     OVERALL B
CURRENT VALUES:        1.0000         0.0000



             SITE  ATOM      OCCUP     X      Y      Z        B


CURRENT VALUES:     1    NATV     0.0100  0.0000  0.0000  0.0000    0.0000


PARAMETER SHIFTS FOR DERIV  4 : set 2 with 1 hg atoms, derivative 1

                 SCALE FACTOR     OVERALL B
CURRENT VALUES:        0.9388         0.0000



             SITE  ATOM      OCCUP     X      Y      Z        B


CURRENT VALUES:     1    Pt      0.6459  0.7123  0.3140  0.2174   23.6165
```

```
*************************************************************
*************************************************************


*** Summary of solutions and their relationships
 to each other and to check solution ***


-------------------------------------------------------
solution #          1 with overall quality =   111.1002
Derivative          1 with          1 sites.
Overall scale =   0.9316655     and  overall b of   0.0000000E+00
 0.4398852      0.1600000      0.3804173      0.6521176       20.94600
Derivative          2 with          1 sites.
Overall scale =   0.9675072     and  overall b of   0.0000000E+00
 0.2302904      0.4511112      0.1651833      0.6698623       21.33070
Derivative          4 with          1 sites.
Overall scale =   0.9388404     and  overall b of   0.0000000E+00
 0.7123117      0.3139645      0.2173692      0.6458988       23.61647
Best match of solution 1 -> solution 2:

-------- solution 1 --------    -------------solution 2 ------
site     x       y       z    site     x       y       z     DIST (A)
Derivative  1
 1    0.440    0.160    0.380    1    0.440    0.160    0.380    0.02
Derivative  2
 1    0.230    0.451    0.165    1    0.230    0.450    0.165    0.04
Derivative  4
 1    0.712    0.314    0.217    1    0.712    0.315    0.217    0.04
Comparison of this solution with check solution:
Number of sites in this solution matching check=          3
... and number not matching =          0
by derivative, this is...
Deriv          nsame          ndifferent
         1          1          0
         2          1          0
         3          0          0
         4          1          0
All sites in this solution are contained in check soln
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
#
#   use all system resources:
#
unlimit
#
solve  <<EOD

! generate se + fe mad dataset
resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym

logfile generate.logfile
solvefile generate.prt
coordinatefile coords.pdb
percent_error  1.0
iranseed 11933
! need to define Fe scattering factors explicitly as it is the 2nd mad atom
! new atom types for Fe (2nd mad atom) at each wavelength:
newatomtype fel1
aval 11.76950       7.357300       3.522200  2.304500
bval 4.761100      0.3072000       15.35350     76.88050
cval 1.036900
fprimv 0.3
fprprv 1.5
newatomtype fel2
aval 11.76950       7.357300       3.522200  2.304500
bval 4.761100      0.3072000       15.35350     76.88050
cval 1.036900
fprimv 0.3
fprprv 1.5
newatomtype fel3
aval 11.76950       7.357300       3.522200  2.304500
```

```
bval 4.761100      0.3072000       15.35350     76.88050
cval 1.036900
fprimv 0.3
fprprv 1.5
newatomtype fel4
aval 11.76950       7.357300       3.522200  2.304500
bval 4.761100      0.3072000       15.35350     76.88050
cval 1.036900
fprimv -9.0
fprprv 2.5
newatomtype fel5
aval 11.76950       7.357300       3.522200  2.304500
bval 4.761100      0.3072000       15.35350     76.88050
cval 1.036900
fprimv -5.0
fprprv 4.5


! dataset treated as se MAD data with Fe as extra atom
mad_atom se                    ! define the scattering factors...
!
lambda 1
label set 1 with se + Fe , lambda 1
wavelength  0.9000            ! wavelength value
fprimv_mad  -1.6              ! f' value for Se (fe defined in fel1)
fprprv_mad  3.4
ATOMNAME Se
! generate automatically sets up se at all the other wavelengths too
xyz  0.44 0.16 0.38
occ 1.0
bvalue 20
atomname fel1
! generate automatically sets up fel2, fel3 etc at other wavelengths
xyz 0.15 0.33 0.40
occ 1.0
bvalue 20

lambda 2
wavelength .9794
fprimv_mad  -8.5
fprprv_mad 4.8

lambda 3
```

```
wavelength 0.9797
fprimv_mad  -9.85
fprprv_mad   2.86

lambda 4
wavelength 1.74            ! wavelength value
fprimv_mad -0.6          ! f prime value for Se (fe defined in fel4)
fprprv_mad  1.4

lambda 5
wavelength 1.736            ! wavelength value
fprimv_mad -0.6          ! f prime value for Se (fe defined in fel4)
fprprv_mad  1.4


generate_mad

EOD
```

```
#!/bin/csh
#
#   solve.com file to run SOLVE/RESOLVE version 2.01
#
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#  set CCP4 and SOLVETMPDIR  and SYMOP variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
#
#   use all system resources:
#
unlimit
#
solve<<EOD > solve.log

! Solve 1 mad dataset with 2 anomalously-scattering atoms

logfile solve.logfile

resolution 20 3.0
cell 76 28 42 90 103 90
symfile /usr/local/lib/solve/c2.sym


readformatted        ! readformatted/readdenzo/readtrek/readccp4_unmerged
premerged            ! premerged/ unmerged
read_intensities     ! read_intensities/read_amplitudes
fixscattfactors      ! fixscattfactors/refscattfactors

title  5-wavelength 2-ano scatterer MAD dataset   !  a title for this dataset

! Read in the data twice: once treated as Se and once as Fe.

mad_atom se                  ! the anomalously scattering atom is selenium
lambda 1                     ! info on wavelength #1 follows
label Wavelength #  1        ! a label for this wavelength
rawmadfile lam1.intensities  ! datafile for lambda 1 containing raw Intensities

wavelength 0.9000            ! wavelength value
```

```
fprimv_mad  -1.6            ! f' value at this wavelength (for Se)
fprprv_mad  3.4             ! f doubleprime value at this wavelength

atomname se                 !  the atom name

lambda 2
rawmadfile lam2.intensities
wavelength 0.9794
fprimv_mad  -8.5
fprprv_mad  4.8

lambda 3
rawmadfile lam3.intensities
wavelength 0.9797
fprimv_mad  -9.85
fprprv_mad  2.86

nres 100                [approx # of residues of protein in asymmetric unit]
nanomalous 1            [approx # of anomalously scattering Se atoms in a.u.]
SCALE_MAD               ! read in and localscale the data
ANALYZE_MAD             ! run MADMRG and MADBST and analyze all the Pattersons

! now do it all over for Fe instead of Se:

NEW_DATASET             ! tell solve we're starting a new dataset

mad_atom Fe             ! the anomalously scattering atom is Fe

lambda 1                ! info on wavelength #1 follows
label Wavelength #  1   ! a label for this wavelength
rawmadfile lam4.intensities  ! datafile for lambda 1 containing raw Intensities
atomname fe

wavelength 1.74         ! wavelength value
fprimv_mad  -9          ! f' value for Fe at this wavelength
fprprv_mad  2.5         ! f doubleprime at this wavelength

lambda 2
rawmadfile lam5.intensities
wavelength 1.73647
fprimv_mad  -5
fprprv_mad  4.5
```

```
lambda 3
rawmadfile lam3.intensities
wavelength 0.9797
fprimv_mad 0.3
fprprv_mad  1.5

nres 100                [approx # of residues of protein in asymmetric unit]
nanomalous 1            [approx # of anomalously scattering Fe atoms in a.u.]
SCALE_MAD               ! read in and localscale the data
ANALYZE_MAD             ! run MADMRG and MADBST and analyze all the Pattersons

! now combine the datasets into one

COMBINE_ALL_DATA

! now solve it as a whole

SOLVE                   ! Solve the structure

!--------all done----------

EOD
#
# Now run Resolve to do density modification and build a model
#
resolve << EOD > resolve.log
!solvent_content 0.4       !    solvent fraction
EOD
#
#  That's it! Now resolve.mtz has your updated phases
#  and resolve.pdb has your model
#
```

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*****************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*****************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 6)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: 5-wavelength 2-ano scatterer MAD dataset ! a title for th

Space group number is: 5
Space group name from file name is: c2
Rescaling standard dataset to put it on approximate absolute scale. NRES =
100; <F**2> expected = 98000.00
; <F**2> observed in lowest resolution shell = 441085.0
... Scale factor = 0.2221794

*** Analysis of this scaled MAD data set ***

Fbar,sigma,Delano,sigma for 3 wavelengths written to:
mad_fbar.scl

F+,sigma,F-,sigma for 3 wavelengths written to:
mad_fpfm.scl

** Completeness of Fbar data at each wavelength: **

Completeness of dataset 1 (F > 2.000000 * sigma)
Wavelength # 1 ! a label for this wavelength

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	239	100.0
2	4.500	317	316	99.7
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1821	99.9

Completeness of dataset 2 (F > 2.000000 * sigma)
set 2

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	238	99.6
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	154	99.4
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	253	99.6
total		1822	1819	99.8

Completeness of dataset 3 (F > 2.000000 * sigma)
set 3

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	238	99.6
2	4.500	317	316	99.7
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	154	99.4
6	3.600	112	112	100.0
7	3.450	143	143	100.0

8	3.300	172	172	100.0
9	3.150	190	189	99.5
10	3.000	254	254	100.0
total		1822	1818	99.8

** R-factors for F-bar data dispersive differences **

Dispersive differences lambda 2 - lambda 1
(Delta f-prime = 6.900000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	235	218.217	0.038	0.999	2.97	9.85	0.30
2	4.500	315	204.705	0.037	1.001	4.90	8.42	0.58
3	4.200	120	194.749	0.038	1.002	4.74	8.01	0.59
4	3.975	120	172.429	0.041	1.001	5.52	7.15	0.77
5	3.750	153	169.131	0.037	1.001	3.84	6.91	0.56
6	3.600	111	149.405	0.035	0.999	2.29	6.20	0.37
7	3.450	143	129.313	0.041	1.000	3.87	5.35	0.72
8	3.300	170	127.987	0.037	1.000	2.87	5.43	0.53
9	3.150	189	121.589	0.038	1.000	2.51	5.06	0.50
10	3.000	253	115.801	0.042	1.001	3.74	4.76	0.79
Total:		1809	162.970	0.038	1.000	3.87	7.06	0.57

Recommended resolution cut-off = 3.00

Dispersive differences lambda 3 - lambda 1
(Delta f-prime = 8.250000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	235	220.145	0.041	0.999	4.42	9.93	0.45
2	4.500	311	201.222	0.039	1.002	5.91	8.18	0.72
3	4.200	119	192.819	0.037	1.002	4.15	7.86	0.53
4	3.975	120	172.429	0.047	1.001	7.32	7.24	1.01
5	3.750	151	167.018	0.035	1.000	2.91	6.88	0.42
6	3.600	110	147.993	0.042	0.999	4.44	6.20	0.72
7	3.450	142	128.263	0.048	1.000	5.46	5.22	1.05
8	3.300	169	127.759	0.043	0.998	4.38	5.39	0.81
9	3.150	189	121.449	0.045	1.000	4.45	5.02	0.89
10	3.000	254	116.982	0.050	0.999	5.25	4.93	1.07
Total:		1800	162.179	0.042	1.000	5.03	7.02	0.77

Recommended resolution cut-off = 3.00

Dispersive differences lambda 3 - lambda 2
(Delta f-prime = 1.350000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	233	215.942	0.026	1.000	0.00	8.86	0.00
2	4.500	312	202.298	0.027	1.000	0.00	7.64	0.00
3	4.200	119	191.608	0.026	1.000	0.00	7.30	0.00
4	3.975	119	172.941	0.028	1.001	0.00	6.55	0.00
5	3.750	153	168.137	0.028	1.000	0.00	6.29	0.00
6	3.600	110	149.875	0.026	1.000	0.00	5.67	0.00
7	3.450	140	127.560	0.028	1.000	0.00	4.87	0.00
8	3.300	169	127.881	0.029	1.000	0.81	5.01	0.16
9	3.150	185	120.177	0.027	1.000	0.00	4.53	0.00
10	3.000	250	114.023	0.028	0.999	0.00	4.25	0.00
Total:		1790	161.583	0.027	1.000	0.00	6.40	0.02

Recommended resolution cut-off = 3.00

Anomalous differences lambda 1 (f" = 3.400000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	230	212.730	0.045	1.000	0.00	12.88	0.00
2	4.500	309	200.601	0.047	1.000	5.44	11.35	0.48
3	4.200	119	193.694	0.049	0.999	5.41	11.00	0.49
4	3.975	117	167.885	0.045	1.001	3.06	9.73	0.31
5	3.750	153	169.878	0.049	1.000	4.61	9.72	0.47
6	3.600	108	145.102	0.044	1.001	2.66	8.24	0.32
7	3.450	140	125.881	0.054	1.000	5.32	7.07	0.75
8	3.300	168	127.966	0.051	1.000	3.89	7.45	0.52
9	3.150	187	120.508	0.054	1.001	4.84	6.89	0.70
10	3.000	243	110.784	0.045	1.000	0.63	6.23	0.10
Total:		1774	160.003	0.048	1.000	3.93	9.49	0.39

Recommended resolution cut-off = 3.00

Anomalous differences lambda 2 (f" = 4.800000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	228	207.783	0.051	1.002	7.28	12.31	0.59
2	4.500	308	200.322	0.048	1.001	5.25	11.36	0.46
3	4.200	114	185.291	0.045	0.999	1.72	10.61	0.16
4	3.975	118	172.446	0.045	0.999	0.00	9.98	0.00
5	3.750	151	165.460	0.053	0.999	6.49	9.36	0.69
6	3.600	110	149.615	0.056	0.999	6.49	8.52	0.76
7	3.450	136	124.869	0.052	0.999	4.26	7.09	0.60
8	3.300	165	125.230	0.056	1.000	4.71	7.22	0.65
9	3.150	187	120.391	0.052	1.000	3.87	6.89	0.56
10	3.000	252	114.486	0.057	1.000	5.47	6.48	0.84
Total:		1769	158.845	0.051	1.000	5.22	9.36	0.56

Recommended resolution cut-off = 3.00

Anomalous differences lambda 3 (f" = 2.860000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	230	211.880	0.040	1.001	0.00	12.59	0.00
2	4.500	306	198.776	0.041	1.000	0.00	11.32	0.00
3	4.200	116	188.011	0.044	1.000	0.00	10.72	0.00
4	3.975	115	164.981	0.047	1.000	3.75	9.42	0.40
5	3.750	147	163.704	0.043	1.001	0.00	9.26	0.00
6	3.600	109	146.694	0.049	1.000	3.97	8.36	0.48
7	3.450	137	121.796	0.048	1.001	3.34	6.88	0.49
8	3.300	164	123.022	0.047	1.000	3.22	7.06	0.46
9	3.150	184	119.630	0.047	0.999	3.29	6.82	0.48
10	3.000	246	113.607	0.048	1.000	3.17	6.50	0.49
Total:		1754	158.047	0.044	1.000	0.00	9.34	0.26

Recommended resolution cut-off = 3.00

ANALYZE_MAD: Run MADMRG and MADBST on MAD data to get ready for SOLVE

Correlation of anomalous differences at different wavelengths.
(You should probably cut your data off at the resolution where
this drops below about 0.3. A good dataset has correlation
between peak and remote of at least 0.7 overall. Data with

correlations below about 0.5 probably are not contributing much.)

DMIN	CORRELATION FOR WAVELENGTH PAIRS 1 VS 2	1 VS 3	2 VS 3
6.00	0.26	0.01	0.11
4.50	0.12	0.18	0.28
4.20	0.32	0.14	0.13
3.98	0.30	0.05	0.14
3.75	0.38	0.20	0.28
3.60	0.45	0.30	0.34
3.45	0.29	0.24	0.31
3.30	0.44	0.22	0.23
3.15	0.36	0.15	0.31
3.00	0.32	0.24	0.32
ALL	0.28	0.15	0.23

Final refined values of fprime and fdoubleprime

Form factors at lambda = 0.9000
f-prime = -1.60
f" = 3.40

Form factors at lambda = 0.9794
f-prime = -8.50
f" = 4.80

Form factors at lambda = 0.9797
f-prime = -9.85
f" = 2.86

Fa Patterson from MADBST to be written to:
patterson.patt

Script file suitable for running SOLVE written to:
solve_mad.script

Datafile for SOLVE with MADMRG-compressed dataset
("Fnat",sig,"Fder",sig,"Delano",sig,iso diffs, ano diffs,
<Fhcos>, <Fhsin> from MADBST) is:
solve.data

----------NEW DATASET BEGINS HERE----------

```
Rescaling standard dataset to put it on approximate absolute scale.  NRES =
       100;  <F**2> expected =    98000.00
;  <F**2> observed in lowest resolution shell =    431826.0
  ... Scale factor =   0.2269433

---------------------------------------------------

 *** Analysis of this scaled MAD data set ***

 Fbar,sigma,Delano,sigma for           3 wavelengths written to:
mad_fbar.scl_2


 F+,sigma,F-,sigma for           3 wavelengths written to:
mad_fpfm.scl


 ** Completeness of Fbar data at each wavelength: **



Completeness of dataset           1  ( F >    2.000000     * sigma)
Wavelength #  1         ! a label for this wavelength

Reflections observed:

                 Possible  Found     % complete
shell     dmin

   1     6.000       239       239     100.0
   2     4.500       317       317     100.0
   3     4.200       120       120     100.0
   4     3.975       120       120     100.0
   5     3.750       155       155     100.0
   6     3.600       112       112     100.0
   7     3.450       143       143     100.0
   8     3.300       172       172     100.0
   9     3.150       190       189      99.5
  10     3.000       254       254     100.0

total              1822      1821      99.9


Completeness of dataset           2  ( F >    2.000000     * sigma)
set 2

Reflections observed:

                 Possible  Found     % complete
shell     dmin

   1     6.000       239       239     100.0
   2     4.500       317       317     100.0
   3     4.200       120       120     100.0
```

4	3.975	120	120	100.0
5	3.750	155	155	100.0
6	3.600	112	112	100.0
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1822	100.0

Completeness of dataset 3 (F > 2.000000 * sigma)
set 3

Reflections observed:

shell	dmin	Possible	Found	% complete
1	6.000	239	237	99.2
2	4.500	317	317	100.0
3	4.200	120	120	100.0
4	3.975	120	120	100.0
5	3.750	155	154	99.4
6	3.600	112	111	99.1
7	3.450	143	143	100.0
8	3.300	172	172	100.0
9	3.150	190	190	100.0
10	3.000	254	254	100.0
total		1822	1818	99.8

** R-factors for F-bar data dispersive differences **

Dispersive differences lambda 2 - lambda 1
(Delta f-prime = 4.000000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	234	220.477	0.031	1.000	0.00	9.60	0.00
2	4.500	314	204.955	0.031	1.000	1.05	8.26	0.13
3	4.200	119	194.771	0.032	1.000	2.08	7.70	0.27
4	3.975	119	171.543	0.031	1.000	0.00	7.11	0.00
5	3.750	152	170.614	0.031	0.999	0.00	6.91	0.00
6	3.600	111	150.925	0.026	1.000	0.00	5.94	0.00
7	3.450	142	128.418	0.033	1.000	1.50	5.09	0.29
8	3.300	170	130.293	0.035	1.000	2.33	5.40	0.43
9	3.150	186	122.859	0.030	0.999	0.00	4.89	0.00
10	3.000	249	115.635	0.032	1.000	0.00	4.72	0.00

Total: 1796 163.839 0.031 1.000 0.00 6.91 0.10

Recommended resolution cut-off = 3.00

Dispersive differences lambda 3 - lambda 1
(Delta f-prime = 9.300000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	236	219.981	0.059	1.001	12.72	9.57	1.33
2	4.500	315	206.125	0.050	1.003	9.92	8.44	1.17
3	4.200	120	196.780	0.047	1.002	8.56	8.09	1.06
4	3.975	120	172.916	0.058	1.000	9.97	7.38	1.35
5	3.750	153	173.204	0.053	0.999	8.83	7.17	1.23
6	3.600	111	150.631	0.059	0.999	9.10	6.12	1.49
7	3.450	142	128.418	0.059	0.999	7.86	5.22	1.50
8	3.300	171	130.508	0.067	0.999	9.17	5.45	1.68
9	3.150	189	123.416	0.058	0.999	7.69	5.06	1.52
10	3.000	253	117.220	0.063	1.000	7.91	4.86	1.63

Total: 1810 164.621 0.056 1.000 9.44 7.06 1.40

Recommended resolution cut-off = 3.00

Dispersive differences lambda 3 - lambda 2
(Delta f-prime = 5.300000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	234	215.887	0.055	1.000	11.13	9.44	1.18
2	4.500	315	205.942	0.046	1.002	8.80	8.50	1.04
3	4.200	120	196.269	0.046	1.001	7.76	8.03	0.97
4	3.975	120	173.187	0.047	1.000	6.60	7.57	0.87
5	3.750	153	172.408	0.045	1.000	6.52	7.09	0.92
6	3.600	110	148.865	0.052	0.999	7.72	6.00	1.29
7	3.450	143	129.807	0.055	1.000	6.87	5.23	1.31
8	3.300	171	130.712	0.054	1.000	6.54	5.55	1.18
9	3.150	188	121.800	0.050	1.000	5.63	5.03	1.12
10	3.000	252	117.205	0.053	1.000	5.95	4.84	1.23

Total: 1806 163.802 0.050 1.001 7.73 7.05 1.12

Recommended resolution cut-off = 3.00

Anomalous differences lambda 1 (f" = 2.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	232	220.771	0.038	0.999	0.00	13.44	0.00
2	4.500	306	199.647	0.043	1.000	0.00	11.31	0.00
3	4.200	117	193.773	0.041	0.999	0.00	11.01	0.00
4	3.975	117	167.285	0.038	1.000	0.00	9.64	0.00
5	3.750	152	168.847	0.046	1.001	3.65	9.66	0.38
6	3.600	110	149.197	0.047	0.999	2.37	8.43	0.28
7	3.450	140	126.492	0.043	1.000	0.00	7.21	0.00
8	3.300	168	128.786	0.047	1.000	2.10	7.50	0.28
9	3.150	183	119.213	0.042	1.000	0.00	6.76	0.00
10	3.000	247	113.788	0.045	0.999	1.46	6.46	0.23
Total:		1772	161.342	0.042	1.000	0.00	9.61	0.11

Recommended resolution cut-off = 3.00

Anomalous differences lambda 2 (f" = 4.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	227	209.192	0.048	1.000	4.27	12.50	0.34
2	4.500	308	200.604	0.047	0.999	5.27	11.34	0.46
3	4.200	117	193.828	0.046	0.999	0.00	11.06	0.00
4	3.975	114	163.635	0.045	1.000	0.00	9.43	0.00
5	3.750	150	168.890	0.051	1.001	5.46	9.65	0.57
6	3.600	112	151.295	0.059	1.000	7.47	8.60	0.87
7	3.450	142	129.488	0.059	1.000	5.65	7.36	0.77
8	3.300	167	126.610	0.058	1.001	5.84	7.33	0.80
9	3.150	184	119.911	0.054	1.001	5.29	6.84	0.77
10	3.000	247	113.895	0.058	1.000	5.33	6.48	0.82
Total:		1768	159.847	0.051	1.000	5.03	9.44	0.56

Recommended resolution cut-off = 3.00

Anomalous differences lambda 3 (f" = 1.500000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	6.000	229	208.844	0.042	1.003	0.00	12.25	0.00
2	4.500	307	201.327	0.041	1.001	0.00	11.49	0.00
3	4.200	115	188.831	0.044	1.001	0.00	10.80	0.00
4	3.975	115	165.530	0.049	0.999	4.91	9.45	0.52
5	3.750	148	168.501	0.043	1.000	0.00	9.59	0.00
6	3.600	109	147.208	0.050	0.999	4.42	8.40	0.53
7	3.450	139	125.412	0.049	1.000	3.96	7.14	0.55
8	3.300	168	127.829	0.049	1.000	4.32	7.38	0.59
9	3.150	183	120.334	0.046	0.999	2.60	6.88	0.38
10	3.000	248	114.963	0.048	0.999	3.02	6.58	0.46
Total:		1761	159.459	0.045	1.000	1.09	9.40	0.27

Recommended resolution cut-off = 3.00

ANALYZE_MAD: Run MADMRG and MADBST on MAD data to get ready for SOLVE

Final refined values of fprime and fdoubleprime

Form factors at lambda = 1.7400
f-prime = -9.00
f" = 2.50

Form factors at lambda = 1.7365
f-prime = -5.00
f" = 4.50

Form factors at lambda = 0.9797
f-prime = 0.30
f" = 1.50

Fa Patterson from MADBST to be written to:
patterson.patt_2

Script file suitable for running SOLVE written to:
solve_mad.script

Datafile for SOLVE with MADMRG-compressed dataset
("Fnat",sig,"Fder",sig,"Delano",sig,iso diffs, ano diffs,
<Fhcos>, <Fhsin> from MADBST) is:
solve.data_2

Combining a total of 0 MIR and 2

MAD datasets to form a composite dataset

----------NEW DATASET BEGINS HERE----------

**** SOLVE: Solutions to MIR or SIR datasets ******

Derivatives considered: 3 (NSET)
Cross-vectors tested in HASSP: 6 (ICRMAX, DEFAULT=20)
HASSP solutions saved per deriv: 30 (NTOPHASSP, DEFAULT=30)
Fourier peaks saved per map: 30 (NTOPFOUR, DEFAULT=10)
Sites per derivative: 1 (NSOLSITE, DEFAULT=20)
Derivative solutions per seed: 5 (NTOPDERIV, DEFAULT=5)
Seeds per derivative tested: 3 (NSEEDTEST,DEFAULT=10)
Sorted seeds to use 5 (NSEEDSOLVE, DEFAULT=5)
Number of final solutions saved: 5 (NTOPSOLVE, DEFAULT=5)

Sites per derivative vary with derivative.

Derivative	Max sites
1	1
2	-1
3	1

Solutions obtained will be compared to input solution (ICHECKSOLVE)
Correlated phasing used (CORRELPHASE)
Patterson map for derivative 1 will be read directly from:
patterson.patt

Patterson map for derivative 3 will be read directly from:
patterson.patt_2

For derivative 1 the heavy atom
structure factor components parallel to and
perpendicular to the native structure factor will be read from columns
9 and 10
Standard difference fouriers will be calculated for derivative 2
For derivative 3 the heavy atom
structure factor components parallel to and
perpendicular to the native structure factor will be read from columns
19 and 20
For derivative 3 the corresponding native data
will be read from columns 11 and 12
For derivative 3 the corresponding native dataset is "derivative"
2

Datafile with 20 columns of data:
Title:solve.data (cols 1 to 10) and solve.data_2
Data: madmrg: MOCK FNAT

```
Data: madmrg: MOCK sig FNAT
Data: madmrg: MOCK FDER
Data: madmrg: MOCK sig FDER
Data: madmrg: MOCK DEL ANO
Data: madmrg: MOCK sig DEL ANO
Data: madmrg: Del iso for Patterson
Data: madmrg: Sigma of del iso for Patterson
Data: <Fa cos theta> = Fa component along Fo weighted by fom
Data: <Fa sin theta> = weighted Fa component perpendicular to Fo
Data: madmrg: MOCK FNAT
Data: madmrg: MOCK sig FNAT
Data: madmrg: MOCK FDER
Data: madmrg: MOCK sig FDER
Data: madmrg: MOCK DEL ANO
Data: madmrg: MOCK sig DEL ANO
Data: madmrg: Del iso for Patterson
Data: madmrg: Sigma of del iso for Patterson
Data: <Fa cos theta> = Fa component along Fo weighted by fom
Data: <Fa sin theta> = weighted Fa component perpendicular to Fo

Fnat,sigma taken from columns          1          2
Fder,sig,Delano,sig deriv          1 from cols:          3          4
          5          6
Fder,sig,Delano,sig deriv          2 from cols:         11         12
          0          0
Fder,sig,Delano,sig deriv          3 from cols:         13         14
         15         16

Check solution to be compared to all solutions found:

Derivative          1:
 Site    X       Y      Z
   1   0.440   0.160   0.380

Derivative          2:
 Site    X       Y      Z

Derivative          3:
 Site    X       Y      Z
   1   0.150   0.330   0.400

*************************************************************

ANALYZE_SOLVE:  analysis of top          1 solutions



*****************************************************************


Solution          1

********************** Analysis of this solution *************
```

****** Analysis of non-randomness of native Fourier map ******

A. Maps with distinct solvent regions havea high standard deviation
of local r.m.s. electron density.
For this map the SD of this local r.m.s. is 0.2356218

B. Maps with distinct solvent regions also have a high correlation
of local r.m.s. electron density with
density at neighboring locations. Typical
values for poor maps in this
structure solution are 9.0876520E-02 +/- 3.9091922E-02
For this map the correlation of r.m.s. density
in neighboring boxes is 0.1658869
The correlation coefficient is used here in scoring.

Skew of the map is: 0.1755039

****** Analysis of derivative solutions with the difference Patterson ******
and with cross-validation difference Fouriers

Derivative # 1

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.443	0.167	0.375	96.025

Evaluation of this test soln with 1 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.885	0.000	-0.750	18441.7	18441.6	2

Overall quality of this Patterson soln = 6520.13
Overall quality of the fit to patterson = 0.621481E-04
Avg normalized peak height = 4610.43

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.440	0.160	0.380	0.486	18.353	17.48

Derivative # 2

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

```
 Site    x       y       z       occ       B     -- PEAK  HEIGHT --


 ------------------------------------------------
     Derivative #          3



List of sites analyzed for compatibility with difference Patterson

  PEAK        X        Y        Z     OPTIMIZED
                                     RELATIVE OCCUPANCY
    1     0.151     0.333     0.396      75.507

Evaluation of this test soln with    1 sites after optimizing
occupancy of each site

Cross-vectors for sites  1 and  1 (excluding origin; 1000 = 1 sigma):
 #      U        V      W      HEIGHT   PRED HEIGHT      SYMM#
  1   -0.302   0.000  -0.792   11402.6     11402.6          2

Overall quality of this Patterson soln =  4031.42
Overall quality of the fit to patterson = 0.690534E-06
Avg normalized peak height =  2850.65


Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

 Site    x       y       z       occ       B     -- PEAK  HEIGHT --

   1   0.150   0.333   0.400   0.433  18.475            22.08

Summary of scoring for this solution:
                        -- over many solutions--    -- this solution --
Criteria                     MEAN          SD         VALUE        Z-SCORE
Pattersons:                   1.86         1.44         6.22         3.03
Cross-validation Fourier:    0.000E+00    0.500        33.0         65.9
NatFourier CCx100:            9.09         3.91        16.6         1.92
Mean figure of meritx100:    0.000E+00     12.1        66.3         5.49
Correction for Z-scores:                                -32.7

Overall Z-score value:                                   43.7


****** Overall analysis of phasing for solution          1************
```

```
HEAVY:  Refine heavy atom parameters

File title: 5-wavelength 2-ano scatterer MAD dataset   !  a title for this d
CRYSTALLOGRAPHIC PARAMETERS

    A =  76.00     B =  28.00     C =  42.00     alpha =  90.00     beta = 103.00     gamma =  90.00

PHASES CALCULATED EVERY   5 DEGREES
RESIDUALS CALCULATED ON EXTRA ZEROTH CYCLE ONLY
SIGMAS FROM data FILE WILL BE USED
STATISTICS WILL BE PRINTED ON ZEROTH CYCLE, SHIFTS ON LAST
PHASING WILL BE DONE TAKING INTO ACCOUNT THE CORRELATIONS AMONG DERIVATIVES
THE GROUPS OF DERIVATIVES WITH CORRELATIONS WILL BE UPDATED
THE BETA VALUES FOR EACH DERIV  WILL BE SET TO 1.0
PHASE-AVERAGED RESIDUALS WILL BE USED FOR PHASING

TYPE OF REFINEMENT SELECTED:
UNPHASED ORIGIN-REMOVED  PATTERSON REFINEMENT ONLY

Bayesian Correlated Phasing will be used

RESOLUTION LIMITS IN ANGSTROMS:      3.000    20.000
MINIMUM RATIO OF FDER TO RMS LACK-OF-CLOSURE FOR  INCLUSION IN REFINEMENT OR PHASING=      0.000
MINIMUM NATIVE F:      0.000
MINIMUM FIGURE OF MERIT FOR PHASED REFINEMENT:      0.000
MINIMUM ALLOWED ISOTROPIC B:      0.000
PARAMETER SHIFTS GREATER THAN     0.0000 TIMES SIGMA  WILL BE SCALED BY     0.5000
MINIMUM RATIO OF FNAT/SIGMA OR FDER/SIGMA TO INCLUDE:      1.000

NUMBER OF REFINEMENT CYCLES IS      2
DERIVATIVES REFINED DURING THESE CYCLES ARE :  0  0
TYPE OF OUTPUT SELECTED IS: +10 COLUMNS OF HENDRICKSON-LATTMAN COEFFICIENTS
1 INPUT data FILE WITH  20 COLUMNS IS:
combine.scl_1_2
COLUMN  0 : solve.data          (cols 1 to 10) and solve.data_2        ,cols 1 to 1
COLUMN  1 : madmrg: MOCK FNAT
COLUMN  2 : madmrg: MOCK sig FNAT
COLUMN  3 : madmrg: MOCK FDER
COLUMN  4 : madmrg: MOCK sig FDER
COLUMN  5 : madmrg: MOCK DEL ANO
COLUMN  6 : madmrg: MOCK sig DEL ANO
COLUMN  7 : madmrg: Del iso for Patterson
COLUMN  8 : madmrg: Sigma of del iso for Patterson
COLUMN  9 : <Fa cos theta> = Fa component along Fo weighted by fom
COLUMN 10 : <Fa sin theta> = weighted Fa component perpendicular to Fo
COLUMN 11 : madmrg: MOCK FNAT
COLUMN 12 : madmrg: MOCK sig FNAT
COLUMN 13 : madmrg: MOCK FDER
COLUMN 14 : madmrg: MOCK sig FDER
COLUMN 15 : madmrg: MOCK DEL ANO
COLUMN 16 : madmrg: MOCK sig DEL ANO
COLUMN 17 : madmrg: Del iso for Patterson
COLUMN 18 : madmrg: Sigma of del iso for Patterson
COLUMN 19 : <Fa cos theta> = Fa component along Fo weighted by fom
```

```
COLUMN 20 : <Fa sin theta> = weighted Fa component perpendicular to Fo


data COLUMNS FOR NATIVE F AND SIGMA:     1    2
data COLUMNS FOR BEST AND MOST PROB PHASES AND FIGURE OF MERIT:     0    0    0
OVERALL SCALE FACTOR FOR ALL data =     1.000
SCALE FACTOR FOR NATIVE SIGMAS =     1.000


DERIVATIVE INFORMATION FOR  3 COMPOUNDS


COMPOUND 1     Wavelength #  1        ! a label for this wavelength
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)    3    4    5    6
THIS DERIVATIVE WILL BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 2     Native from dataset # 2 (a MAD set) used as a deriv.
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)   11   12    0    0
THIS DERIVATIVE WILL BE USED IN PHASING
OVERALL SCALING FOR THIS DERIVATIVE WILL BE REFINED
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 3     set 3
COLUMNS FOR F, SIGMA, AND ANOM DIFF and sig (=F+ - F-)   13   14   15   16
THIS DERIVATIVE WILL BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
CARRYING OUT STANDARD REFINEMENT
Total of           2 cycles will be done
Derivs refined will be           0          0

SUMMARY OF RESULTS ON FINAL CYCLE:

NUMBER OF REFLECTIONS READ =    1822
NUMBER OF F .GT. FMIN      =    1813
NUMBER OF F IN RES. LIMITS =    1813
NUMBER OF F .GT. MIN <M>   =    1810
NUMBER OF F USED TO REFINE =       0



FIGURE OF MERIT <       0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1.0
# OF REFLECTIONS        81    96   110   111   114   123   162   202   357   456

FIGURE OF MERIT WITH RESOLUTION
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
```

```
N:              1812    105    150    200    218    246    277    297    319
MEAN FIG MERIT:   0.66   0.64   0.67   0.64   0.60   0.64   0.68   0.69   0.70
```

```
COMPOUND  1          Wavelength #  1         ! a label for this wavelength
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

CENTRIC REFLNS:   260.    33.    35.    31.    30.    36.    33.    33.    29.
RMS HA F:        24.5   33.6   27.9   28.3   21.5   24.3   20.7   16.7   17.2
RMS RESIDUAL:     19.9   32.7   18.2   22.2   19.8   19.4   13.7   13.7   10.1
RMS(FH)/RMS(E):   1.23   1.03   1.54   1.27   1.09   1.25   1.51   1.22   1.70
CENTRIC R FACT:   0.46   0.48   0.44   0.42   0.52   0.54   0.51   0.42   0.32

ACENTRIC REFLN:  1540.    72.   115.   169.   187.   210.   239.   260.   288.
RMS DERIV FPH:   192.1  319.2  221.8  232.5  228.4  199.3  158.0  146.6  132.2
RMS SIGMA FPH:    23.4   30.0   44.2   20.4   23.7   22.5   19.7   17.9   18.1
RMS SIGMA FP:     23.7   30.6   44.4   21.0   24.1   22.9   20.0   18.2   18.3
RMS HA F:        22.6   32.5   28.1   27.9   24.1   22.2   20.9   18.7   16.8
RMS RESIDUAL:     21.1   26.5   43.9   23.2   20.4   19.6   15.0   13.5   14.5
RMS(FH)/RMS(E):   1.07   1.23   0.64   1.20   1.18   1.13   1.39   1.39   1.15

ANOM DIFFS:      1540.    72.   115.   169.   187.   210.   239.   260.   288.
RMS OBS DIFF:      9.4   12.5   11.6   11.0   10.4    9.6    8.8    7.8    7.2
RMS CALC DIFF:     6.8    8.8    7.8    8.2    6.9    6.9    6.7    6.1    5.5
RMS RESIDUAL:      7.3   10.5    8.7    8.9    8.7    7.1    6.2    5.6    5.6
RATIO ISO/ANO:    4.64   5.15   4.99   4.82   4.67   4.54   4.42   4.30   4.19

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:             0.0    0.0    0.0    0.0    0.0    0.0    2.8    0.0
ANOMALOUS LOC:           0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS FPH :            319.2  221.8  232.5  228.4  199.3  158.0  146.6  132.2
RMS FH  :             32.5   28.1   27.9   24.1   22.2   20.9   18.7   16.8
RMS SIGMA:            42.9   62.6   29.2   33.8   32.1   28.1   25.6   25.8
```

```
COMPOUND  2          Native from dataset # 2 (a MAD set) used as a deriv.
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

CENTRIC REFLNS:   245.    31.    32.    31.    27.    34.    32.    31.    27.
RMS HA F:         0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS RESIDUAL:     55.7   85.0   59.7   69.4   51.5   47.0   41.8   39.4   28.5
RMS(FH)/RMS(E):   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
CENTRIC R FACT:   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
```

```
ACENTRIC REFLN:  1530.    71.   114.   167.   188.   206.   241.   256.   287.
RMS DERIV FPH:   196.5  327.9  222.3  243.9  232.5  192.3  163.7  154.6  138.3
RMS SIGMA FPH:    14.7   30.0   17.4   18.4   16.9   13.7   11.3   10.3    9.4
RMS SIGMA FP:     23.7   30.2   44.6   21.1   24.0   22.5   20.1   18.4   18.4
RMS HA F:          0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS RESIDUAL:     44.7   70.2   69.3   54.0   43.9   43.6   36.3   33.2   33.3
RMS(FH)/RMS(E):   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:            75.1   53.3   23.9   44.8   40.8   37.9   37.0   26.1
RMS FPH :              327.9  222.3  243.9  232.5  192.3  163.7  154.6  138.3
RMS FH  :               0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS SIGMA:              42.6   47.8   28.0   29.4   26.4   23.1   21.1   20.7




COMPOUND  3          set 3
DMIN:          TOTAL    9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10

CENTRIC REFLNS:   257.    32.    34.    31.    29.    36.    33.    33.    29.
RMS HA F:         18.0   23.9   21.7   20.7   17.0   15.9   15.5   12.6   12.8
RMS RESIDUAL:     39.0   59.6   41.4   47.4   37.6   32.4   31.6   28.4   18.5
RMS(FH)/RMS(E):   0.46   0.40   0.52   0.44   0.45   0.49   0.49   0.44   0.69
CENTRIC R FACT:   0.66   0.73   0.53   0.67   0.65   0.75   0.68   0.65   0.56

ACENTRIC REFLN:  1544.    71.   115.   169.   188.   206.   244.   261.   290.
RMS DERIV FPH:   193.1  317.8  223.0  234.6  230.9  194.7  161.5  150.0  135.6
RMS SIGMA FPH:    14.5   29.8   16.9   18.1   16.7   13.6   11.1   10.0    9.2
RMS SIGMA FP:     23.6   30.2   44.4   21.0   24.0   22.5   20.0   18.2   18.3
RMS HA F:         16.4   24.1   21.2   19.6   17.3   16.4   14.5   13.7   12.3
RMS RESIDUAL:     30.5   44.5   52.5   36.0   29.4   28.9   24.2   22.4   23.1
RMS(FH)/RMS(E):   0.54   0.54   0.40   0.54   0.59   0.57   0.60   0.61   0.53

ANOM DIFFS:      1544.    71.   115.   169.   188.   206.   244.   261.   290.
RMS OBS DIFF:     10.4   17.4   12.0   12.4   11.3    9.8    9.0    9.1    8.1
RMS CALC DIFF:     5.8    7.5    6.9    6.8    6.0    5.9    5.4    5.2    4.9
RMS RESIDUAL:      8.8   15.7   10.4   10.7    9.7    8.4    7.5    7.2    6.4
RATIO ISO/ANO:    4.03   4.50   4.34   4.19   4.05   3.93   3.83   3.73   3.64

ESTIMATES OF LACK-OF-CLOSURE RESIDUALS LESS AVERAGE VALUE OF
SIGMAS IN data (AS INPUT TO NEXT CYCLE)
CENTRIC LOC:            45.0   32.5    0.0   28.6   23.3   26.5   25.2   15.0
ANOMALOUS LOC:           0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
RMS FPH :              317.8  223.0  234.6  230.9  194.7  161.5  150.0  135.6
RMS FH  :              24.1   21.2   19.6   17.3   16.4   14.5   13.7   12.3
RMS SIGMA:              42.5   47.5   27.7   29.3   26.3   22.8   20.8   20.5
```

Analysis of correlated modeling and non-isomorphism errors

obtained using phased residuals.
The derivatives were grouped into 2 sets where the members of a set had some mutual correlation.

Set 1 contains derivatives 1

Set 2 contains derivatives 2 3

SUMMARY OF CORRELATED ERRORS AMONG DERIVATIVES

```
DERIVATIVE:          1
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
RMS errors correlated and uncorrelated with others in group:
    Correlated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  Uncorrelated:   1.0    0.0    0.0    0.0    0.0    0.0    0.0    2.8    0.0

Correlation of errors with other derivs:
DERIV 2:         0.20   0.44   0.28   0.09   0.34   0.20   0.03   0.41   0.19
DERIV 3:         0.21   0.47   0.27   0.08   0.37   0.26   0.16   0.37   0.19

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
Errors correlated and uncorrelated with others in group:
    Correlated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  Uncorrelated:   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 2:         0.36   0.30   0.51   0.47   0.33   0.33   0.25   0.23   0.29
DERIV 3:         0.39   0.33   0.58   0.49   0.35   0.35   0.26   0.22   0.32

DERIVATIVE:          2
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
RMS errors correlated and uncorrelated with others in group:
    Correlated:  34.7   60.1   41.4    0.0   34.9   31.6   29.0   28.5   18.7
  Uncorrelated:  29.0   45.1   33.6   23.9   28.2   25.9   24.4   23.5   18.3

Correlation of errors with other derivs:
DERIV 1:         0.20   0.44   0.28   0.09   0.34   0.20   0.03   0.41   0.19
DERIV 3:         0.81   0.91   0.83   0.67   0.82   0.86   0.80   0.81   0.81

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
Errors correlated and uncorrelated with others in group:
    Correlated:  36.4   59.7   52.8   50.8   33.6   35.1   28.0   25.0   26.7
  Uncorrelated:  33.4   51.4   47.3   40.9   31.6   34.2   28.2   26.4   25.4

Correlation of errors with other derivs:
DERIV 1:         0.36   0.30   0.51   0.47   0.33   0.33   0.25   0.23   0.29
DERIV 3:         0.90   0.87   0.93   0.89   0.89   0.90   0.91   0.87   0.91

DERIVATIVE:          3
```

```
CENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
RMS errors correlated and uncorrelated with others in group:
    Correlated:   34.9   60.3   41.6    0.0   35.2   31.8   29.1   28.6   18.8
  Uncorrelated:    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 1:         0.21   0.47   0.27   0.08   0.37   0.26   0.16   0.37   0.19
DERIV 2:         0.81   0.91   0.83   0.67   0.82   0.86   0.80   0.81   0.81

ACENTRIC REFLECTIONS:
DMIN:          ALL     9.91   6.56   5.22   4.46   3.96   3.60   3.32   3.10
Errors correlated and uncorrelated with others in group:
    Correlated:   36.5   59.7   53.2   50.9   33.6   35.1   28.2   25.2   26.8
  Uncorrelated:    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0

Correlation of errors with other derivs:
DERIV 1:         0.39   0.33   0.58   0.49   0.35   0.35   0.26   0.22   0.32
DERIV 2:         0.90   0.87   0.93   0.89   0.89   0.90   0.91   0.87   0.91


PARAMETER SHIFTS FOR DERIV  1 : Wavelength #  1        ! a label for this wavelength

                   SCALE FACTOR     OVERALL B
CURRENT VALUES:         1.0000         0.0000



              SITE  ATOM      OCCUP     X      Y      Z        B


CURRENT VALUES:     1    Fe      0.4865  0.4405  0.1600  0.3801   18.3530


PARAMETER SHIFTS FOR DERIV  2 : Native from dataset # 2 (a MAD set) used as a deriv.

                   SCALE FACTOR     OVERALL B
CURRENT VALUES:         1.0000         0.0000



              SITE  ATOM      OCCUP     X      Y      Z        B


CURRENT VALUES:     1    Fe      0.0100  0.0000  0.0000  0.0000    0.0000


PARAMETER SHIFTS FOR DERIV  3 : set 3

                   SCALE FACTOR     OVERALL B
CURRENT VALUES:         1.0000         0.0000
```

```
                SITE  ATOM      OCCUP     X       Y       Z        B


CURRENT VALUES:     1    Fe      0.4329  0.1499  0.3331  0.4001   18.4753



*************************************************************
*************************************************************


*** Summary of solutions and their relationships
 to each other and to check solution ***


-------------------------------------------------------
solution #          1 with overall quality =   43.69228
Derivative           1 with           1 sites.
Overall scale =    1.000000     and  overall b of   0.0000000E+00
 0.4404781      0.1600000      0.3801189      0.4864987       18.35298
Derivative           3 with           1 sites.
Overall scale =    1.000000     and  overall b of   0.0000000E+00
 0.1498983      0.3331375      0.4001377      0.4328635       18.47534
Best match of solution 1 -> solution 2:

-------- solution 1 --------    -------------solution 2 ------
site     x        y        z    site     x        y        z     DIST (A)
Derivative  1
 1    0.440    0.160    0.380    1    0.440    0.160    0.380    0.04
Derivative  3
 1    0.150    0.333    0.400    1    0.150    0.330    0.400    0.09
Comparison of this solution with check solution:
Number of sites in this solution matching check=          2
... and number not matching =           0
by derivative, this is...
Deriv          nsame          ndifferent
         1          1          0
         2          0          0
         3          1          0
All sites in this solution are contained in check soln
```

Contents	Index

All commands for SOLVE

This is a list of all the commands for SOLVE. Commands cause SOLVE to do something (solve a structure, scale data, search for heavy atoms, draw a map), while keywords set values of parameters (number of heavy atom sites, f' value)

Also see the list of common keywords that apply to automated SOLVE operation and the list of all keywords.

```
 ALIAS: set an alternate name for a keyword
 ANALYZE_MAD: analyze MAD data with MADBST and MADMRG
 ANALYZE_MIR: analyze MIR data
 AVG_OMIT: average key parts of a set of omit maps
 BTOF: convert from binary dorgbn file to formatted one

 CART_TO_FRACT: convert Cartesian coordinates to fractional
 COMBINE_ALL_DATA: put together several datasets into pseudo-MIR dataset
 COMPARE_SOLN:  compare two heavy-atom solutions and see if they are equivalent
 COMPLETE: determine completeness of a dataset
 DRGTOXPLOR: write out FOBS, SIGMA in X-PLOR format
 END: end the program
 EXPORT: write out h,k,l, data without any titles
 FDIFF: Create pseudo-mutant dataset for difference refinement
 FFTTOCCP4: convert asymmetric unit of FFT to asymmetric unit in ccp4 map format
 FFTTOEZD: convert asymmetric unit of FFT to any portion of cell in EZDNEW format
(for O)
 FFTTOMAPVIEW: convert asymmetric unit of FFT to any portion of cell in a format for
MAPVIEW
 FILEMERGE: combine or extract data columns from dorgbn files
 FTOB: convert from formatted dorgbn file to binary one
 FRACT_TO_CART: convert fractional coordinates to Cartesian
 GENERATE_MAD:  generate MAD dataset
 GENERATE_MIR:  generate MIR dataset
 GETANOM: convert from F+ and F- to Fbar and DelAnom
 GETISO: calculate differences between two data columns
 GETPHASES: convert from A,B coefficients to F and phase
 HA_PDB:  write out current set of heavy atoms in pdb format
 HASSP:  search for solutions to a patterson map
 HISTORY: list commands and keywords input to SOLVE
 HEAVY:  refine heavy atom parameters
 HELP: get information on this program
 IMPORT:  read in formatted file with h,k,l,data, stripping off any text
 LOCALSCALE: scale one dataset to another with local scaling
 MADBST: estimate heavy-atom structure factors from MAD data
 MADMRG: create pseudo-SIR+anom data from MAD data
 MAPS:  calculate Fourier and Patterson maps
 MAPTOASYM: Map a PDB file onto the asymmetric unit of the crystal
 MAPTOOBJECT: Map atoms to equivalent position close to  specified object
 MATH: Add, subtract, generate columns of data, used to  construct test datasets and
```

operate on data.
MERGE: merge equivalent reflections and write out asymmetric unit
NEW_DATASET: start reading in a new dataset to be combined later
NOLOG: stop writing log file

PEAKSEARCH: find peaks in Fourier map
PREFERENCES: display symmetry file, cell dimensions, and grids for FFT
RHO: Write out value of a map at positions of atoms in a PDB file.

SAD: solve a SAD dataset
SCALE_NATIVE: read in and scale native data for a MIR dataset
SCALE_MAD: read in and scale MAD data
SCALE_MIR: read in and scale MIR data following SCALE_NATIVE
SOLVE: solve an MIR or MAD dataset
STOP: end the program

VIEW: view a binary dorgbn file
WEIGHTS: set up weighting scheme for atomic refinement

Contents	Index

Automated structure determination with SOLVE

What you will need to input to SOLVE

To solve a structure automatically using SOLVE you need the following things:

- Your unit cell information and resolution.
- some raw datafiles in Scalepack or d*TREK or just a plain text format
- knowledge of what your heavy atoms are, and for MAD data, approximate values of the scattering factors (f' and f") at each wavelength. The best place to get these f' and f" numbers are from the beamline staff where you collected your data. If you have no other source of f' and f" information, you can use tables of values, but this won't work as well.

How to run SOLVE

To run SOLVE, all you need to do is:

- Set the environmental variable CCP4_OPEN to UNKNOWN with the following command:
  - setenv CCP4_OPEN UNKNOWN
  - (You can do this in your .login_custom file if you like)
- Choose a sample control file, edit it to match your data, and run SOLVE with it.
- You can watch SOLVE run by looking at the end of the "solve.status" with "tail -30f solve.status". This file will tell you where to look if something got typed in wrong and it will keep you informed about the structure determination as it goes.
- After SOLVE finishes, have a look at how to interpret your SOLVE output.
- If SOLVE is looking through more solutions than you want it to, and you would like SOLVE to finish up as quickly as possible, use the solve.control file to tell SOLVE to finish up.
- You may want to look at the keywords that you can use to control how SOLVE carries out its search for solutions.
- Also have a look at what to do next such as the ways that you can try and get more solutions
- Have a look at how to pass your data on to other packages after SOLVE has finished.

If you want to run solve more interactively or use it to do a host of other things such as peaksearches on maps, converting from binary to formatted files, or calculating coefficients for difference refinement,

have a look at the table of contents for a list of things you can do.

You might want to try SOLVE out on some test data in your space group using the generate feature. This allows you to create a MAD or MIR dataset with any heavy atom sites you want and then run SOLVE on it. If you start with a PDB coordinate file, you can generate a dataset, solve it, and use "O" to display the NEWEZD electron density map that SOLVE creates along with the correct structure.

Contents	Index

How to interpret the output from SOLVE

SOLVE writes out some files that have information on how the structure solution went and others that allow you to look at a map and export phases. Here are the most important files that SOLVE writes out when it is done:

- solve.prt (Summary of SOLVE results)
- solve.ezd (Portable electron density map)
- solve.ccp4_map (CCP4 electron density map)
- solve.mtz (ccp4 mtz file with F, Phases, Hendrickson-Lattman coefficients)
- phases-hl.export (Formatted file with phases and Hendrickson- Lattman coefficients ready to read into ccp4)
- phases-hl.script (Final parameters ready to use again in SOLVE)

Other files SOLVE writes out include:

- patt_Fa.ezd (Portable Fa Patterson, MAD data only)
- patt_iso_orig_removed_der1.ezd (Portable origin-removed isomorphous difference patterson, MIR only)
- patt_ano_orig_removed_der1.ezd (Portable origin-removed anomalous difference patterson, MIR with ano differences only)
- ha.pdb (heavy atoms in pdb format)
- solve.fft
- phases-hl.drg

solve.prt: an analysis of the top solutions found.

- If there is a really clear solution, then just one solution is written out
- For each solution, the solve.prt file lists (see examples):
    - the Patterson solution,
    - values of cross-validation difference Fouriers for each heavy atom site
    - phasing statistics (see HEAVY, for details)
    - Z-scores for
        - the Patterson
        - cross-validation difference Fouriers
        - analysis of the native Fourier
        - figure of merit

        - overall scoring
- You want to look for:
    - mostly positive peaks in your patterson solution,
    - high peaks (5 to 15 sigma) for all your cross-difference Fouriers
    - a decent figure of merit (0.5 or better)
    - Z-score for analysis of the native Fourier of 2 or more if you have good MAD data and a figure of merit over 0.65
    - an overall Z-score of 5-10 for a 2-site solution or 20-50 for a 10-site solution
    - If SOLVE printed out several solutions, then you should expect to see some duplication among the top solutions
        - this is because the solutions are each from a different seed and different seeds should give the same overall solution
        - See the analysis of relationships among solutions at the end of solve.prt
        - If you have no duplication at all and all solutions seem to have about the same poor scores, then SOLVE probably has not obtained the right solution
        - If there is one really high score or lots of duplication, then it is probably on the right track
- See what to do next after you've run SOLVE for how to look at your map, export your data, or improve your solution

Files for exporting output from SOLVE

solve.ezd: a portable native electron density map.

The "solve.ezd" file is an NEWEZD electron density map. Read this into "O" and you're ready to look at your electron density! In "O", just type,

```
@solve.ezd
ezd_draw
<cr>
<cr>
```

and you've got your map.

You can also read this map into the program mapman with the flag NEWEZD and convert it to ccp4 or other formats.

solve.ccp4_map: a CCP4-formatted native electron density map.

The solve.ccp4_map file is ready to be read into CCP4 programs for viewing or modification.

solve.mtz: a ccp4 mtz file with F, phases and Hendrickson- Lattman coefficients.

The "solve.mtz" file is a binary file in ccp4 mtz format containing Fnative, sigma, phases, figure of merit, and Hendrickson-Lattman coefficients calculated using Bayesian phasing. The column names in this file are:

- H K L FP SIGFP PHIB FOM HLA HLB HLC HLD
- H K L are indices
- FP SIGFP PHIB FOM are F, sigma, best phase, and figure of merit. For MAD data, the FP and phase are values estimated at the reference wavelength (usually the shortest wavelength).
- HLA HLB HLC HLD are Hendrickson-Lattman coefficients

phases-hl.export: a portable list of phases.

The "phases-hl.export" file is a formatted file containing Fnative, sigma, phases, figure of merit, and Hendrickson-Lattman coefficients calculated using Bayesian phasing. Copy and edit the header at the top into a command file and use it to read the "phases-hl.export" file into ccp4 mtz format. Then you're ready to do density modification or whatever you like with your map.

Also see Importing and Exporting data with SOLVE for more information about importing datafiles and maps.

phases-hl.script: a script file with heavy atom parameters

The routine also writes out a file "phases-hl.script" containing your refined parameters for the heavy atoms.

- You can paste the heavy atom parameters in phases-hl.script back into the solve_mad.script file or solve_mir.script file written by ANALYZE_MAD (or ANALYZE_MIR)
- Then you are set to continue on with ADDSOLVE or ANALYZE_SOLVE

patt_Fa.ezd -- a portable Fa Patterson map (MAD data)

patt_iso_orig_removed_der1.ezd and patt_ano_orig_removed_der1.ezd -- portable isomorphous and anomalous origin-removed patterson maps

These maps are in NEWEZD format and can be viewed with O or converted to other formats with mapman in the same way as the solve.ezd electron density map. (For derivative 1 the origin-removed maps are called ...der1 and for derivative 2 ...der2, etc). Note that if you have a multiple-MAD dataset the names of all these files will have _02, _03 etc appended to indicate the dataset number they came from.

ha.pdb -- heavy atoms in PDB format

Solve writes out the heavy atom positions for the top solution in PDB format in the file ha.pdb. For MAD data the heavy atom names are something like LAM1 (for lambda-1). The residue names are the derivative (or wavelength) number (DE1 means derivative 1).

solve.fft -- UCLA format map file

Here solve.fft is the same map as "solve.ezd", but in the ucla format.

phases-hl.drg -- binary output file

The "phases-hl.drg" contains the same information as "phases-hl.export" but it is in the binary ucla format. "phases-hl.drg" can be used to calculate electron density maps in this package or in another package.

Contents	Index

Generating and solving model datasets with errors

Generate allows you to construct an MIR or MAD dataset in which you specify the heavy atom locations and types. You can even specify the cell parameters for each derivative of an MIR dataset to simulate non-isomorphism. The output from GENERATE is suitable as input to SOLVE and you can run them in one script file to generate, then SOLVE, a dataset.

If you start with a PDB file in "coords.pdb" and specify "checksolve", then you can generate a dataset, solve it, and display the "solve.ezd" electron density map that SOLVE comes up with using "O". The map will automatically be referred to the same origin as the coords.pdb structure so you can overlay your map and the model to see how good the solution is. Please note that the EZD map will cover the asymmetric unit only. You may need to put your model in the asymmetric unit or else use Gerard Kleywegt's program mapman; manipulate your map (read it in to mapman as "NEWEZD") before you overlay the map.

Here are sample files that generate and solve MIR and MAD datasets. The keywords for generate_mir and generate_mad follow after the samples.

If you specify "checksolve" when you run one of these command files then SOLVE will automatically compare all the solutions it is getting with the one that you started with.

```
!-----------------------------------------------------------------
!gensolvemir.script
! command file to generate an MIR dataset and solve it

CELL 76 28 42 90 103 90
SYMFILE /usr/local/lib/solve/c2.sym
resolution 3.0 20.0
logfile gensolvemir.logfile
solvefile gensolvemir.prt
percent_error 3.0                       ! 3% error added to intensities
coordinatefile coords.pdb              ! coordinate file used to generate
                                        ! the starting I's (if none supplied,
                                        ! the routine makes up I's
deriv 1
cell_derivative  77 28 41 90 103 90    ! Try cell params for derivatives that
                                        ! are about 1% different from wt
inano
atom hg
occ 1.0
bvalue 31.
xyz 0.15 0.25 0.35

deriv 2
cell_derivative  75 28 42 90 102.5 90
inano
atom au
```

```
occ 0.8
bvalue 25.
xyz 0.33 0.15 0.17

GENERATE_MIR                            ! generate the MIR dataset now.

! Now the data are in: native.intensities, der1.intensities, and der2.intensities

!...  now analyze this MIR dataset...

rawnativefile native.intensities        !file for native data

premerged
readformatted

gotoder 1
rawderivfile der1.intensities           ! We have to use "gotoder" because we're in the
                                        ! middle of SOLVE, not starting from the
                                        ! beginning, and we have already specified
                                        ! more than one derivative.
gotoder 2
rawderivfile der2.intensities

nres 87                                 [approx # of residues in protein molecule]
nsolsite 1                              ! one site per derivative
checksolve                              ! compare the solutions to the correct one
comparisonfile native.fft         ! get correlation coefficient of map
                                 !calculated from each solution along the
                                 !way with the true map in native.fft

scale_native
scale_mir
analyze_mir
solve
!-----------------------------------------------------------------
```

... and now for a MAD dataset:

```
!-----------------------------------------------------------------
!gensolvemad.script
! command file to generate a MAD dataset and solve it
CELL 72 28 42 90 103 90
SYMFILE /usr/local/lib/solve/c2.sym
resolution 3.0 20.0
logfile gensolvemad.logfile
solvefile gensolvemad.prt
percent_error 3.0                       ! 3% error added to intensities
coordinatefile coords.pdb              ! coordinate file used to generate
                                        ! the starting I's (if none supplied,
                                        ! the routine makes up I's
mad_atom se                              ! define the scattering factors...
lambda 1
```

```
wavelength 0.90
fprimv_mad -1.6
fprprv_mad 3.4
atomname se
xyz 0.197 0.377 0.216
occ 1.0
bfactor 20
atomname se                             ! you only have to specify the coords for
                                        ! this one wavelength (they're copied to the
                                        ! others)
xyz 0.216 0.115 0.399
occ 1.0
bfactor 20
lambda 2
wavelength 0.9794
fprimv_mad -8.5
fprprv_mad 4.8
lambda 3
wavelength 0.9797
fprimv_mad -9.85
fprprv_mad  2.9

GENERATE_MAD                            ! generate the MAD dataset now.

! Now the data are in: lam1.intensities, lam2.intensities, and lam3.intensities for
!  the 3 wavelengths of data

! solve the dataset

premerged
readformatted

gotoder 1
rawmadfile lam1.intensities
gotoder 2
rawmadfile lam2.intensities
gotoder  3
rawmadfile lam3.intensities

nres 87                  [approx # of residues in protein molecule]
nanomalous 2
checksolve
comparisonfile lambda_1.fft         ! get correlation coefficient of map
                             !calculated from each solution along the
                             !way with the true map in lambda_1.fft

scale_mad
analyze_mad
solve
!-----------------------------------------------------------------
```

Notes on using GENERATE_MAD

You can have your generated MAD dataset contain more than one anomalously-scattering atom. You input information on the first atom type in the usual way as described above. For the second atom type, you need to:

- input one NEWATOMTYPE with scattering factors for each wavelength of MAD data. The NEWATOMTYPE for the various wavelengths must be of the Form PTL1 PTL2 PTL3 etc., for lambda 1, 2, 3.
- input the heavy atom parameters for this atom for lambda 1
- SOLVE will generate heavy atom parameters for this atom for all the other wavelengths and will include it in the generate procedure.

Keywords for GENERATE_MIR and GENERATE_MAD

```
coordinatefile          pdb file with coordinates.  Used to generate the starting
                        values of F and phase for the structure. Only C N O and S
                        atoms are read in.

percent_error           % error added to intensities

cell_derivative a b c alpha beta gamma    (only for generate_mir) cell parameters
                                          for this derivative.

derivative nn
lambda nn               derivative or wavelength number

atomname   xx           name of an atom about to be specified
xyz x y z               coordinates of this atom
bvalue b                b-factor
occupancy               occupancy value
```

Contents	Index

SCALE_MAD: Scaling MAD datasets

SCALE_MAD is a routine to read in MAD data from a Scalepack output file or from another formatted file and to scale it and put it all in a file suitable for ANALYZE_MAD and SOLVE. It is ordinarily called as the first step in a completely automated structure determination and is followed by ANALYZE_MAD and SOLVE. It can also be run on its own. The routine works well when the MAD data has been collected in very nearly the same way at all wavelengths and anomalous differences have been measured at all wavelengths either by inverse beam or by mirror plane symmetry. The routine assumes that either:

- you have already merged your anomalous data and your data files contain 4 columns (I+,sigma,I-, sigma), or
- you have not merged the data and your data columns contain 2 columns (I, sigma).

(Note: the data can be amplitudes, not intensities if you read in the data with READFORMATTED and you specify the flag READ_AMPLITUDES)

For the second case, reflections with H K L corresponding to I+ and I- are treated as such.

SCALE_MAD first estimates the overall absolute scale of the data using the data in the lowest resolution shell from the first datafile and the number of protein residues in the asymmetric unit (NRES). SCALE_MAD separates Bijvoet pairs into separate files. SCALE_MAD scales all the input data files to the first data file using just an overall scale and B. Then it merges all the data from all datafiles into one averaged dataset and rescales all the datasets to the averaged dataset using LOCALSCALE.

A typical input script for SCALE_MAD looks just like the one for automated MAD structure determination except you leave off the commands "ANALYZE_MAD" and "SOLVE".

```
                    KEYWORDS for SCALE_MAD:


LAMBDA n            The next RAWMADFILE(s) that are read in will be
                     for lambda n



MADFBARFILE xx.scl Output file with (Fbar,sigma,DelAno,sigma)
                     for each wavelength will be xx.scl
```

```
                    (DEFAULT="mad_fbar.scl")

MADFPFMFILE yy.scl Output file with (F+,sigma,F-,sigma) for each
                    wavelength will be yy.scl
                    (DEFAULT="mad_fpfm.scl")
```

See also the commonly-used keywords for automated SOLVE operation.

Contents	Index

ANALYZE_MAD: Analyzing a MAD dataset

ANALYZE_MAD is a routine to run MADMRG and MADBST. You usually do not have to worry about this routine at all because it is ordinarily called right after running SCALE_MAD for automatic structure determination, and all the parameters are already set for you. It is ordinarily followed by running SOLVE with the solve_mad.script file written out by this routine or by using the keyword "SOLVE" after running this routine.

ANALYZE_MAD assumes that you have a datafile ("mad_fbar.scl") that contains (Fbar,sigma,DelAno,sigma) for each wavelength of MAD data. That is, there are exactly 4 data columns for each wavelength. This is a dorgbn file. It is ordinarily created by SCALE_MAD, but you can create your own if you like.

To run ANALYZE_MAD, you need to input your standard setup file ("solve.setup"), and the wavelength to define as the standard (JSTD, see below). You need to input the scattering factors for the anomalously scattering atom at the various wavelengths too. You also need to input an estimate of the number of anomalously scattering atoms in the asymmetric unit (NANOMALOUS) and the number of protein residues in the asymmetric unit (NRES). It is only the ratio of NANOMALOUS to NRES that is important.

ANALYZE_MAD will run MADMRG (output ="madmrg.out") and MADBST ("madbst.out"). See the writeups below for these two routines. It will calculate origin-removed difference Pattersons for all dispersive and anomalous differences and Pattersons based on the output of MADMRG and MADBST. These maps are all compared to each other and the correlation coefficients are displayed in a table. In a typical MAD experiment the anomalous and dispersive difference Pattersons have correlations with each other on the order of 0.1 to 0.3 or so (pretty low, so don't be worried).

A typical input script for ANALYZE_MAD follows.

```
                    Script for ANALYZE_MAD

!--------------------------Run ANALYZE_MAD----------------------- -----------
@solve.setup                  !  standard information for this crystal

madfbarfile mad_fbar.scl      !  input dorgbn file with (Fbar,sig,Delano,Sig)
                              !    for each wavelength
madfpfmfile mad_fpfm.scl      !  input dorgbn file with (F+,sig,F-,Sig)
                              !    for each wavelength

logfile analyze_mad.logfile   !  write out most information to this file

mad_atom se                   !  atom type is selenium

fixscattfactors               ! do not refine scattering factors (you can if
                              ! you want and the data is good)
```

```
jstd 1                        ! Define wavelength #1 as the reference wavelength
lambda 1                      ! wavelength #1 information is to follow
label   Wavelength #1         ! label for lambda 1
wavelength 0.9000             ! wavelength value
fprimv_mad  -1.6              ! f' value at this wavelength
fprprv_mad  3.4               ! f" value at this wavelength

lambda 2
wavelength 0.9794
fprimv_mad  -8.5
fprprv_mad  4.8

lambda 3
wavelength 0.9797
fprimv_mad  -9.85
fprprv_mad  2.86

nres 100                   [approx # of residues in asymmetric unit]
nanomalous 2               [approx # of anomalously scattering atoms in a.u.]

madmrgfile madmrg.out    ! write the SIRAS-like MAD dataset to madmrg.out
madbstfile madbst.out    ! write the Bayesian Patterson to madbst.out

ANALYZE_MAD              ! run MADMRG and MADBST and analyze all the Pattersons

!-----------------------------------------------------------------------------
```

Keywords for ANALYZE_MAD:

```
LOGFILE xx.logfile  log file for output will be xx.logfile

madfbarfile xx.scl  input dorgbn file with (Fbar,sig,Delano,Sig)
                        for each wavelength will be xx.scl
                        [Default="mad_fbar.scl"]
madfpfmfile xx.scl  input dorgbn file with (F+,sig,F-,Sig)
                        for each wavelength will be xx.scl
                        [Default="mad_fpfm.scl"]

madmrgfile xxx.out   write the SIRAS-like MAD dataset to xxx.out
                    [     default="madmrg.out"]

madbstfile yyy.out   write coefficients for a Bayesian Patterson to yyy.out
                         [default="madbst.out"]

SOLVEDATAFILE  xxx  Output datafile with MADMRG output and MADBST
                    output  combined together, suitable for use
                    with routine SOLVE, will be xxx.   (DEFAULT file name =
                    "solve.data") The datafile has the following
```

columns of data:

1	Fnative (from MADMRG)
2	Sigma of Fnative (from MADMRG)
3	Fderiv (from MADMRG)
4	Sigma of Fderiv (from MADMRG)
5	DelAno (from MADMRG)
6	Sigma of DelAno (from MADMRG)
7	<Fh cos(theta)> (from MADBST)
8	<Fh sin(theta)> (from MADBST)

SCRIPTFILE xxx — Output script file containing instructions for running SOLVE written to xxx [default="solve_mad.script"]. Starting SOLVE with this script is equivalent to following ANALYZE_MAD with SOLVE.

FIXSCATTFACTORS — Fix scattering factors at their input values. This is a good idea if you have a reasonable idea of the f' and f" values. [this is the default]

REFSCATTFACTORS — refine scattering factors f' and f". If you refine them, be sure to look at their new values at the end of the routine MADMRG and verify that they are reasonable.

mad_atom xx — anomalously scattering atom is "xx"

The atom types recognized by SOLVE are:
H, H-1, He, Li, Li+1, Be, Be+2, B, C, Cv, N, O, O-1,
F, F-1, Ne, Na, Na+1, Mg, Mg+2, Al, Al+3, Si, Siv, Si+4,
P, S, Cl, Cl-1, Ar, K, K+1, Ca, Ca+2, Sc, Sc+3, Ti, Ti+2,
Ti+3, Ti+4, V, V+2, V+3, V+5, Cr, Cr+2, Cr+3, Mn, Mn+2, Mn+3,
Mn+4, Fe, Fe+2, Fe+3, Co, Co+2, Co+3, Ni, Ni+2, Ni+3, Cu,
Cu+1, Cu+2, Zn, Zn+2, Ga, Ga+3, Ge, Ge+4, As, Se, Br,
Br-1, Kr, Rb, Rb+1, Sr, Sr+2, Y, Y+3, Zr, Zr+4, Nb, Nb+3,
Nb+5, Mo, Mo+3, Mo+5, Mo+6, Tc, Ru, Ru+3, Ru+4, Rh, Rh+3,
Rh+4, Pd, Pd+2, Pd+4, Ag, Ag+1, Ag+2, Cd, Cd+2, In, In+3,
Sn, Sn+2, Sn+4, Sb, Sb+3, Sb+5, Te, I, I-1, Xe, Cs, Cs+1,
Ba, Ba+2, La, La+3, Ce, Ce+3, Ce+4, Pr, Pr+3, Pr+4, Nd,
Nd+3, Pm, Pm+3, Sm, Sm+3, Eu, Eu+2, Eu+3, Gd, Gd+3, Tb,
Tb+3, Dy, Dy+3, Ho, Ho+3, Er, Er+3, Tm, Tm+3, Yb, Yb+2,
Yb+3, Lu, Lu+3, Hf, Hf+4, Ta, Ta+5, W, W+6, Re, Os, Os+4,
Ir, Ir+3, Ir+4, Pt, Pt+2, Pt+4, Au, Au+1, Au+3, Hg, Hg+1,
Hg+2, Tl, Tl+1, Tl+3, Pb, Pb+2, Pb+4, Bi, Bi+3, Bi+5, Po,
At, Rn, Fr, Ra, Ra+2, Ac, Ac+3, Th, Th+4, Pa, U, U+3, U+4,
U+6, Np, Np+3, Np+4, Np+6, Pu, Pu+3, Pu+4, Pu+6, Am, Cm, Bk, Cf

```
                    If xx is not recognized by SOLVE you need to
                    specify a newatomtype and then refer to it

lambda n            wavelength number (n=1,2,3...) for values
                        to be entered next of f' and f"

wavelength xx       wavelength for lambda xx (i.e., 0.9798)

fprimv_mad  xx      1 real number for f' value for anomalously
                    scattering atom at the current wavelength.
                    Wavelength is defined by the most recent
                    value of the keyword "LAMBDA".  You always
                    need to enter f' and f" explicitly for MAD
                    data.

fprprv_mad  xx      1 real number for f" value for anomalously
                    scattering atom at the current wavelength.
                    Wavelength is defined y the most recent value
                    of the keyword "LAMBDA"

JSTD  n             Wavelength "n" will be used as the standard
                    to which all data is referred (see MADMRG)
                    [default= the lowest wavelength]

NRES   n            # of residues in asymmetric unit
                    [default=100]

NANOMALOUS n        # of anomalously scattering atoms in asymmetric unit.
                    Used to estimate how big the Fa values might be

NSHELLS n           Number of shells for analysis is n
                    [default=10]

bayes               Use Bayesian MAD phasing at the
                       very end of SOLVE. Requires that madfpfmfile
                        exists.  It also requires that INPHASE be
                       specified for all wavelengths. (This is the default)

nobayes             Use the compressed MADMRG datafile for all
                       phasing when program gets to SOLVE.
```

Contents	Index

MADMRG: Conversion of MAD data to pseudo-SIR with anomalous scattering

[Keywords | Notes]

MADMRG is a routine for compressing a MAD dataset into a nearly-equivalent MIR + anomalous scattering pseudo-dataset. This is useful for refinement of parameters describing the anomalously scattering atoms for several reasons, not the least of which is the very large improvement in speed of refinement. This fast refinement is critical for the SOLVE automated structure determination routine. Ordinarily you will not run MADMRG by itself, but rather as part of ANALYZE_MAD or an automated structure determination. This means that you usually will not need to be familiar with all the keywords for MADMRG.

MADMRG reads in measurements of Fbar and the anomalous differences DelAno at several wavelengths for each reflection. From this data and the known values of f' and f" for the anomalously scattering atoms at these wavelengths, the program estimates (1) the magnitude of the structure factor corresponding to all atoms except the anomalous scatterer (Fo), (2) the "isomorphous" difference that would be measured +/- the anomalous scatterer at a standard wavelength, and (3) the anomalous difference that would be measured at this standard wavelength. In this way, the MAD data is converted to a form identical to that used in the analysis of SIR+anomalous differences data.

MADMRG assumes that structure factor due to anomalous scatterer is not large compared to that due to all other atoms. In this case iso differences among various wavelengths are proportional to differences in (f+f') for the anomalous scatterer, and ano diffs at each wavelength are proportional to f". MADMRG scales all the ano diffs to a common wavelength, then averages them. It takes all the iso diffs (e.g., L3-L1, L3-L2, L1-L2), and scales each iso diff by: (f+f' at std wavelength)/(difference in f+f' at the 2 wavelengths) to obtain estimate of what would be measured for the structure factor amplitude due to the entire structure at the standard wavelength minus the structure factor amplitude of the entire structure without the anomalously scattering atoms. That is, it estimates, delta Fiso (+/- ano scatterer at the standard wavelength). Finally, MADMRG obtains estimates of what delta Fiso would be at each wavelength by scaling std Fiso by f+f' at that lambda. This allows the program to obtain estimates of Fo, the structure factor amplitude due to all non-anomalously scattering atoms from each value of (Fbar - Fiso at that lambda). These estimates of Fo are averaged.

The result of these manipulations is a pseudo-SIR+anomalous differences dataset. The "native" structure factor amplitude is Fo, the estimate of the structure factor amplitude due to all non-anomalously scattering atoms at the standard wavelength. The "derivative" structure factor amplitude is Fo plus the isomorphous difference, delta Fiso, corresponding to the contribution of the anomalously scattering atoms at the standard wavelength. The anomalous difference is the averaged anomalous difference, scaled to the value at the standard wavelength. Generally, the standard wavelength is chosen to be one well away from the absorption edge of the anomalously scattering atoms, so that f' is small or negligible. This is not essential, however.

Keywords for MADMRG

```
NSHELLS   n      number of shells of resolution used to group data is
                   n(default=10)

INFILE  xxx      input file is xxx
OUTFILE xxx      output file is xxx

mad_atom xx              anomalously scattering atom is "xx".

lambda xx               wavelength number (1,2,3...) for values to be entered
                        next of f' and f" and for column numbers
wavelength  xx          wavelength value for lambda xx (e.g., 0.9798)
fprimv_mad  xx          1 real number for f' value for anomalously scattering
                        atom at the current wavelength.  Wavelength is defined
                        by the most recent value of the keyword "LAMBDA". You
                        need f' and f" regardless of whether you input
                        mad_atom or aval_mad...
fprprv_mad  xx          1 real number for f" value for anomalously scattering
                        atom at the current wavelength.  Wavelength is defined
                        by the most recent value of the keyword "LAMBDA"
LABEL           label for this  wavelength
NCOLFBAR   n     Fbar for this wavelength
NCOLSFBAR  n     sigma of Fbar
NCOLDELF   n     Del F ano (Fplus - Fminus)
NCOLSDELF  n     sigma of del F ano

JSTD    n        wavelength ID for wavelength to be considered the STANDARD
                 [default=1]
FIXSCATTFACTORS   Fix scattering factors at their input values
REFSCATTFACTORS   refine scattering factors f' and f"
                        [default=fix (i.e., do not refine them)]
```

Notes on MADMRG:

OUTPUT FILE: This a DORGBN-style output file containing 8 columns of data. They are:

1 madmrg est of Fp-zero ("Fnative")
2 madmrg sig of fp-zero ("sig of Fnative")
3 madmrg: MOCK FDER ("Fderiv";
equal to Fp-zero + del iso)
4 madmrg sig of del iso ("Sig of Fderiv")

5 madmrg est of del ano ("Delano")
6 madmrg sig of del ano ("Sig of Delano")
7 madmrg weighted est of del iso for Patterson
8 madmrg weighted est of del ano for Patterson

To use this data as MIR + anomalous differences, simply use columns 1 and 2 as Fp (native F) and sigma; columns 3 and 4 as Fder (derivative F) and sigma; and columns 5 and 6 as delano and sigma.

The last two output columns are for use in drawing Patterson maps only. Column 7 is equal to column 3 - column 1, multiplied by a weighting factor. Column 8 is equal to column 5, multiplied by a weighting factor. These weighting factors are based on a Bayesian weighting scheme. The mean square iso difference in a range of resolution, less the mean square sigma of this, is a good estimate of the mean square true iso difference, Del**2, in this shell. The observed iso differences are weighted down by a factor equal to Del**2/(Del**2+sigma**2). If sigma is small relative to Del, the weight will be about 1. If sigma is big relative to Del, the difference will be weighted down. A similar approach is taken for the ano differences. Look at the Patterson maps you get with columns 7 and 8 and compare them to the ones you get using column 3 - column 1 and column 5.

Please note: The output of MADMRG is set up to be used as "Mock" native and derivative data. When you refine heavy atom parameters using this mock dataset, you must define a heavy atom type that has scattering factors identical to those you use in MADMRG at the "standard" wavelength. That is, if you define lambda 3 as "standard" in madmrg and f" at lambda 3 is 8.9933, then when you get to heavy atom refinement with routine HEAVY you will need to define an atom type "L3" (or something) that has all the right scattering factor information including an f" of 8.9933. Use the keyword NEWATOMTYPE in HEAVY to do this easily.

Also note: when you use this data in your MIR program, DO NOT refine an overall scale factor and B for the "derivative." The overall scale factor and B of the derivative relative to the (pseudo) native are absolutely perfect to start with (because of the way the derivative has been set up). In this package, use the flag "NOREFINESCALE" for the derivative.

The reason to use column 4 as sigma of Fder is that heavy atom refinement programs such as HEAVY assume that errors in Fp and Fder are independent. In this case they are not. Suppose you estimated the error in Fder by combining errors in Fo and deliso. Then your heavy atom refinement program would estimate the error in Fder-Fnat by combining the errors you give it for Fder (based on errors in deliso + Fo) and the errors you give it for Fnat (the error in Fo). The estimates of errors in Fder-Fnat would therefore contain the errors in Fo twice. If you use column 4 as sigma of Fder, the errors in Fder-Fnat will be correctly calculated based on deliso and Fo.

Input data file

This input data must be scaled carefully. MADMRG does not scale your data for you.

Number of protein residues and anomalous scatterers

The program assumes that the B-factor for the anomalously scattering atoms is similar to that for all other atoms. Using the number of protein residues and the number of anomalously scattering atoms on the next line, the program estimates the rms value of structure factor amplitudes due to anomalously scattering atoms as a function of resolution. Each of these are for the asymmetric unit.

Contents	Index

MADBST: Calculation of Bayesian Fa values from MAD data

[How MADBST works | Output | Keywords]

Madbst reads in MAD data from a dorgbn file. It estimates the components of the heavy (anomalously scattering) atom structure factor (Fa) parallel to and perpendicular to the structure factor for all non-anomalously scattering atoms. It also estimates the magnitude of Fa and its uncertainty.

Generally you will want to use "ANALYZE_MAD" to run "MADMRG" and "MADBST" for you. Ordinarily, you will take your scaled MAD data (the exact same data you have used for MADMRG) and use it with MADBST. The output of MADBST can be used in (1) Patterson maps, (2) difference fouriers, (3) direct methods, and (4) in automated structure determination with SOLVE in this package.

How MADBST works

Here is how MADBST works. MADBST first estimates rms heavy atom structure factor for anom scattering atoms at each wavelength used from form factor tables and # of atoms. Then MADBST goes through all the reflections and estimates probability distributions for the components of Fh parallel (Fha) and perpendicular (Fhb) to the structure factor corresponding to non-anomalously scattering atoms. Possible values of Fha, Fhb are deduced from rms value of Fh and Wilson statistics. Range of Fh tested are from -3 sigma to +3 sigma. Relative probabilities of each possible (Fha,Fhb) are calculated from Wilson statistics.

For each reflection, all above possiblities for (Fha, Fhb) are tested. For each (Fha,Fhb), values of Fp (structure factor for protein atoms only, assumed to be along the x-axis as we have no information on it) are tested. For any set of values of (Fha,Fhb) and Fp, it is easy to calculate values of Fbar and DelFano at each wavelength. The relative likelihood that a particular set of values of (Fha,Fhb) and Fp is correct is estimated (from Bayes' Rule), as the weighted residual: exp(- sum of (calc - obs)/sigma**2) for Fbar and DelfAno at all wavelengths. The a priori probability of (Fha,Fhb) is also included (Wilson statistics). The final "best" value of (Fha,Fhb) is just the weighted average over all possibilities. Similarly the "best" FH**2 is the weighted average of FH**2 over all possibilities (note that the best (FH**2) is not the (best FH)**2. Reflections for which no values of Fha,Fhb are likely (P<0.001) are rejected.

MADBST output

The output file is a copy of the input file, with 6 data columns appended to it, and with all reflections out of the resolution range or with no data tossed. The extra 6 data columns are:

ncol+1 -- <Fa cos theta> = Fh component along Fo weighted by figure of merit
ncol+2 -- <Fa sin theta> = weighted Fh component perpendicular to Fo
ncol+3 -- <Fa> = best estimate of Fa
ncol+4 -- sigma of <Fa>
ncol+5 -- sqrt(<Fa**2>) = sqrt of best estimate of Fa**2
ncol+6 -- sigma of sqrt(<Fa**2>)

The first two of these data columns can be used in difference Fourier maps to show the positions of anomalously scattering atoms. For example, if you have an estimate of phases for Fo, the non-anomalously scattering part of the structure from some partial solution, then you can add the phase angle for Fo on to the phase angle for Fa and draw a Fourier for Fa, the anomalously scattering atoms. This is done in SOLVE.

The third or fifth data columns can be used in a Patterson map (after squaring Fa) or in direct methods. SOLVE can use this Patterson map that you calculate.

Keywords for MADBST

(many are the same as for MADMRG):

```
nshells xx              # of shells of resolution to use (usu. 5 to max of 10)
nres  xx                # of protein residues in the asymmetric unit
nanomalous  xx          # of anomalously scattering atoms in the a.u.

mad_atom xx              Name of anomalously scattering atom.  This generates
                         aval_mad, bval_mad and cval_mad for you.

lambda n                wavelength # n (n=1,2,3..) for data to follow
ncolfbar n             column # of Fbar at wavelength 1
ncolsfbar n             column # of sigma of Fbar, wavelength 1
ncoldelf  n             column # of DelAno at wavelength 1
ncolsdelf n            column # of sigma fo DelAno at wavelength 1
                        ... (same for wavelength 2 etc...)
fprimv_mad  xx          1 real number for f' value for anomalously scattering
                        atom at the current wavelength.  Wavelength is defined
                        by the most recent value of the keyword "LAMBDA".
                        Note that f' and f" are required regardless of whether
                        you input aval_mad... or mad_atom
fprprv_mad  xx          1 real number for f" value for anomalously scattering
                        atom at the current wavelength.  Wavelength is defined
                        by the most recent value of the keyword "LAMBDA"

infile xxxx             Input data file (.drg file)
outfile xxxx            Output data file (.drg file, same as input except
                        reflections out of range are tossed and there are
                        6 additional columns of data)
```

Using SOLVE with MAD data analyzed with MADBST

If you want to phase the difference Fouriers calculated by SOLVE using both the anomalous and dispersive data data then you can specify,

NCOLFHCOS xx

NCOLFHSIN xx

which will, for lambda (1), use the value in ncolfhcos and ncolfhsin as estimates of the heavy atom structure factor components parallel to and perpendicular to the native structure factor. This is done automatically by ANALYZE_MAD if you are using automated structure determination. Here ncolfhcos(1) is identical to ncolfhcos. If you specify ncolfhcos(2) it refers to "derivative" 2.

The output of MADBST provides ncolfhcos(1) and ncolfhsin(1) as "<fa cos theta>" and "<fa sin theta>" (See MADBST writeup). You might also want to use a "combined" Bayesian patterson map as output by MADBST or an optimized difference Patterson map as calculated by MADMRG for your Patterson searches. If you want to specify a previously calculated Patterson map for lambda 1, use the command

PATTFFTFILE xxxxxx

where xxxxxx is the name of the FFT file containing this patterson. The FFT must have been calculated with this package using the same grid as currently specified. PATTFFTFILE is equivalent to PATTFFTFILE(1), where the 1 refers to lambda 1. This keyword will result in the use of file xxxxxx as the patterson for lambda #1.

Contents	Index

SCALE_NATIVE: Scaling native data

SCALE_NATIVE and SCALE_MIR are routines to read in MIR data from Scalepack output files or from other formatted files and to scale it and put it all in a file suitable for ANALYZE_MIR and SOLVE. They are ordinarily called as the first steps in a completely automated structure determination and are followed by ANALYZE_MIR and SOLVE. They can also be run on their own. The routines work best when the MIR data is reasonably complete. The routines assume that either:

- you have already merged your anomalous data and your data files contain 4 columns (I+,sigma,I-,sigma), or
- you have not merged the data and your data columns contain 2 columns (I, sigma).

(Note: the data can be amplitudes, not intensities if you read in the data with READFORMATTED and you specify the flag READ_AMPLITUDES)

For the second case, reflections with H K L corresponding to I+ and I- are treated as such. These choices are specified with the keywords "merged" or "unmerged". SCALE_NATIVE can read up to 4 data files and SCALE_MIR can read in up to 4 data files per derivative.

SCALE_NATIVE uses the first dataset you give it as a basis for setting the scaling of all the others. It reads in all the data, converts from I to F, maps it to the asymmetric unit, and uses the mapped data to localscale all the original unmapped datasets. The scaled datasets are then mapped to the asymmetric unit, merged, and written out to the file you specify with the keyword "scalednativefile" (default = "native_f.drg").

Sample script file for SCALE_NATIVE

```
!-----Command file to read in raw native data and scale it----

@solve.setup              ! get our standard information read in
logfile native.logfile    ! write out most information to this file.
rawnativefile native.int ! file with native data H K L Iobs Sigma usually
unmerged                  ! data have not yet been reduced to the a.u.
readdenzo                 ! data written by Scalepack
read_intensities        ! alternative is read_amplitudes
scalednativefile native_f.drg    ! output file with scaled native data
scale_native              ! scale this native dataset
!-----------------------------------------------------------------------------
```

Keywords for SCALE_NATIVE:

```
RAWNATIVEFILE xxx.int       File "xxx.int" will be read in as data for native


PREMERGED          The data in all RAWMADFILEs have H K L and 4 columns:
                   I+, sigma, I-, sigma

UNMERGED           The data in all RAWMADFILEs have H K L and 2 columns:
                   I, sigma

READDENZO          The datafiles are written by Scalepack.  For unmerged data
                   they will be read with the formatting:(6i4,i6,2i2,i3,2f8.0)
and
                   nsym*2+1 lines are skipped at the top of the file. For
                   merged data the formatting is:  (3i4,4f8.0) and
                   3 lines are skipped at the top of the file.

READFORMATTED      The datafiles were not written by Scalepack.
                    They will be read with "*" formatting.

READ_INTENSITIES     The data are intensities (default)
READ_AMPLITUDES    The data are amplitudes
                   (This is valid only with READFORMATTED)


NSKIP n              Skip exactly n lines at the top of each data file

NSKIP 0              Do not skip any lines at the top of each data file

NSKIP -1             Skip 0 lines at the top of each data file
                     unless the keywords READDENZO and PREMERGED
                     are set, in which case,skip NS*4+1 lines
                     where NS is the number of symmetry elements
                     in the space group (this is the # of lines in
                     a Scalepack header) [Default: NSKIP = -1]

RATMIN xx             Minumum ratio of F/sig to read in data for a
                      reflection at all is xx [default=0.0]. This is
                      useful for eliminating weak data.

OVERALLSCALE          Do not do local scaling; just an overall
                      scale factor for F+, F- at each wavelength.
                      Use this if you already have scaled the data
                      and you don't want any more scaling done.

SCALEDNATIVEFILE aa.drg       Scaled native data Fnat,sig will be written to and
                              read from "aa.drg" (default="native_f.drg")
```

Contents	Index

SCALE_MIR: Scaling derivative data to a native dataset

SCALE_NATIVE and SCALE_MIR are routines to read in MIR data from Scalepack output files or from other formatted files and to scale it and put it all in a file suitable for ANALYZE_MIR and SOLVE. They are ordinarily called as the first steps in a completely automated structure determination and are followed by ANALYZE_MIR and SOLVE. They can also be run on their own. The routines work best when the MIR data is reasonably complete. The routines assume that either:

- you have already merged your anomalous data and your data files contain 4 columns (I+,sigma,I-,sigma), or
- you have not merged the data and your data columns contain 2 columns (I, sigma).

(Note: the data can be amplitudes, not intensities if you read in the data with READFORMATTED and you specify the flag READ_AMPLITUDES)

For the second case, reflections with H K L corresponding to I+ and I- are treated as such. These choices are specified with the keywords "merged" or "unmerged". SCALE_NATIVE can read up to 4 data files and SCALE_MIR can read in up to 4 data files per derivative.

SCALE_MIR reads in and converts the data from I to F, scales all the F- data in the derivative datasets to the F+ data, then it scales each derivative (F-, F+) to the native. All the scaling is done using LOCALSCALE. The F- and F+ data for the derivatives are separately merged and use to create two data files. The first contains Fnat,sigma, and (Fbar,sig,Delanom, sig) for each derivative. It is specified by the keyword "mirfbarfile" and is usually called "mir_fbar.drg". The other contains Fnat,sigma, and (F+,sig,F-,sig) for each derivative. It is specified by the keyword "mirfpfmfile" and is usually called "mir_fpfm.drg". If you don't want to use anomalous differences later, you can specify the "noinano" keyword for that derivative in SOLVE. At the end of the output for SCALE_MIR will be a summary of the scaling R-factors for isomorphous and anomalous differences and of the completeness of the datasets.

Sample script file for SCALE_MIR

```
!-----Command file to read in raw derivative data and scale it to the native----

@solve.setup                        ! get our standard information read in
logfile deriv.logfile               ! write out most information to this file.

scalednativefile native_f.drg         ! you need a native to scale the derivatives...

derivative 1                        ! about to enter information on derivative #1
rawderivfile deriv1.int             ! the derivative data is in deriv1.int

derivative 2                        ! next derivative
rawderivfile deriv2.int

unmerged                            ! data have not yet been reduced to the a.u.
readdenzo                           ! data written by Scalepack
read_intensities                  ! data are intensities (alternative is
```

```
read_amplitudes)
scale_mir                          ! scale the derivs to the native
!-----------------------------------------------------------------------------
```

Keywords for SCALE_MIR:

```
RAWNATIVEFILE xxx.int       File "xxx.int" will be read in as data for native

DERIVATIVE n                The next RAWMIRFILE(s) that are read in will be
                            for derivative n

RAWDERIVFILE xxx.int          File "xxx.int" will be read in as data for
                            the current derivative

PREMERGED          The data in all RAWMADFILEs have H K L and 4 columns:
                   I+, sigma, I-, sigma

UNMERGED           The data in all RAWMADFILEs have H K L and 2 columns:
                   I, sigma

READDENZO          The datafiles are written by Scalepack.  For unmerged data
                   they will be read with the formatting:(6i4,i6,2i2,i3,2f8.0) and
                   nsym*2+1 lines are skipped at the top of the file. For
                   merged data the formatting is:  (3i4,4f8.0) and
                   3 lines are skipped at the top of the file.

READFORMATTED      The datafiles were not written by Scalepack.
                    They will be read with "*" formatting.

READ_INTENSITIES     The data are intensities (default)
READ_AMPLITUDES    The data are amplitudes
                   (This is valid only with READFORMATTED)


NSKIP n              Skip exactly n lines at the top of each data file

NSKIP 0              Do not skip any lines at the top of each data file

NSKIP -1             Skip 0 lines at the top of each data file
                     unless the keywords READDENZO and PREMERGED
                     are set, in which case,skip NS*4+1 lines
                     where NS is the number of symmetry elements
                     in the space group (this is the # of lines in
                     a Scalepack header) [Default: NSKIP = -1]

RATMIN xx             Minumum ratio of F/sig to read in data for a
```

```
                        reflection at all is xx [default=0.0]. This is
                        useful for eliminating weak data.

FPFM_ONLY               Toss all acentric reflections where either F+
                        or F- is missing.

FP_OR_FM                Use F+ or F- as an estimate of Fbar if F+ and
                        F- are not both present. This is useful if
                        your data is not that complete. It is
                        better to obtain complete data however.
                        This is the default for MIR data.

OVERALLSCALE            Do not do local scaling; just an overall
                        scale factor for F+, F- at each wavelength.
                        Use this if you already have scaled the data
                        and you don't want any more scaling done.

SCALEDNATIVEFILE aa.drg       Scaled native data Fnat,sig will be
                              read from "aa.drg" (default="native_f.drg")

MIRFBARFILE xx.scl         Output file with Fnat,sig, and (Fbar,sigma,DelAno,sigma)
                           for each derivative will be xx.scl
                           (DEFAULT="mir_fbar.scl")

MIRFPFMFILE yy.scl         Output file with Fnat, sig, and (F+,sigma,F-,sigma)
                           for each derivative will be yy.scl
                           (DEFAULT="mir_fpfm.scl")
```

Contents	Index

ANALYZE_MIR: Analyzing an MIR dataset

ANALYZE_MIR is a routine to calculate difference Pattersons and anomalous difference Pattersons for an MIR dataset, to calculate correlation coefficients among them all, and to set up a "solve_mir.script" file that can be used to run SOLVE on the dataset. You usually do not have to worry about this routine at all because it is ordinarily called right after running SCALE_MIR for automatic structure determination. It is ordinarily followed by running SOLVE with the solve_mir.script file written out by this routine or by using the keyword "SOLVE" after running this routine.

ANALYZE_MIR assumes that you have a datafile ("mir_fbar.scl") that contains Fnat, sigma, and (Fbar,sigma,DelAno,sigma) for each derivative of MIR data. That is, there are exactly 2 columns of data for the native and 4 data columns for each derivative. This is a dorgbn file. It is ordinarily created by SCALE_MIR, but you can create your own if you like.

To run ANALYZE_MIR, you need to input your standard setup file ("solve.setup"), and the name of the mir datafile. ANALYZE_MIR will calculate origin-removed difference Pattersons for all isomorphous and anomalous differences. These maps are all compared to each other and the correlation coefficients are displayed in a table. In a typical MIR experiment the anomalous and isomorphous difference Pattersons have correlations with each other on the order of 0.0 to 0.2 or so (pretty low, so don't be worried).

ANALYZE_MIR writes out a script file "solve_mir.script" that you can use to go on with the SOLVE routine to solve this mir structure. You can edit this script file to modify it if you like or you can run it as is. You can also go right on with SOLVE by adding the command "SOLVE" after "ANALYZE_MIR".

Please note that there is a slight difference between running ANALYZE_MIR right after SCALE_MIR and in a separate session. If you run them right after each other, you should input all the information about each derivative at once, as in the sample script for automatic analysis of MIR data. (If you wanted to input information about derivative #1 in two places in your script file, you would need to use the keyword "gotoderiv 1" the second time. You can avoid this by putting all the information for derivative #1 together.)

Script for ANALYZE_MIR

A typical input script for ANALYZE_MIR follows:

```
!-------------------------Run ANALYZE_MIR--------------------------------

@solve.setup                          ! standard information for this crystal
mirfbarfile mir_fbar.scl              ! input dorgbn file with Fnat,sig, and
                                      ! (Fbar,sig,Delano,Sig) for each wavelength
logfile analyze_mir.logfile           ! write out most information to this file

derivative 1                          ! derivative #1 information is to follow
label deriv #1 Hg                     ! label for deriv 1
atomname HG                           ! this is an HG derivative
```

```
derivative 2
label deriv #2 Iodine
atomname I-

ANALYZE_MIR                          !analyze all the Pattersons and
                                     !write solve_mir.script

!-----------------------------------------------------------------------------
```

Keywords for ANALYZE_MIR:

```
LOGFILE xx.logfile                 log file for output will be xx.logfile

mirfbarfile xx.scl                  input dorgbn file with Fnat, sig, and
                                    (Fbar,sig,Delano,Sig)
                                    for each wavelength will be xx.scl
                                    [Default="mir_fbar.scl"]

SCRIPTFILE xxx                    Output script file containing instructions
                                  for running SOLVE written to xxx
                                  [default="solve_mir.script"]. Starting SOLVE with
                                  this script is equivalent to following
                                  ANALYZE_MIR with SOLVE.

derivative n                      derivative number (n=1,2,3...) for values
                                  to be entered next of atomname and label

label xxxx                        xxx is a label for this derivative

ATOMNAME XXXX                     XXXX is the atom name of the atoms in
                                  this derivative.  This name can be in
                                  SOLVE's database or you can
                                  enter it using the NEWATOMTYPE keyword.
                                  PLEASE NOTE: the f' and f" values assumed
                                  by SOLVE are for Cu Kalpha radiation (1.54 A).
                                  If you collected your MIR data at a synchrotron
                                  then you will want to define a new atom type
                                  with the appropriate values of f' (fprimv) and
                                  f" (fprprv).  See the HEAVY or SOLVE keyword
                                  lists for entering a NEWATOMTYPE.
                                  The atom types recognized by SOLVE are:
                                  H, H-1, He, Li, Li+1, Be, Be+2, B, C, Cv, N, O, O-1,
                                  F, F-1, Ne, Na, Na+1, Mg, Mg+2, Al, Al+3, Si, Siv, Si
+4,
                                  P, S, Cl, Cl-1, Ar, K, K+1, Ca, Ca+2, Sc, Sc+3, Ti,
Ti+2,
                                  Ti+3, Ti+4, V, V+2, V+3, V+5, Cr, Cr+2, Cr+3, Mn, Mn
+2, Mn+3,
                                  Mn+4, Fe, Fe+2, Fe+3, Co, Co+2, Co+3, Ni, Ni+2, Ni
+3, Cu,
```

Cu+1, Cu+2, Zn, Zn+2, Ga, Ga+3, Ge, Ge+4, As, Se, Br, Br-1, Kr, Rb, Rb+1, Sr, Sr+2, Y, Y+3, Zr, Zr+4, Nb, Nb+3, Nb+5, Mo, Mo+3, Mo+5, Mo+6, Tc, Ru, Ru+3, Ru+4, Rh, Rh+3, Rh+4, Pd, Pd+2, Pd+4, Ag, Ag+1, Ag+2, Cd, Cd+2, In, In+3, Sn, Sn+2, Sn+4, Sb, Sb+3, Sb+5, Te, I, I-1, Xe, Cs, Cs+1, Ba, Ba+2, La, La+3, Ce, Ce+3, Ce+4, Pr, Pr+3, Pr+4, Nd, Nd+3, Pm, Pm+3, Sm, Sm+3, Eu, Eu+2, Eu+3, Gd, Gd+3, Tb, Tb+3, Dy, Dy+3, Ho, Ho+3, Er, Er+3, Tm, Tm+3, Yb, Yb+2, Yb+3, Lu, Lu+3, Hf, Hf+4, Ta, Ta+5, W, W+6, Re, Os, Os+4, Ir, Ir+3, Ir+4, Pt, Pt+2, Pt+4, Au, Au+1, Au+3, Hg, Hg+1, Hg+2, Tl, Tl+1, Tl+3, Pb, Pb+2, Pb+4, Bi, Bi+3, Bi+5, Po, At, Rn, Fr, Ra, Ra+2, Ac, Ac+3, Th, Th+4, Pa, U, U+3, U+4, U+6, Np, Np+3, Np+4, Np+6, Pu, Pu+3, Pu+4, Pu+6, Am, Cm, Bk, Cf

NSHELLS n — Number of shells for analysis is n [default=10]

Contents	Index

HASSP: Patterson searches by the superposition method

[Script for HASSP | More about HASSP | Analyzing a solution]

HASSP is a program for automatically searching for solutions to a Patterson function. It is used by SOLVE as a means of generating plausible starting solutions for a MAD or MIR dataset. You can use it to analyze any Patterson or difference Patterson function that you have calculated with this package. Although HASSP is good at finding possible solutions to a Patterson, it is not as good as SOLVE at evaluating these solutions. It is a good idea to run HASSP on your Patterson functions to get an idea of what they look like, but you should really run SOLVE to get a complete solution to your dataset.

Using HASSP is very easy. Here is a script file that will run HASSP on a patterson that you have calculated and put into "patterson.patt".

Script for HASSP:

```
!------------------Script for running HASSP ---------------------------------
@solve.setup                  !  set up those standard keywords
fftfile patterson.patt        !  name of file with the patterson
logfile hassp.prt             !  write out the results to "hassp.prt"
hassp                         !  run hassp
!-----------------------------------------------------------------------------
```

HASSP will then analyze your patterson in "patterson.patt" and will write out a sorted list of likely solutions to this patterson.

More about HASSP

The HASSP routine uses a space-group symmetry minimum method to obtain sets of atomic sites consistent with a patterson function. The usual procedure followed in using this program is to search for single-atom solutions to patterson function, then to search for two-atom solutions to the patterson function. The two-atom solutions are obtained from two sources: combination of single-atom solutions (after figuring out origin shifts and translations along polar axes), and cross-vector searches. The idea behind a cross-vector search is that many of the peaks in general positions of a patterson are cross-vectors between sites. If you know the cross-vector between two sites then once you know the position of 1 site you know the position of the second. Considering a particular cross-vector, HASSP tests all possible positions for atom 1, generates the position of atom 2 and all the predicted peaks it the patterson. These are compared to the patterson itself and the solution is scored.

Searching for single-site solutions to patterson function.

A map is calculated over the range supplied (XS-XE; YS-YE; ZS-ZE), except that search is not carried out over axis which are not fixed (all three in space group P1). The value of the map at each grid point is the minimum of values of the patterson functon at the (NEQUIV-1) Harker vectors corresponding to this grid point. Peaks in this map are stored, sorted according to symmetry, and listed after elimination of redundancies.

For points in general positions, the peak height listed is simply the minimum value of the patterson function at the (NEQUIV-1) Harker vector associated with this point. For points in special positions, the listed height is the minimum of the values of the patterson function at each of the (NEQUIV-1) Harker vectors divided by the number of times a Harker vector associated with this point falls on that position. For example, in space group P222, an atom at (x,y,z) yields Harker vectors (0,0,0), (2x,2y,0), (2x,0,2z), and (0,2y,2z). If x=0 and y=0, though, (0,0,0)= (2x,2y,0) and (2x,0,2z)=(0,2y,2z) and there is only one unique Harker vector (excluding the origin), which is repeated twice. The value of the peak height listed would be 1/2 the height at (0,0,2z).

The probability that a given peak of height A in this function is due to a random combination of peaks is roughly given by:

P=(1.- (1.- p(A)**M)**N) , where,

- A= minimum value of (value of patterson function at Harker vectors divided by expected noise at that position).
- p(A) is probability of observing a value of A or higher on a given try.
- M=number of independent Harker vectors examined for this peak
- N=number of independent grid points used in search for peaks.

The noise in the map is taken to be the RMS value of the patterson function if this is a general position. If it is a position of higher symmetry, the noise = sqrt(SIGMA) * the symmetry number of this position. The number of independent grid points used in the search for peaks would roughly be equal to the number of reflections used to make up the map if reflections at all resolutions contributed equally. A better estimate of this numbr is probably the number of peaks+valleys in the patterson map. In this routine, we actually use 2* the number of peaks.

The grid used for all searches is exactly the same as the grid for the input patterson map, but each time a peak is found, all neighboring grid points are tested on a grid twice as fine and the highest of these test values is used. Values of the patterson function between grid points are interpolated. Do not use a grid coarser than 1/3 the resolution for the input patterson map. Also don't bother to use a grid finer than 1/6 the resolution. NOTE: the input patterson map must be on a grid such that the symmetry elements lie on grid points. That is, if there is a two-fold axis at 1/12 in z, then the z-axis must be divided into a number of grid points that is a multiple of 12. The easiest way to be safe is to make sure all unit cell translations are multiples of 12.

Significance tests

Difference patterson functions have a considerable amount of noise if acentric reflections are present. (For each acentric reflection, the expected error [|Fph-Fp| - |fh|] is roughly equal to |Fph-Fp|). It can be shown that SIGMA, the RMS noise in the map is roughly equal to the RMS value of the patterson function.

Peaks in the patterson map which have a height much less than SIGMA are therefore likely to be unrelated to atomic sites. On special positions, the RMS noise in the map will be sqrt(NSYM)*SIGMA, where NSYM is the symmetry number of this position.

In order not to include too many peaks due to this noise in any of the searches carried out, a significance test is made for each peak if ISIGNF=0 (default). A peak is rejected if there is a probability less than SIGNIF that no peak of this height or higher would occur by chance in this search.

Symmetry numbers of positions in real and patterson cells

For this program, the symmetry number of a position in a real or patterson cell is the number of ways that a symmetry operator in the group (patterson or real cell) can map the point onto itself (within a toler- ance of 2 grid units). The symmetry number of a general position is 1, for a point on a dyad, it is 2, etc..

Local symmetry

Local symmetry is not yet implemented in SOLVE

.

Obtaining an analysis of a solution that you input yourself.

In order to generate a list of local symmetry-related points and minimum self- and cross-vectors corresponding to a set of unique sites you specify, use ITYPE = -6 with a very small search region (keyword searchregion; if you set it to zero, though, it will replace your zeroes with the asymmetric unit of the cell as defined by your FFTGRID).

If ITYPE=-6 is specified along with a small (but non-zero) search region, an analysis of the sites you input on lines 9a-... will be printed. This analysis includes the minimum values of the self- and cross-vectors for this set of sites. This procedure will help you determine if there is anything unusual about your map.

Contents	Index

HEAVY: Heavy atom refinement and phasing

[MAD phasing | Refinement | Rejecting data | Statistical output |]

[Correlated phasing | Typical cycle| Changing heavy atom parameters]

HEAVY is a general-purpose heavy atom refinement routine. It can be used to carry out either phase refinement or origin-removed Patterson refinement, as well as to calculate coefficients for native Fourier and difference Fourier maps. Ordinarily you will use HEAVY as part of an automated structure solution with SOLVE. In this case SOLVE will write out a "phases-hl.script" script file that you can edit and use for further refinement. This should usually mean that you will not have to generate the rather lengthy inputs to HEAVY by yourself.

Keywording inputs are most conveniently entered using a script file. Note that any values previously defined do not need to be specified. If you run HEAVY a second time without quitting the main program and do not specify any new parameters, the routine will start where it left off and carry out another set of refinements of the same type that you specified the last time you ran it. Note also that average residuals are maintained throughout. This means that if you want to refine a completely new set of data, you should start the program over.

An easy way to get a file containing all the keywords you can use for HEAVY is to edit the file generated by running HEAVY specifiying "newfile heavy.new".

MAD phasing with HEAVY

There are 3 ways to phase MAD data using HEAVY.

(1) You can use MADMRG to compress MAD data to data that look like SIR+anomalous data at one wavelength (e.g., L1), then refine heavy atom parameters and phase just as if you did have SIRAS data. This gives you F and phase for the structure without the anomalously scattering atoms.

(2) You can use MADMRG and then Bayesian phasing. This is what SOLVE does. In this case you convert MAD data to SIRAS using MADMRG, then refine heavy atom parameters all as in #1. Then you use these heavy atom parameters with the original MAD data to phase it using Bayesian MAD phasing. As you have already refined the heavy atom parameters at L1, you do not need to redo the refinement at L2, L3 because they are all the same. You simply run HEAVY again, specifying the keyword IMADPHASE n (where n=the wavelength # for which heavy atom structure factors are to be calculated), and specifying REFINENONE for each heavy atom site. You also need to specify a new input file that has (instead of the MADMRG data or FBAR,DELANO data) the complete F+,sig+,F-,sig- scaled MAD data. This can be the data file used to create the Fbar,delano data file. You specify column numbers with

NCOLFPLUS, NCOLSIGPLUS, NCOLFMINUS, NCOLSIGMINUS. Derivative 1 is now your L1 data, derivative 2, L2, etc. You include the refined heavy atom parameters for your L1 derivative and dummy atoms with the correct scattering factors for the L2 and L3 derivatives. The dummy atoms are used to calculate scattering at these other wavelengths; they are not used in phasing per se. You don't even have to put in any occupancies or xyz for these atoms. In this way, you have 1 set of heavy atom parameters that are applied to all 3 wavelengths. If your heavy atoms are not selenium at L1, L2, and L3, you will need to use the NEWATOMTYPE keyword to input their scattering factors.

The SOLVE routine with MAD data does all this without even leaving SOLVE. You can also run HEAVY once after MADMRG, then edit the HEAVY.NEW (or whatever you have called it) file as described above, then run HEAVY again with IMADPHASE specified.

If you specify IMADPHASE 1 then your output map will be calculated at the lambda of "derivative" 1, if you specify n then it will be at the wavelength of dataset n. The phasing will be the same in any case. Note that the value of "n" you specify will determine which heavy atom values are used in the MAD phasing calculation. If you specify "2" then the heavy atom parameters that you type in for derivative 2 will be used.

(3) Refinement as if the MAD data were MIR data. In this case, you choose one wavelength (the one with a small f" and the most negative f' usually, often called "L1") as "native" and treat the other wavelength data as derivatives. In this case you will need to define new "atoms" with the NEWATOMTYPE keyword that have values of f" that are actual values, but values of f' that are the difference between the value at that wavelength and the value at L1, and values of all the other parts (a1, b1, etc) of zero. For example, if the L1 values of f' and f" are -9.6 and 2.2, and at L2 they are -7.6 and 5.8, then you need a new atom type as follows for the f' difference:

```
NEWATOMTYPE L2L1
AVAL      0.00000000     0.00000000     0.00000000     0.00000000
BVAL      0.00000000     0.00000000     0.00000000     0.00000000
CVAL      0.00000000
FPRIMV      2.0
FPRPRV      5.8
```

You then use "L2L1" as your atomname for heavy atoms in derivative 2 (L2). If you have 3 wavelength MAD data, you now have 1 native and 2 derivatives. You can refine the heavy atom parameters of the 2 derivatives in just the usual way and obtain phases for the L1 data (including the heavy atoms) as if this were MIR data.

Refinement against the origin-removed Patterson map

Refinement against an origin-removed Patterson map is a way of refining heavy atom parameters of each derivative independently, and is particularly useful because the occupancies of heavy atom sites are quite accurately estimated and because the refinement is very fast. When using this package, the recommended refinement method is this one, with JALT=0 and KALT=0.

This refinement minimizes the sum over all reflections of,

R = WGT * DEL**2

with respect to heavy atom parameters. WGT is a weighting factor, and DEL is defined as:

DEL = (Fph-Fnat)**2 - K*FH**2 - < (Fph-Fnat)**2 - K*FH**2 >

where the average <> is taken in a shell of resolution and FH is the magnitude of the calculated heavy atom structure factor. K is 1 for centric reflections, 1/2 for acentric reflections.

Rejecting data with large Del F's:

HEAVY uses all the data that you give it that satisfy the criteria of minimum F, FOM, etc that are set above. If you want it to reject data with especially large Del F, then you need to specify this when you scale the data with LOCALSCALE. There is an option in localscale to "reject large del F" (TOSSBAD). Use this to get rid of the large Del F before going into HEAVY.

Interpreting statistical output from HEAVY

Many of the values listed at the end of a set of refinements are more-or-less self explanatory. This should include the number of reflections read, within resolution limits, and greater than the minimum figure of merit. As these statistics are usually printed for a cycle in which refinement is not carried out, the number of reflections used to refine is usually zero in this listing.

The statistical output for MAD phasing using Bayesian correlated MAD phasing is not as intuitive as the output for MIR phasing because the standard phasing statistics do not really apply. If are running SOLVE in automatic format and you want approximate phasing statistics, you can run SOLVE specifying "NOBAYES". This will suppress the Bayesian correlated MAD phasing at the end of SOLVE and use SIRAS phasing which isn't quite as good but for which the statistics are easy to understand.

Other values listed at the end of a set of refinements include:

RMS HEAVY ATOM F: The rms value of the calculated heavy atom F in the resolution range

RMS PHASE AVG'D RESIDUAL: This is the rms value of the difference between calculated and observed derivative F, where it is averaged not only over all reflections, but over all phases for each reflection, weighted by the phase probability

RMS(FH)/RMS(E): This is the ratio of the rms heavy atom F to the rms phase averaged residual

CENTRIC R FACTOR: This is <| |Fder-Fnat| - |FH| | >/< |Fder-Fnat| >

RMS DERIVATIVE F: This is the rms value of Fder

RMS SIGMA OF FPH: This is the rms sigma of Fder

RMS SIGMA OF FP: This is the rms sigma of Fnat

RMS OBSERVED DIFFERENCE: For anomalous differences, this is the rms value of DelAno= (F+ - F-)

RMS CALCULATED DIFFERENCE: This is the rms calculated anomalous difference

MEAN RATIO OF ISO TO ANO: This is the ratio of calculated |FH| due to normal scattering relative to that due to anomalous scattering. If all anomalous scatterers are identical, this is equal to (f+f')/f" for that anomalous scatterer.

RMS(RES HA SF+LACK OF ISO SF): This is an estimate of the total errors in the heavy atom model plus lack of isomorphism that remain. It is obtained from the rms phase averaged residual and the rms native and derivative sigmas.

RMS LACK OF ISOMORPHISM SF: This is an estimate of the remaining lack of isomorphism. It is based on a comparison of the anomalous and isomorphous differences that remain

RMS RESIDUAL HEAVY ATOM SF: This is an estimate of the remaining heavy atom structure factor, based on the anomalous differences and the errors in measurement.

CENTRIC LOC: This is an estimate of the "centric" lack-of-closure residual, obtained using both centric and acentric reflections and correcting acentric lack-of-closure residuals by a factor of 2. These residuals are all corrected for errors in measurement, so that if the derivative is "solved" and there is little lack of isomorphism, these values should all be near zero.

ANOMALOUS LOC: This is the lack-of-closure error for anomalous differences, corrected for errors in measurement.

Correlated phasing statistics

If you specify the keyword "CORRELPHASE" then HEAVY will use a routine in phasing that takes into account the correlations in non-isomorphism errors among the derivatives. The derivatives must be grouped into sets with correlated errors. You can specify this grouping using IEGROUP (see below) or you can let HEAVY group them for you using the flag GETGROUPS. Note that correlated phasing makes a major improvement in the phasing power of a set of derivatives if the errors are highly correlated (>50%). If they are not highly correlated, the routine yields essentially the same results as the standard routine.

Whether or not correlated phasing is being used, the correlation of errors among derivatives is analyzed by HEAVY. An example of part of a log file that illustrates this is shown below:

```
-----------------------------example-----------------------------------------
 Analysis of correlated modeling and non-isomorphism errors
```

```
obtained using phased residuals.
The derivatives were grouped into 1 sets  where the members of a set
had some mutual correlation.


Set 1 contains derivatives  1 2 3

SUMMARY OF CORRELATED ERRORS AMONG DERIVATIVES

DERIVATIVE:            1
CENTRIC REFLECTIONS:
DMIN:            ALL     10.81   6.94   5.46   4.65   4.11   3.73   3.43   3.20
RMS errors correlated and uncorrelated with others in group:
     Correlated:  363.5  322.2  291.7  253.8  458.7  434.5  371.2  404.0  337.8
   Uncorrelated:  285.9  362.3  340.3  292.9  288.9  279.9  229.0  201.9  174.7

Correlation of errors with other derivs:
DERIV 2:           0.83   0.66   0.76   0.70   0.88   0.89   0.89   0.98   1.00
DERIV 3:           0.74   0.59   0.64   0.51   0.78   0.83   0.82   0.86   0.95
-------------------------------------------------------------------------------
```

In this example, there are 3 derivatives, all in the same group (IEGROUP=1 for each). Of the lack-of-closure errors for derivative 1, most (363.5, arbitrary units) were correlated with derivatives 2 and 3, and some (285.9) were unique to this derivative. The overall correlation of errors with derivatives 2 and 3 were 83% and 74%, respectively. In fact, correlated phasing made a major improvement in the phasing for this group of derivatives.

Normal refinement/phasing cycles.

A. Refinement vs. origin-removed Patterson map.

Input parameters: all defaults used

NCYCLE = 1 to 30

IREFCY(I) = 1,1,1,2,2,2.....6,6,6,0

results:

Zeroth cycle: phases calculated for all derivatives identified with INPHASE using input lack-of-closure residuals. New lack-of-closure residuals are calculated for these derivatives. Statistics are printed.

Cycles 1 through NCYCLE-1: in this example, IREFCY(I) is zero onlast cycle, but non-zero for all other cycles. For each cycle when IREFCY(I) is non-zero: no phases are calculated no new residuals are calculated derivative IREFCY (I) is refined as described above

Note that only 1 derivative is refined at a time and all are independent. Therefore in polar space groups, the coordinate

(s) of at least one atom in each derivative must be fixed. In space group P1 parameters for a single heavy atom may not be refined at all. If two atoms are present, the occupancy, xyz, B of one of them only may be refined. If you use IHEAVYPROC then all this is taken care of for you (in P1 SOLVE will refine with phase refinement if any derivs have fewer than 3 sites, otherwise it will use patterson refinement).

Cycle NCYCLE: IREFCY(NCYCLE)=0 in this example, so this cycle is like the zeroth cycle: phases are calculated, new residuals calculated. If KOUT is non-zero, output data are calculated as well.

B. Refinement by minimization of lack-of-closure at most probable phase.

Input parameters: all default except JALT=1, KALT=0.

Results: identical to the above example except:

(1) phases will be calculated every cycle

(2) derivatives will be refined by minimization of (Fph-Fc)**2

This is not the recommended manner of using HEAVY in this package. In most circumstances origin-removed Patterson refinement is much more accurate. There are some instances in which phase refinement may be useful, however. One is when it is necessary to correlate the origins in different derivatives. In space group C2, for example, the y-coordinate is indeterminate. That means that if you have two derivatives and refine them independently, you will not have refined the relative y-coordinates of the atoms in the two derivatives (though you will have refined the relative y-coordinates of atoms within each derivative). You might wish to use phase refinement to carry this out, using one derivative to phase and refining y-coordinates in the other derivative. In practice, however, these relative y-coordinates can be obtained even more accurately by simply calculating a difference Fourier for one derivative, phasing with the other derivative. The centroid of the peak corresponding to the heavy atom site (which can be found, for example, by PEAKSEARCH in this package) will give you the relative y-coordinate you need with very good accuracy, and refinement of this coordinate is unnecessary. This is how SOLVE does this.

Note that still only 1 derivative may be refined at one time. (If you really want to phase only once per refinement of all derivatives, calculate phases during one run and write them out with KOUT=7. Then merge file containing phases with input DORGBN file (3 extra columns). Then run HEAVY with INPHAS=0 for each derivative and specifying INOLD=1. Also set INRESD=-1. The program will then use the input phases during phase refinement if JALT=1. Its probably faster to just phase each time.)

C. Just calculating phases and a map or other output.

Input: all default, except NCYCLE >0

If the input lack-of-closure residuals are ok., you can set INRESD = -1 so that new residuals will not be calculated and a zeroth cycle will not be included. Otherwise leave INRESD = 0.

Specify the type of map with KOUT, the derivative (if applicable) with KDER.

D. Carrying out a procedure with IHEAVYPROC. Heavy has the capability of carrying out an ordered sequence of refinements. These are useful if you want to carry out refinement in a semi-automatic fashion. When you specify a procedure with iheavyproc, you need to specify all the parameters that you want refined at all. Then the procedure you choose decides which parameters to refine on which cycles. Usually you will specify REFINEALL for all atoms, then let the procedures decide which to refine. If you use a procedure, the program will automatically fix all coordinates that cannot possibly be refined. For example, in space group C2 one atom in each derivative must have y fixed if origin-removed Patterson refinement is used, because the y-direction is polar. The program will fix the coordinate(s) of the atom that is the strongest in each derivative. If you have already fixed the coordinate(s) of an atom in a derivative (by not specifying that they be refined) then the program will just fix the atom you chose and not fix any others.

Note that you can carry out any series of refinements that you wan by setting up all your keywords for the first type of refinement, initiating refinement with the command HEAVY, then going back to KEYWORD mode, specifying the next type of refinement without changing or setting any other parameters unless you want to, then initiating the next refinement cycles with HEAVY, and so on. For example, you might type in all your heavy atom parameters, finishing with

```
...
NREP 5
IHEAVYPROC 2
!  now refine 5 cycles with iheavyproc=2
HEAVY


NREP 7
IHEAVYPROC 4
! now refine 7 cycles with iheavyproc=4
HEAVY
```

This sequence of commands results in 5 cycles of refinement of xyz of all atoms that you specified refinement of xyz, then 7 cycles of refinement of xyz,occ, and B of all atoms that you specified these parameters to be refined in. You can do this sort of thing in any order and ad infinitum if you wish.

Note that there is no procedure to refine just thermal factors. With this package there is no need to alternately refine occupancies and thermal factors. If there is insufficient data (i.e., very low resolution) to refine both occupancies and thermal factors, then set the thermal factors to any reasonable value and just refine the occupancies.

Changing heavy atom parameters after you have gone on to the next atom or derivative

If you want to change which parameters for which atoms are refined after you have already set up the atoms and refinement parameters, then you have to use a special way to reset them. The reason you have to do something special

is that if you say "DERIVATIVE" then the routine assumes you are inputting data for a new derivative, so you can't go back to a previous one with that command. Instead, you type:

```
GOTODERIV 2    ( to go to derivative #2)
GOTOATOM 3     (to atom #3 in deriv #2)
REFINENONE     (set all refinement flags back to zero)
REFINEXYZB     (or whatever you want to refine for this atom)

GOTOATOM 1    ( now do atom 1 in deriv 2)
GOTODERIV 1    (now do derivative 1)
```

Contents	Index

Local scaling and merging of data

[Script for localscale | Keywords for localscale]

[Merge | Keywords for Merge | More on Merge]

[Complete]

LOCALSCALE

LOCALSCALE is a routine to scale a "derivative" dataset to a "native" dataset using local scaling. In this method the scale factor for a particular reflection is based on the ratio of derivative:native for reflections surrounding this reflection. This method is useful because the scale factor is not restricted to any particular function of position in reciprocal space.

In this implementation, at least 30 reflections surrounding the reflection to be scaled are used to obtain a scale factor. Additionally, the reflections used in obtaining a scale factor are always chosen so that they form a complete sphere around the reflection of interest (inasmuch as possible). Initial Wilson scaling is carried out before local scaling.

Data files: The program expects to read in two data files: one for the native dataset and one for the derivative. The two files may in fact be the same if desired. The native dataset is expected to have h,k,l, F and sigma (at least). The derivative dataset is expected to have h,k,l, F, and sigma, and, if desired, del F ano and sigma of del F ano. The scale factor obtained for the derivative F is applied to all of the derivative data.

A dorgbn-style file is written out containing the scaled derivative data. If you wish to have the derivative and native data in the same file, then follow this with the routine "FILEMERGE" and merge the two files. The output data file is NOT mapped to the asymmetric unit. Ordinarily you will want to follow LOCALSCALE with MERGE to merge the symmetry-related reflections and map everything to the asymmetric unit. You may need to run MERGE on your native data as well, to map it to the asymmetric unit.

Sample script to localscale der.drg to nat.drg:

```
!---------------Script for localscaling of derivative F to native F -----
@solve.setup                  ! standard parameters for this dataset
infile nat.drg                ! native in infile
nnatf 1                       ! column for native F
nnats 2                       ! column for native sigma
infile(2) der.drg             ! derivative in infile(2)
nderf 1                       ! column for deriv F
nders 2                       ! column for deriv sigma
outfile der.scl               ! output file
```

```
localscale                    ! do local scaling
!-----------------------------------------------------------------------------
```

Keywords for LOCALSCALE

```
NSHELLS n          number of shells of resolution used to group data(default=10)
INFILE(1) xx       file with Native which is not further scaled
INFILE(2) xx       file with Derivative data to scale to Native
OUTFILE xx         output file with scaled derivative data
NNATF n            column # for F of native data
NNATS n            column # of sigma of F of native data
NDERF n            column # for F of deriv data
NDERS n            column # for sigma of F of deriv data
NANOF n            column # of anomalous difference (Fplus-Fminu) of deriv data
NANOS n            column # of sigma of anomalous difference
note: be sure to set those columns you don't want to 0

FILETITLE          optional title for output file

KEEPALL            keep reflections even with high differences
TOSSBAD            (default)Toss reflections if differences between native and
                   derivative are more than 3 * the rms found for other
                   reflections.
                   Note: KEEPALL and TOSSBAD apply to MERGE, LOCALSCALE,
                   SCALE_MAD, SCALE_MIR, SCALE_NATIVE. This is the
                   place to reject derivative reflections with very large del F
                   if you want to reject them at all.
ANCUT              minimum # of reflections to use to scale a reflection (30.)
RATMIN             minimum ratio of F/sigma to include (default=2)
NOBFACTOR          if specified, do not apply overall Wilson scaling before
doing
                   local scaling. Generally used only along with DAMPING=0.
BFACTOR            undoes NOBFACTOR. Do apply Wilson scaling before local scaling
DAMPING xx         scale factor (after Wilson scaling) is damped by taking it
                   to the power xx. Generally used with NOBFACTOR and a value of
                   0 to not do any scaling at all.
NODAMPING          undoes DAMPING by resetting damping factor to 1.0
OVERALLSCALE       just get 1 scale factor for the whole dataset. No local
                   scaling, no wilson scaling. Same as NOBFACTOR + DAMPING 0.0
NOOVERALLSCALE     undoes OVERALLSCALE. SAME AS BFACTOR + DAMPING 1.0
```

More on localscale:

1. A value of 0 or less for fnat or fder is assumed to mean data are not measured. A value of 0.0 or -1.0 for del f ano is assumed to mean the data are not measured also.

2. If sigmas are not supplied at all, then a value of 1.0000 will be assumed. This can affect what data are read in if you specify a minimum F/sig >0.0

3. If a particular (h,k,l) is found more than once, only the first is used. This is because localscale uses neighboring reflections to scale each (h,k,l) and if it is found more than once there is no way to know which observations are really its neighbors in both time and position.

.

MERGE

MERGE is a routine that merges measurements of structure factor amplitudes and rejects outliers. It summarizes the quality of the dataset in a listing of R-factors on I and on F.

Sample script file for MERGE

```
!-------------Script file for merging of native F from 2 data files------
@solve.setup              ! standard data for this dataset
nset 2                    ! number of input files to follow
infile(1) nat1.drg        ! input data file with F's unmerged
infile(2) nat2.drg        ! another input data file
ncolf_merge 1             ! get native F from column 1 in each data file
ncolsig_merge 2           ! get sigma from column 2
outfile native.mrg        ! output file with cols 1, 2 of "Favg" and sigma
merge
!-----------------------------------------------------------------------
```

The method followed by the program is:

1. group equivalent reflections together, analyze 1 group at a time.

2. get mean, sd for this group

3. reject observations differing from mean by >4 sigma

4. reject reflection outright if Chi-squared is greater than 20 and ikeepflag=0

5. calculate stats based on what's left

6. figure out the relationship between sigmas in the input files and reasonable estimates of the true sigmas by assuming that the reduced chi-square would equal 1.0 if the correct sigmas were present. The data are fit to the equation,

Sig**2(I)=Sig**2(Poisson)+(A*I)**2

and all sigmas are corrected with this factor.

6. write out mean, SEM for the reflection

Keywords for MERGE

```
NSHELLS n        number of shells of resolution used to group data (default=10)
NFILES n         # of input files (1 to 4)
INFILE(1) xx     input file 1
INFILE(2) xx     input file 2 (up to 4 files)
NCOLF_MERGE n    column number in input file for F (default = 1)
NCOLSIG_MERGE n  column number in input file for sigma of F (default =2)
KEEPALL          keep all reflections, regardless of merging chisqr
TOSSBAD          toss reflections with merging chisqr> 20 (default)
                 Note: KEEPALL and TOSSBAD also apply to LOCALSCALE
OUTFILE xx       output file with 2 columns (F,sig)
```

More on MERGE:

It is ASSUMED that columns 1,2 are your values of F and sigma. (If this is not true, you need to run FILEMERGE first to create such a file). If your data is I and sigma of I, then run MATH with I_TO_F to convert from I to F.

The input data files do not need to have data in any particular order or to have complete datasets.

The data are written out starting with minimum H,K,L and incrementing L fastest, then K, then H.

The routine reports the number of rejects as NNN + MMM where NNN = the number rejected as being too far from the mean for that reflection and MMM is the number of reflections rejected completely with chisqr > 20.

Estimating completeness of a dataset

COMPLETE

COMPLETE a routine to determine the completeness of a dataset. It maps input data to the asymmetric unit of the space group and calculates the percentage of data that is present.

Sample script file for COMPLETE:

```
!----------------Script to estimate completeness of a dataset ---------------
@solve.setup        !  standard information about this dataset
infile  data.drg    !  input dorgbn file with data to be examined
nnatf 1             !  column for F
nnats 2             !  column for sigma
ratmin 2.0          !  only use data with F/sigma > 2.0
complete            !  figure out completeness of this dataset
!-----------------------------------------------------------------------------
```

Contents	Index

Binary data formats and map formats used by SOLVE

DORGBN-style data files.

These files can all be viewed with VIEW and can be exported with EXPORT. ASCII files can be imported with IMPORT (see Importing and Exporting for more information on this

DORGBN files

The DORGBN-style files used in this package are binary files with data sorted by hkl. Format of the binary (FORTRAN unformatted) data file:

record 1 INTEGER*4 NCOL - the number of columns of data in the file.

record 2 (LOGICAL*1) TITLE(80) - An overall title

records 3... NCOL more titles, one for each column of data.

record 4 Data records - IH,IK,IL,RES,(F(I),I=1,NCOL)

1. IH,IK,IL - INTEGER*4 The indices of the reflection.

2. RES - The d-spacing in Angstroms.

3. F(I) - Data. These can be structure factors, sigmas, phase information stored as phase, figure-of-merit, etc. When data are missing for one or more columns the value -1.0 is stored in the appropriate columns.

```
MADFBARFILE xx.scl Output file with (Fbar,sigma,DelAno,sigma)
                    for each wavelength will be xx.scl
                    (DEFAULT="mad_fbar.scl")

MADFPFMFILE yy.scl Output file with (F+,sigma,F-,sigma) for each
                    wavelength will be yy.scl
                    (DEFAULT="mad_fpfm.scl")

madfbarfile xx.scl   dorgbn file with (Fbar,sig,Delano,Sig)
```

```
                        for each wavelength will be xx.scl
                        [Default="mad_fbar.scl"]
madfpfmfile xx.scl  dorgbn file with (F+,sig,F-,Sig)
                        for each wavelength will be xx.scl
                        [Default="mad_fpfm.scl"]

madmrgfile xxx.out    SIRAS-like MAD dataset .See MADMRG documentation.
[     default="madmrg.out"].
The file has 8 columns of data:

       1 madmrg est of Fp-zero ("Fnative")
       2 madmrg sig of fp-zero ("sig of Fnative")
       3 madmrg: MOCK FDER ("Fderiv"; equal to Fp-zero + del iso)
       4 madmrg sig of del iso ("Sig of Fderiv")
       5 madmrg est of del ano ("Delano")
       6 madmrg sig of del ano ("Sig of Delano")
       7 madmrg weighted est of del iso for Patterson
       8 madmrg weighted est of del ano for Patterson

madbstfile yyy.out  coefficients for a Bayesian Patterson to yyy.out
[default="madbst.out"]  See MADBST documentation.
The file has the following columns of data:

       1 madmrg est of Fp-zero ("Fnative")
       2 madmrg sig of fp-zero ("sig of Fnative")
       3 madmrg: MOCK FDER ("Fderiv"; equal to Fp-zero + del iso)
       4 madmrg sig of del iso ("Sig of Fderiv")
       5 madmrg est of del ano ("Delano")
       6 madmrg sig of del ano ("Sig of Delano")
       7 madmrg weighted est of del iso for Patterson
       8 madmrg weighted est of del ano for Patterson
         9--  = Fh component along Fo weighted by figure of merit
       10 --  = weighted Fh component perpendicular to Fo
       11 --  = best estimate of Fa
       12 -- sigma of
       13 -- sqrt() = sqrt of best estimate of Fa**2
       14 -- sigma of sqrt()


SOLVEDATAFILE  xxx  Output datafile with MADMRG output and MADBST
                    output  combined together, suitable for use
                    with routine SOLVE, will be xxx.   (DEFAULT file name =
                    "solve.data") The datafile has the following
                    columns of data:

                    1         Fnative (from MADMRG)
```

```
2          Sigma of Fnative (from MADMRG)
3          Fderiv (from MADMRG)
4          Sigma of Fderiv (from MADMRG)
5          DelAno (from MADMRG)
6          Sigma of DelAno (from MADMRG)
7          <Fh cos(theta)> (from MADBST)
8          <Fh sin(theta)> (from MADBST)
```

Map formats for SOLVE

The format of all maps in SOLVE is a simple binary file with the z-axis varying most rapidly and the x-axis the most slowly. They are all written with something like:

```
        do 100 iz=izs,ize
            do 100 iy=iys,iye
              write(2)(rho(ix,iy,iz),ix=ixs,ixe)
 100  continue
```

You can convert from this format of maps to NEWEZD maps (that can be read into O or mapman using FFTtoEZD (see the section on FFTtoEZD in Importing and Exporting.)

Contents	Index

Importing and Exporting data with SOLVE

IMPORT and EXPORT are utilities to bring formatted data into dorgbn format and to write out formatted data without titles. BTOF and FTOB write out and read formatted data with titles. Also DRGTOXPLOR converts from some dorgbn files to XPLOR-style files.

FFTTOEZD and FFTTOMAPVIEW convert from the binary format for maps to the formatted NEWEZD or binary MAPVIEW (PHASES) formats.

IMPORT

The routine IMPORT has several options. You can simply read the data from a formatted file in, assuming it is h,k, l, and columns of data. You can also swap indices (as H->K, K->L, L->K) as you read it in. You can also sort the data and map it to the asymmetric unit of the space group. Ordinarily you will want to sort and map the data, as some of the other routines in the package (notably FILEMERGE) assume that the data has been sorted in a particular order of hkl. When you sort the data, the program asks if any columns are to be interpreted in terms of phases in degrees. Such data has to be correctly adjusted when it is mapped to the asymmetric unit used by SOLVE. If the data has not yet been merged to the asymmetric unit and you are about to scale the data with localscale, then you may not want to sort and map the data now. If you choose not to sort it now, then you must sort and map it with MERGE after running localscale so that filemerge can recognize the data.

When you IMPORT data, it is essential that the input file has the same number of data columns for every hkl in the file. You have two options for the format of the data in the input file.

(1) You can specify that there is exactly ONE line for each HKL record. In this case, the input file can have text in the middle of a data column which will be ignored.

(2) Alternatively, you can specify the exact number of data columns for each HKL record, in which case they can be spread over any number of lines. Data "columns" refer to the assumption that for each HKL in the data file, there are a fixed number of "columns" of associated data.

That is, if your data file looks like:

```
 -3  -5  9   300.39 1.6 420.3 1.5
        265.9 5.6 991.2
         0.2
 -3  -5 10   225.35 2.5 413.1 3.4
            441.9 3.4 114.2 0.25
```

then the HKL are (-3, -5, 9) and (-3,-5,10). For the record with HKL=(-3,-5,9) the data columns are 300.39, 1.6,

420.3, etc... In this case you can use option (2), specifying that there are 8 data columns for each HKL. You can not use option (1) for this data file because the data for each hkl are not on a single line.

The commands for IMPORTing data are:

1. Input formatted data file name (the program will then type the first 3 lines of the file as read in, and then again after stripping off any text)

2. Option (1) or option (2)

3. Output dorgbn-style file name

4. Overall title for output file

5. Number of columns of data (not counting h,k,l) in input file

6a...Title for each of these columns of data

7. Overall scale factor to apply to all data

8. lsort: 'y' to sort and map data, 'n' to leave it as is

9. lswap: 'y' to swap indices hkl [only read if lswap='y']:

8a. HNEW: index H will be mapped to HNEW. That is, if you want to map old H->new K, old K->new L, old L ->new K, then you specify HNEW = "K"

10. [only if you have said 'y' to #8, lsort] # of columns of data to interpret as phases in degrees.

11. column #'s to be interpreted as phases in degrees. Note: for equivalent reflections the phase varies depending on the associated translation.

12. dmin, dmax = resolution range to consider. All data outside of this range is ignored. Note: don't set the dmin much lower than you really want it or the routine will be very slow.

Sample IMPORT scripts

```
!-------Import script: formatted datafile with 2 data columns on one line-----
IMPORT
infile.dat
1,        !  option 1 = one line per record
```

```
output.drg
Output data after infile.dat is converted to dorgbn-style file
2,        !  2 columns of data.
Fobs (title for column 1 of data)
Sigma (title for column 2 of data)
1.0,      ! overall scale factor = 1.0
y          ! Yes sort and map data
n         ! No, do not swap indices
0         ! don't interpret any columns as phases in degrees
3.5 50    ! resolution range to consider
!-----------------------------------------------------------------------------


!-------Import script: formatted datafile with 2 data columns on 2 lines -----
IMPORT
infile.dat
2,        !  option 2 = fixed # of data columns per record
output.drg
Output data after infile.dat is converted to dorgbn-style file
2,        !  2 columns of data.
Fobs (title for column 1 of data)
phase (title for column 2 of data)
1.0,      ! overall scale factor = 1.0
y          ! Yes sort and map data
n         ! No, do not swap indices
1         ! Interpret 1 column as phases in degrees
2         ! column 2 of data is to be interpreted as phases in
degrees
3.5 50    ! resolution range to consider
!-----------------------------------------------------------------------------
```

EXPORT

Export is a lot easier than IMPORT. All it does is write out a formatted file with h k l and the contents of each column of data. Here is a sample script to run export:

```
!----------------------Sample script to run EXPORT----------------------------
INFILE  data.drg         !    input file name
OUTFILE data.export      !    output file name
export
!-----------------------------------------------------------------------------
```

Binary TO Format conversion (BTOF)

Format TO Binary conversion (FTOB)

These routines convert data in DORGBN-structured binary files to and from formatted files with titles. They are most useful for transferring data files from one computer system to another. A sample script for doing each is shown next:

```
!------------Script to convert from dorgbn -> formatted file ----------------
INFILE data.drg
OUTFILE data.fmt
BTOF
!-----------------------------------------------------------------------------


!---Script to convert from formatted file with titles -> dorgbn file----------
INFILE data.fmt
OUTFILE data.drg
FTOB
!-----------------------------------------------------------------------------
```

DORGBN files

The files used in this package are binary files with data sorted by hkl. Format of the binary (FORTRAN unformatted) data file:

record 1 INTEGER*4 NCOL - the number of columns of data in the file.

record 2 (LOGICAL*1) TITLE(80) - An overall title

records 3... NCOL more titles, one for each column of data.

record 4 Data records - IH,IK,IL,RES,(F(I),I=1,NCOL)

1. IH,IK,IL - INTEGER*4 The indices of the reflection.

2. RES - The d-spacing in Angstroms.

3. F(I) - Data. These can be structure factors, sigmas, phase information stored as phase, figure-of-merit, etc. When

data are missing for one or more columns the value -1.0 is stored in the appropriate columns.

DRGTOXPLOR

The routine DRGTOXPLR writes out X-PLOR format file with INDIC, FOBS, SIGMA based on data in a DORGBN-style file. Here is a sample script:

```
!---------------------Script for DRGTOXPLOR ---------------------------------
INFILE  data.drg
OUTFILE  data.xplor
NXPLORF  1               ! column 1 of data.drg is F
NXPLORSIG 2              ! column 2 of data.drg is Sigma of F
DRGTOXPLOR
!-------------------------------------------------------------------------------
```

FFTtoEZD and FFTtoCCP4 and FFTtoMAPVIEW

FFTTOEZD

FFTTOEZD is a routine that converts an asymmetric unit of an FFT in the UCLA FFT format to any region of the map in the newezd format suitable for reading right into "O". This routine is applicable to Fourier maps, but can be used with Patterson maps as long as the output region is contained within the input FFT.

The output map is calculated using the same grid as the input FFT, but the endpoints in x,y, and z can be different. The grid points in x and y must not be more negative than -512 or greater than 512. The program generates the entire unit cell from the input FFT if the endpoints of the output EZD map are not contained within the input FFT. For Patterson maps, the output EZD map MUST be contained within the calculated FFT (i.e, no expansion is allowed).

The output map is scaled so that the rms value of the map is 1.0.

If you are in "O" and want to read in and display "fourier.ezd" all you need to type is:

```
@fourier.ezd
ezd_draw
<cr>
<cr>
```

Script file for converting from FFT to EZD

```
!-----------Script for conversion from UCLA FFT format to EZD ---------
@solve.setup                 !standard parameters for this dataset
fftfile fourier.fft          ! input FFT file
bossfile fourier.ezd          ! output EZD file
fourier                      ! this is a fourier, not a patterson
ezdgrid -35 37 -23 96 5 23  ! Output grid covers region from -35 to 37
                             ! in x (on the same grid that the fourier was
                             ! calculated with)
ffttoezd
!-----------------------------------------------------------------------
```

Keywords for FFTTOEZD

```
FFTFILE   xx      name of FFT-containing file
EZDMAPFILE   xx      name of output EZD format file
FILETITLE xxx      optional title for file
PATTERSON       this is a Patterson map
FOURIER         this is a fourier map
EZDGRID   ixstart ixend iystart iyend izstart izend -- these are
                 starting and ending grid units on the PATTGRID
                 or FFTGRID you have specified.
```

FFTTOCCP4

FFTTOCCP4is a routine that converts an asymmetric unit of an FFT in the UCLA FFT format to CCP4 map format. The region defined by FFTGRID is output

Keywords for FFTTOCCP4

```
FFTFILE   xx      name of FFT-containing file
CCP4MAPFILE   xx      name of output EZD format file
```

FFTTOMAPVIEW

FFTTOMAPVIEW is a routine that converts an asymmetric unit of an FFT in the UCLA FFT format to a format compatible with the PHASES package. This is useful for displaying the map using MAPVIEW in the PHASES package.

Note: this routine is only valid on an SGI machine.

The output map is calculated using the same grid as BOSS-style output maps. This grid may be set with the keyword BOSSGRID. The output map range of grid points must be contained, for x and y, between -512 and 512. For Pattersons, this grid must be contained within the FFT grid, set with FFTGRID

Script file for FFTTOMAPVIEW

```
!---------------Script for conversion of UCLA FFT to MAPVIEW format--------
!  Note:  this only works on an SGI
@solve.setup                 ! standard information for this dataset
fftfile fourier.fft          ! input fft file
mapviewfile fourier.mapview  ! output mapviewfile for PHASES
fourier                      ! this is a fourier, not a patterson
bossgrid -35 37 -23 96 5 23  ! Output grid covers region from -35 to 37
                             ! in x (on the same grid that the fourier was
                             ! calculated with)
ffttomapview
!-------------------------------------------------------------------------------
```

Keywords for FFTTOMAPVIEW

```
FFTFILE   xx      name of FFT-containing file
MAPVIEWFILE xx    name of output MAPVIEW format file
PATTERSON       this is a Patterson map
FOURIER         this is a fourier map
BOSSGRID  ixstart ixend iystart iyend izstart izend -- these are
                  starting and ending grid units on the PATTGRID
                  or FFTGRID you have specified.
```

Contents	Index

FILEMERGE

Merging dorgbn data files

[Script]

FILEMERGE allows manipulation of binary data files that are in the "dorgbn" format. You can extract one or more "columns" of data from a file, duplicate columns from a file, or combine parts of different files.

This routine is based on the UCLA program DORGBN. Data are stored in binary files in the form h,k,l,resolution,and columns of "data". At the beginning of the file are an overall title for the dataset and individual titles for each "data" column.

FILEMERGE runs interactively. The command format is:

1. # of Files to be opened for input (up to 4)

2a... Input file names 1...n.

3. Output file name

4. Title for output file. This title should describe the

contents of the entire file.

5. Command lines. Each command can specify a range of columns of data from some particular file to be incorporated into the output file. These data columns are incorporated in the order in which the commands are specified. Command input parameters: (these are 1 or 2 digit integers and an optional title, all separated by commas and no blanks):

1. NFILE: The number of the file treated.

2. ICOL,JCOL,TITLE: the RANGE of columns to be copied. If you want to copy just column #3, specify: 3,3 here.

The title is an optional title to be substituted for that already associated with the first column in the range. If the specified range includes more than one column and this title field is used, the titles for all the other columns in the range specified by this command must be input in the very next records. If this field is blank, the old title will be used.

PLEASE NOTE: if you are running filemerge from a command file, you cannot leave any blanks or other characters afer the column numbers here. That is,

3,3

is fine, but

3,3 !xxx

is not.

7. More command lines.

8. A "0" to signify the end of input.

An additional useful feature of the program is that if a command is entered with a valid file number but with the column numbers missing or incorrect, the titles of the columns on that file are printed for the user.

Script for FILEMERGE

Here is a sample script that will take columns 1,2 from "file1.drg" and columns 1 and 3 from "file2.drg" and write them out to "fileout.drg".

```
!----------------------Sample script for FILEMERGE -------------------------
FILEMERGE
2
file1.drg
file2.drg
fileout.drg
Title for fileout: cols 1,2 from file1 and cols 1,3 from file2
1
1,2
2
1,1
2
3,3
0
!-------------------------------------------------------------------------------
```

Contents	Index

Calculating and working with maps

[Keywords for MAPS | Script for MAPS]

[Peaksearch | Script for Peaksearch | Keywords for Peaksearch]

[MAPtoASYM |Rho | Omit_map | Avg_omit]

MAPS

```
This routine reads in data from a dorgbn-style file and calculates any of the
following types of maps:

PATTERSON                        !Fourier coefficients are F or F**2
ORIGIN_REMOVED PATTERSON         !As Patterson, but origin-removed
FOURIER                          !Fourier coefficients are A+iB
NATFOURIER/ISOFOURIER            !Fourier coefficients are mF exp[i Phi]
ANOFOURIER                       !Fourier coefficients are mF exp[i (Phi + 90)]
2FOFC                            !Fourier coefficients are (2Fo-Fc) exp[i Phi]
FOFC                             !Fourier coefficients are (Fo-Fc) exp[i Phi]
```

Note that:

In all cases a "0.0" or "-1.0" in a column is interpreted as missing data.

DELMAX can be used to cut off the biggest |F| or |DelF|

If NCOLFIGM is specified then the map is weighted by figure of merit

If NCOLPATTSIG is specified then Patterson maps are weighted.

If you specify a PEAKFILE then routine PEAKSEARCH is called automatically.

If you specify a MAPVIEWFILE then FFTTOMAPVIEW is called.

If you specify a BOSSFILE then FFTTOBOSS is called.

Keywords for MAPS

```
PATTERSON             Patterson map
```

```
ORIGIN_REMOVED_PATT  Origin-removed Patterson map
NCOLPATT**2          Column # in input map for squared Patterson coefficients
            -or-
NCOLPATT                column # containing Patterson coefficients
NCOLPATTSIG          Column # for uncertainty in values of ncolpatt. If you
                     specify ncolpattsig then you will get a weighted
                     patterson or difference patterson.

FOURIER         Fourier map, reading in A, B coefficients directly
NCOLFA                  column for A
NCOLFB                  column for B

NATFOURIER      same as isofourier, below
ISOFOURIER      F exp(iPHIc) map (F = del iso or F, phase = PHI)
ANOFOURIER      F exp(i[PHIc-90]) map (F = del Ano, phase = PHI - pi/2)
NCOLF           column for F or Del F
NCOLPHI         column for PHI (in degrees)
NFIGM           column for figure of merit if you want a weighted map
                  Note: if you set NFIGM for routine HEAVY it will be
                  applied here too.  Be sure you reset it to the value you
                  want.

FOFC            Fo-Fc exp(i PHIc) map
2FOFC           2Fo-Fc exp(i PHIc) map
NCOLFOBS        column for Fo
NCOLFC          column for Fc
NCOLPHI         column for PHI (in degrees)

DELMAX          maximum |F| or |del F| to include . If zero-use all
                          data.  DEFAULT=0

INFILE   xxx       name of input DORGBN file with data
FOURCOFILE xxx     name of optional output DORGBN file with A,B coefficients
                  (this can be used as input to another program for calculation
                  of maps, if desired, by converting it with EXPORT)

FFTFILE   xxx      name of output file with FFT in UCLA format
PEAKFILE  xxx      name of optional file with nlist high and low peaks
                  (program finds peaks if this file is defined)
BOSSFILE  xxx      name of optional file with FFT in BOSS format
                  (program writes bossfile if it is defined, see FFTTOBOSS)
MAPVIEWFILE xxx    name of optional file with FFT in MAPVIEW format
                    (program writes MAPVIEW file if it is defined,
                   see FFTTOMAPVIEW)


NLIST   n        number of high and low peaks to list to peakfile
ISYMMETRY   n     number of symmetry equivalents for each peak to list
```

```
                        (put a big number to get all within all adjacent
                        unit cells)
PDB             list peaks in PDB format
FRACT           list peaks in fractional format
```

Script using MAPS to generate a difference Patterson function:

```
!------------Script to calculate a difference Patterson ---------------
@solve.setup        ! standard information for this dataset
!  wtder.drg contains Fnat,sig,Fder,sig:

INFILE wtder.drg        ! input datafile with Fnat,sig, Fder, sig
OUTFILE iso.drg         ! intermediate file with Fder-Fnat
NNATF 1                 ! column for native F
NNATS 2                 ! column for sigma of native F
NDERF 3                 ! column for derivative F
NDERS 4                 ! column for sigma of derivative F
GETISO                  ! get isomorphous differences
INFILE iso.drg          ! now read in those isomorphous diffs again
NCOLPATT 1              ! use column 1 as del F for a patterson
PATTERSON               ! map type is patterson
FFTFILE ha.patt         ! output fft goes to ha.patt
MAPS                    ! calculate the map

[HASSP]                 ! analyze the patterson with HASSP, if desired
!-----------------------------------------------------------------------
```

PEAKSEARCH

PEAKSEARCH is a routine that finds high and low points in a Fourier map. It is not applicable to Patterson maps (use HASSP for that purpose). It assumes that the FFT has been calculated over the entire asymmetric unit and uses the symmetry file to map neighboring grid points on to the asymmetric unit. It reports the highest peaks in the map, with the height being the highest value of the FFT on a grid point and the coordinates being the centroid of the peak.

The routine can read in an FFT written by MAPS or it can be called at the end of routine MAPS. Note that it cannot read BOSS format data.

The routine will write out NPEAK highest and NPEAK lowest peaks to the output file. The "B-factor" in the PDB format file is the peak height/1000. The final column in the fractional-format file is the peak height/1000.

Script file for PEAKSEARCH

```
!-----------Script file to run peaksearch on a fourier----------------------
@solve.setup          ! standard information for this dataset
fftfile ha.fft        ! input map name
peakfile ha.peaks     ! write list of peaks to "ha.peaks"
peaksearch
!-----------------------------------------------------------------------------
```

Keywords for PEAKSEARCH

```
FFTFILE  xxx       name of fft-containing file
NLIST    n       number of peaks (high, same # of low) to list
ISYMMETRY  n     # of symmetry equivalents to list for each peak
                   -1 for all within FFTTOBOSS region, 0 for all within
                    FFT region. 1= default= just 1 for each peak
PEAKFILE   xx     name of output file
PDB             write peaks in PDB format
FRACT           write peaks in fractional format
POSITIVEONLY    only list positive peaks
NEGATIVEONLY    only list negative peaks
```

MAPTOASYM

MAPTOOBJECT

Routines MAPTOASYM and MAPTOOBJECT map the atoms in a PDB file, one by one, using crystallographic symmetry. MAPTOASYM maps atoms into the asymmetric unit of the space group, as defined by the symmetry file and the (FFTGRID) grid specified for FFT calculations, assumed to contain the asymmetric unit. MAPTOOBJECT maps atoms to their symmetry equivalents closest to an atom in a second PDB ("object") file.

Sample script files for MAPTOASYM and MAPTOOBJECT

```
!--------Script file to map atoms in a PDB file to the asymmetric unit ----
infile(1)  atoms.pdb      !  input file with coordinates to be mapped
outfile    atoms.mapped   !  atoms mapped to asymmetric unit
maptoasym                 !  map them to asymmetric unit
!---------------------------------------------------------------------------

!---Script file to map atoms in a PDB file close to atoms in another file ----
infile(1)  atoms.pdb      !  input file with coordinates to be mapped
infile(2)  object.pdb     !  pdb file with object that we want to be close to
outfile    atoms.mapped   !  atoms mapped close to object
dismin     0.0            !  Only atoms with a closest distance to an atom
dismax 1000.0             !  in object between dismin and dismax will be
                          !  written out. Default= 0 to 1000000.
maptoobject                 !  map them as close to the object as possible
!---------------------------------------------------------------------------
```

RHO

RHO is a routine to write out the values of a map at the coordinates of atoms in a PDB file. The values of the map at the grid point nearest the input coordinates are listed, coordinate-by-coordinate. This routine is very useful for evaluating how good the fit of a model to a map is.

Sample script file for RHO

```
!--------Script file to write out values of a map at coords in a PDB file ----
@solve.setup              !  standard setup for this dataset
infile(1)  atoms.pdb      !  input file with coordinates to be examined
fftfile fourier.fft       !  map file
rho                       !  interpolate value of fft at all coords in model
!-----------------------------------------------------------------------------
```

OMIT_MAP

OMIT_MAP is a routine to calculate a set of overlapping omit maps and then to average them together using AVG_OMIT. The routine requires a .PDB file with coordinates and a .DRG data file with Fobs values. A set of omit maps in which 7 residues at a time are deleted, overlapping by 2 (1-7, 6-11, etc) is calculated, and the atoms used to calculate each map are written out in a set of pdb files as well. Then AVG_OMIT is called to average the parts of the maps that are in the regions not occupied by atoms used in the phase calculations. If you are doing simulated annealing-omit maps, then you need to use routine "AVG_OMIT" instead.

Sample script file for OMIT_MAP

```
!--------Script file to calculate overlapping omit maps ------------------
@solve.setup              !  standard setup for this dataset
INFILE(1) atoms.pdb       !  Name of PDB file with coordinates
INFILE(2) data.drg        !  Name of .DRG data file
NNATF 1                   !  Column number for F in data file
NNATS 2                   !  Column number for sigma of F in data file
RATMIN 2.0                !  Minimum ratio of F/sigma to use (default=0.)
OUTFILE avg.fft           !  File name for file with output averaged FFT map
XCUT 2.0                  !  Density within XCUT of an atom is not used
                          !        default=2.0 (Angstroms)
OMIT_MAP
!-----------------------------------------------------------------------------
```

AVG_OMIT

AVG_OMIT is a routine to average key parts of a set of omit maps. For each map, only the region of the map that is further than DCUT from all atoms in a PDB file is used in the averaging. Omit maps or simulated annealing-omit maps can be used here.

The basic idea is that an omit map contains useful information only for parts of the map that are not occupied by the atoms used to phase that map.

To use this routine, calculate a series of omit maps for your structure that sequentially leave out residues 1-5, 4-9, 8-12, etc... (or whatever is a good size to leave out). Then you read in each omit map and the coordinates of atoms left in for that calculation. If you are using just one PDB file, then use the routine "OMIT_MAP" which will do everything for you. If you are doing a bunch of simulated annealing-omit maps, this is the routine to use.

NOTE: any points in the map that are not more than XCUT away from all atoms in at least 1 map will get a value of 0.0. This can lead to planes and other small regions in the map being zeroed out if all omit maps contain atoms near them. This is a problem in cases where two atoms distant in sequence are close together so that one or the other is always there in your omit maps. The solution is to construct a location-based omit map to delete all the atoms in that region.

Sample script file to run AVG_OMIT

```
!-------Script file to average overlapping omit maps ------------------------
@solve.setup            !  standard setup information

                        ! next call avg_omit.  Inputs are:
                        ! 1.  XCUT
                        ! 2.  NMAPS
                        ! 3a. file with coordinates of all atoms used to
                        !     calculate map 1
                        ! 3b. name of fft file with map 1
                        ! 4a.  as 3a, but coordinates for map 2
                        ! 4b.  as 3b, but map 2
                        ! output map file
avg_omit                ! call avg_omit.  It reads from standard input
2.0,                    !  XCUT = region around atoms to be excluded
2,                      ! # of maps to average
pdbfile1.pdb
mapfile1.fft
pdbfile2.pdb
mapfile2.fft
average.fft
! Now average.fft has in it the averaged parts of each input map that are not
!  near an atom in the corresponging pdb file
!-----------------------------------------------------------------------------
```

Contents	Index

Misc commands

HA_PDB: Write out a heavy atom solution in PDB format

COMPARE_SOLN: Comparing two heavy-atom input files

FRACT_TO_CART and CART_TO_FRACT: Converting between fractional and Cartesian coordinates

MATH: simple operations on a dataset

GETISO: Get isomorphous differences

GETANOM: Get anomalous differences

GETPHASES: Get phase and F from A, B

Weights: weighting for macromolecular refinement

Script for weights

Keywords for weights

HA_PDB: Write out a heavy atom solution in PDB format

Solve will write out your current heavy-atom model in PDB format if you simply specify the command HA_PDB. This can be done, for example at the very end of an automated run of SOLVE to output the final heavy-atom model in pdb format.

The output file will be called "ha.pdb".

COMPARE_SOLN: Comparing two heavy-atom input files

Compare_soln reads in heavy atom sites from 2 files and compares them. It tests all possibilities for inversion and origin shifts that could make the solutions indistinguishable. It reports back if the solutions are the same and how many sites are the same if not . The files need to be in the format used by HEAVY and SOLVE. All lines in the 2 files except those that say "DERIVATIVE" or "XYZ" or "INANO" are ignored. Each time "DERIVATIVE" is encountered, a new derivative is started. Each XYZ is read as a keyword for xyz of a new atom in the current derivative. If "INANO" is specified for a derivative then the handedness of the solution is considered in comparing the solutions. Otherwise mirror images are considered the same. In all cases solutions that simply differ by an origin shift are considered the same. The grid used for FFT calculations is used to determine how close atoms must be to be considered the same, and the cutoff is 1 to 2 grid units. If you have a PDB file with your coordinates for one derivative, you can substitute

PDB_XYZ_IN filename instead of the XYZ lines. You still need the DERIVATIVE statement.

For example:

If xyz1.dat looks like:

```
Derivative 1
xyz 0.45 0.93 .33

and xyz2.dat looks like

Derivative
xyz .51 .48 .93
derivative
xyz .44 .351 .31
```

then use a script like this:

```
!----------Script to compare to heavy-atom input files -------------------
infile(1) xyz1.dat
infile(2) xyz2.dat
compare_soln
!-----------------------------------------------------------------------
```

FRACT_TO_CART and CART_TO_FRACT: Converting between fractional and Cartesian coordinates

FRACT_TO_CART converts from fractional to Cartesian coordinates. CART_TO_FRACT converts from Cartesian to fractional coordinates. In each case, coordinates are read from the file defined by INFILE and written to the file defined by OUTFILE. Coordinates are read in free-format.

The Cartesian coordinate system is the PDB default: X is along **a** and Z is along **c***; X Y and Z are mutually perpendicular and right-handed.

Use a script like this:

```
!----------Script to convert from fractional to Cartesian coords----------
infile fract.xyz
outfile cart.xyz
fract_to_cart
!-----------------------------------------------------------------------
```

MATH

MATH is a routine that is useful for generating test data, for converting I,sig to F,sig, and other simple conversions. In most cases, the routine takes an input file and column numbers for the input data and writes out an output file.

KEYWORDS for routine MATH:

KEYWORD	parameters	description
GENF_PHI	SCALE B	Generates an asymmetric unit of data in this space group based on resolution limits set by DMIN DMAX. The rms F will be SCALE at low resolution and will decrease according to the thermal factor B. Output file has F, PHI. F and Phi are distributed according to Wilson statistics.
I_TO_F	ncolI ncolSIGI	Convert from I, sigma to F, sigma
SIGMA_SCALE	xx	scale all input intensity sigmas * xx
A_B_TO_F_PHI	ncolA ncolB	Convert from (REAL,IMAGINARY) to (F,PHI) (phi in degrees).
F_PHI_TO_A_B	ncolF ncolPHI	Convert from (F,PHI) to (REAL,IMAGINARY)
FOBS_SIG_FROM_F_ERR	ncolF ERR	Add "measurement error" on to values of F. % error in F will be about ERR. Uses error model of: sigma(I)=sqrt(I+0.5*(ERR/100)*I**2), where I=F**2.
SUM	ncol1 ncol2	Output column 1 is sum of data in ncol1 and ncol2 in input file
VECTORSUM	ncol1A ncol1B ncol2A ncol2B	Output column 1 = ncol1A+ncol2A Output column 2 = ncol1B+ncol2B
SEPARATEANO	ncolF ncolsig	Read in hkl, F,Sig from file defined by INFILE and write out the reflection to the file defined by FPLUSFILE if this is F+, or the file defined by FMINUSFILE if this is F-. A reflection is F+ in this definition if it is in the asymmetric unit as defined by HEAVY or it can be rotated into the asymmetric unit with reciprocal lattice symmetry. It is F- if it can be rotated onto -h-k-l of a reflection that is an F+.
FLIP		In separateano, write out the inverse of the input indices for F- (i.e, if h,k,l are read in, write out -h -k -l)

NEQUIV_SEPARATE n	In separateano, map the F- reflections to equivalent reflection "n" before applying FLIP and writing out. This allows you to match up reflections collected on a mirror plane as opposed to those collected with phi+180. If you run SCALE_MIR or SCALE_MAD the value of "n" is chosen for you automatically by maximizing the number of F+/F- pairs related by symmetry operation "n".
TRIM	Delete every line of a reflection file that has "-1" in any column
RESOLUTION xx yy	resolution limits If resolution limits and infile and outfile are specified, copies infile to outfile, using only data in the resolution range.
INFILE xxxx	input file name
OUTFILE xxxx	output file name (except separateano)
FPLUSFILE xxxxx	output file for F+ in separateano
FMINUSFILE xxxxx	output file for F- in separateano

Sample script for routine "MATH"

Here is a simple script that will convert from A, B fourier coefficients to F and phi:

```
!-------------------Script for conversion of A B to F, Phi --------------
@solve.setup               !  standard setup for this dataset
infile ab.drg
outfile fphi.drg
a_b_to_f_phi 1 2            ! take columns 1 and 2 of infile and convert to
                           !   F and phi
math
!-----------------------------------------------------------------------
```

GETISO

GETISO is a routine to subtract 2 columns of data in a dorgbn file and to write out a new file with the difference. Reflections with a "-1.0" or "0.0" in either column are ignored as are reflections with F/sigma < ratmin. Here is an example:

```
!-------------Script file for getting isomorphous differences ------
@solve.setup                        ! setup script file
nnatf 1                             ! column for Fnat
nnats 2                             ! column for sigma of Fnat
nderf 3                             ! column for Fder
nders 4                             ! column for sigma of Fder
ratmin 2.0                          ! minimum F/sig to include
infile file1.drg                    ! input file
outfile file2.drg                   ! output file
getiso                              ! get isomorphous differences
!-----------------------------------------------------------------------
```

GETANOM

GETANOM is a routine to convert from F+, F- to Fbar, DelAno. The routine calculates Fbar = (F+ + F-)/2 and del Ano = (F+ - F-), for the selected reflections and writes it to the output dorgbn file. If F+ or F- are missing (F less than or equal to 0), and the keyword "fp_or_fm" is specified, the one present is written out as Fbar and del Ano and sig of del Ano are set to 0.0. If the keyword "fpfm_only" is set then reflections with F+ or F- missing that are acentric are tossed. Here is an example script file for GETANOM:

```
!-------------Script file for converting from F+,F- to Fbar, Delano-------
@solve.setup
ncolfp             ! column for F+
ncolsfp            ! column for sigma of F+
ncolfm             ! column for F-
ncolsfm            ! column for sigma of F-
fpfm_only          ! toss acentric reflections if F+ or F- is missing
infile file1.drg   ! input file
outfile file2.drg  ! output file with 4 columns
getanom
!-----------------------------------------------------------------------
```

GETPHASES

GETPHASES is a routine to convert from A and B fourier coefficients to F and Phi. Here is a script file to do this:

```
!------------------Script file to convert from A, B to F, Phi -------------
@solve.setup
ncolfa 1           ! column for F cos(phi) = A
ncolfb 2           ! column for F sin(phi) = B
infile file1.drg   ! input file with A, B
outfile file2.drg  ! output file with F, Phi (2 columns)
getphases
!-----------------------------------------------------------------------------
```

Bayesian weighting for atomic refinement

WEIGHTS is a routine to generate weighting factors for atomic refinement. The weighting factors are based on both experimental sigmas and on rms values of (Fobs-Fcalc)**2 in ranges of resolution. The premise for this type of weighting is that the atomic model used to generate Fcalc is incomplete. Note that this leads to an expected difference between Fobs and Fcalc that is larger for centric reflections than for acentrics by a factor of 1.414. The errors in the fit of the model to the data are divided into two parts, one due to errors in measurement and one due to errors in the model. It is assumed that errors in measurement are reasonably well known. They are required for this routine. The input file must contain Fobs, sigma-obs, and Fcalc. It may also contain a flag marking "TEST" reflections for free-R calculations. To generate the input file, you will need to run X-PLOR or another program to get Fcalc values, then IMPORT the Fcalc values and FILEMERGE with your Fobs, sigma-obs data.

The output dataset ("Fobs, sigma, weight") is written in X-PLOR format and can be read in to X-PLOR just as if it were Fobs, sigma, weight. X-PLOR automatically uses the weight as a weighting factor in refinement if it is input in this way in the structure factor file. Note that the sigma here is NOT the experimental error in measurement any more.

You can select reflections with F>n*sigma-obs using RATMIN, but even if you include all reflections, a reasonable weighting factor will be generated for the weak reflections.

The program allows you to keep reflections flagged with RTEST>0 separate from the working set of reflections. The RTEST flag must be in a "column" of data in the input file. RTEST=1 indicates a TEST reflection, 0 indicates a reflection to use in refinement.

The errors in the model are estimated in a shell of resolution as

E**2 = [< (Fobs-Fcalc)**2 > - <Sigma-obs**2>]

where centric and acentric reflections are treated separately. The weighting factor applied to a particular reflection is then:

WEIGHT = 1/(E**2 + Sigma-obs**2)

Reflections where Sigma-obs is not >0 or Fobs is not > ratmin*sigma-obs or Fcalc is not >0 are ignored and not written out.

Script file for WEIGHTS

```
!----------Script file generating Bayesian weighting for refinement------------
@solve.setup              ! standard information for this dataset
infile fofc.drg           ! input dorgbn file with Fo, sigma, Fc, Rtest
outfile fo.xplor          ! output file with FOBS SIGMA RTEST for X-PLOR
NCOLFOWT  1               !  column # for F of data
NCOLSWT   2               !  column # of sigma-obs
NCOLFC    3               !column # of Fcalc
NCOLRTEST  4              ! column # for RTEST indicator (0 if not present)
weights                   ! get Bayesian weights...
!-------------------------------------------------------------------------
```

Keywords for WEIGHTS

```
NSHELLS n       number of shells of resolution used to group data (default=10)
INFILE xx       name of file with Fobs,sigma-obs, and Fcalc and optional Rtest
OUTFILE xx      name of output file in X-PLOR format with
                 h,k,l,fobs,sig,weight,rtest
NCOLFOWT n       column # for F of data
NCOLSWT n        column # of sigma-obs
NCOLFC n        column # of Fcalc
NCOLRTEST n     column # for RTEST indicator (0 if not present)
RATMIN xx       minimum ratio of F/sigma to read in at all (default=0)
```

Contents	Index

Bayesian difference refinement

[Script for Fdiff | Keywords for Fdiff]

FDIFF is a routine to generate a pseudo-mutant dataset for difference refinement. The output dataset ("Fdiff, sigma-fdiff, weight") is written in X-PLOR format and can be read in to X-PLOR just as if it were Fobs, sigma, weight. X-PLOR automatically uses the weight as a weighting factor in refinement if it is input in this way in the structure factor file.

Fdiff is used in cases where a "WT" structure has been refined and a "mutant" dataset is available, and where it is the differences between the WT and mutant structures that is of interest. This routine takes Fcalc for the WT and mutant datasets and Fobs for the WT and mutant datasets to create a pseudo-mutant dataset: Fdiff and sigma and weights. These are then used just as if they were Fobs,mutant and sigma and weighting factors in refinement of the mutant structure. You can select reflections with F>n*sigma using RATMIN, but even if you include all reflections, a reasonable weighting factor will be generated for the weak reflections.

The value of Fdiff is given by:

Fdiff = Fc (WT) +Beta* (Fobs, mutant - Fobs, WT)

where the factor Beta is essentially the correlation coefficient between (Fo-Fc),WT and (Fo-Fc),Mutant. Bayesian difference refinement is similar to difference refinement except for the factor Beta (which is 1.0 for difference refinement).

The input file must contain Fobs, sigma-obs, and Fcalc, for BOTH the WT and "MUT" (variant) datasets. It optionally may also contain a flag marking "TEST" reflections for free-R calculations. To generate the input file, you will need to run X-PLOR or another program to get Fcalc values, then IMPORT the Fcalc values and FILEMERGE with your Fobs, sigma-obs data.

The program allows you to keep reflections flagged with RTEST>0 separate from the working set of reflections. The RTEST flag must be in a "column" of data in the input file. RTEST=1 indicates a TEST reflection, 0 indicates a reflection to use in refinement.

Basic script file for FDIFF

```
! -------------Script file to generate data for difference refinement -------
```

```
@solve.setup                  ! standard data for this dataset
infile data.drg               ! input file with WT Fo, sig, Fc and mut Fo, sig
outfile fdiff.drg             ! output file with Fdiff, sigma,weight, RTEST
ncolfowt 1                    ! column # for WT Fobs
ncolswt  2                    ! column # for sigma of WT Fobs
ncolfc   3                    ! column # for WT Fcalc
ncolfomut 4                   ! column # for Mutant Fobs
ncolsfomut 5                  ! column # for sigma of mutant Fobs
ncolrtest 6                   ! column # for RTEST indicator
fdiff                         ! setup up difference refinement
!-------------------------------------------------------------------------------
```

A more advanced script file for FDIFF written by Joel Berendzen (joelb@lanl.gov) that works with CNS

The file bayesdiff.ksh contains a script that does Bayesian differencing on two CNS output files and writes the output to a third CNS input file.

Keywords for FDIFF

```
INFILE xx input file with WT Fc, Fo, sig and MUT Fo,sig, and optional
Rtest column
OUTFILE xx output file with FDIFF,sdiff,weight,rtest
NCOLFC n column # for WT Fc (WT Fcalc) in input file
NCOLFOWT n column # for WT Fo (WT Fobs) in input file
NCOLSWT n column # for sigma of WT Fo
NCOLFOMUT n column # for MUT Fo (MUT Fobs) in input file
NCOLSMUT n column # for sigma of MUT Fo
NCOLFCMUT n column # for MUT Fc
NCOLRTEST n column # for RTEST indicator (0 if missing)
RATMIN xx minimum ratio of F/sig to read in data (default=0)
```

Contents	Index

New features for versions 1.10-2.10 of SOLVE

- SOLVE now has easy scripts for SAD phasing.
- SOLVE has even better MAD phasing! SOLVE now re-refines scattering factors using the final heavy atom parameters, improving the final phases. In combination with version 1.04 or later of RESOLVE, the final maps are greatly improved over version 1.17.
- Must faster search for solutions: SOLVE now follows only the very best solution by default, greatly speeding up the search in most cases.
- You can start from MR or other input phases.
- You can read CCP4 unmerged intensities directly.
- You can specify a SOLVETMPDIR where SOLVE will write scratch files.
- You can tell SOLVE where some sites are and go on from there easily with ANALYZE_SOLVE and with ADDSOLVE
- There is now a SOLVE FAQS page.

- SOLVE versions starting with 1.16 fix a serious bug that was present in versions 1.10 to 1.15. This bug was that upon conversion from SOLVE's asymmetric unit to the CCP4 asymmetric unit (done in routine "SOLVE" for all data, whether mtz input or not) the phase of FA relative to Fp was not swapped when the asymmetric unit changed. This affects the difference Fouriers calculated by SOLVE for space groups where the CCP4 and SOLVE asymmetric units for reflections are not the same.
- SOLVE version 1.17 fixes some array dimensioning errors in versions up to 1.16 that can cause it to hang when finding very large numbers (50 or so) sites.
- Solve version 1.17 sets the default of "resolution_steps" to 1 (it was 3 in previous versions). This improves MIR results for polar space groups but could slow down Solve in some cases.
- Solve version 1.18 had a minor bug in the way it dealt with SAD phasing, always substituting in the default f-doubleprime values for the heavy atoms from the tables, and including non-zero atomic scattering factors. Version 1.19 allows the user to input f-doubleprime (with fprprv_mad) and sets the form factors other than f-doubleprime to zero. This creates the appropriate scattering factors for a pseudo-derivative that is just like a native but has anomalous differences. This pseudo-derivative is used for phasing with the observed anomalous differences.
- Solve version 1.18 had a dimension error preventing the use of more than 5 derivatives; this is fixed in 1.19.
- Solve version 2.00 fixes a bug in previous versions of combine_all_data where the number of wavelengths was not read in correctly for the second dataset.
- Solve version 2.00 has scattering factors for Br

- Solve version 2.00 fixes bug which prevented solving SAD datasets in space group P1.
- Solve version 2.00 restores compatibility with new d*trek format.
- Solve version 2.01 incorporates Sim weighting on the heavy-atom sites in SAD phasing (this has the effect of replacing the heavy-atom sites in the map, and for cases with a high proportion of anomalous scatterers such as phosphorous phasing of DNA, makes a much better map).
- Solve version 2.03 writes out fractional coordinates to solve.xyz and also the inverse to solve_inverse.xyz.
- Solve version 2.03 fixes a bug in SAD which resulted in no phases written out if refinement of scattering factors failed.
- resolve version 2.03 fixes a bug in the output which caused the HL coefficients from the input file to be copied without changes to the output file whenever resolve terminated without carrying out all requested cycles.
- SOLVE version 2.03 changes the default BMIN (minimum B-value) from 15 to 2.0
- RESOLVE version 2.03 includes iterative model-building and model rebuilding
- RESOLVE version 2.04 will automatically figure out the optimal solvent content and histograms for your structure
- RESOLVE version 2.05 will use local patterns of density to improve the map
- RESOLVE version 2.05 has increased the weighting on map-based information by a factor of 2 relative to earlier versions to match recalculated estimates of the number of degrees of freedom in the map (now 2 * n_refl; was previously n_refl).
- RESOLVE version 2.06 introduces a new resolve_autobuild script that will carry out pattern identification, fragment identification, and model-building, using the superquick_build option, to get a partially refined model from MAD/SAD/MIR data in just a few cycles.
- RESOLVE version 2.08 fixes a bug which caused RESOLVE to fail when NCS was found in the ha sites, but then later rejected due to lack of NCS in the map.
- The RESOLVE_BUILD script for version 2.08 uses the average of the top 5 structures built in the initial stages to start the rebuilding process.
- RESOLVE version 2.08 introduces scoring of electron density maps based on skew, correlation of local rms (as in SOLVE), correlation of map calculated with map-probability with original map, tertiary structure.
- RESOLVE version 2.09 introduces greatly improved automated ligand fitting
- RESOLVE version 2.09 fixes a problem with finding sites with MAD data in trigonal/hexagonal space groups with 2-wavelength data or 4-wavelength data. These searches will have generally failed with previous versions, while single-wavelength or 3-wavelength data from the same datasets would have succeeded. The phasing was fine for cases where the solution was found, but the bug would normally prevent finding the heavy-atom sites. There may be some datasets where version 2.09 can solve 2-wavelength data that could not be solved by previous versions. (This problem was due to accidentally swapping the sign of anomalous differences used in the MADMRG routine during the conversion to the ccp4 asymmetric unit a number of times equal to the number of wavelengths...thereby cancelling out the necessary swapping for even-numbers of wavelengths... The problem did not affect phasing, unless the "nobayes" flag was specified, because phasing is done with all wavelengths, not with the MADMRG data.
- RESOLVE version 2.09 fixes a problem in reading heavy-atom sites and problems with reading

non-protein atoms in PDB files. RESOLVE earlier versions would ignore the atom HG as a heavy-atom in the NCS calculation, because the routines assumed that it was "H" (which is ignored)ignored) This is fixed in version 2.09, which now allows the use of the element symbol specification in the PDB format to define the atom type uniquely. (If no atom type is specified, in SOLVE/RESOLVE the first character of the atom name is now used to define the atom type for ATOM records, and all characters of the atom name are used in HETATM records. This is all because otherwise it is difficult to tell the difference between C-alpha and calcium atoms without the element symbol specification, and many PDB files do not contain element symbol specifications).

- RESOLVE version 2.10 fixes a bug in which a list can be overrun in model-building when the version is just big enough to run. The symptom was the cryptic error message "Please increase size of n_ca_max in iter_frag_place".
- RESOLVE version 2.10 has a very powerful loop fitting algorithm that you can run using a script or by using the new PHENIX software.

Table of Contents	Alphabetical Index

Copyright notice and acknowledgements

Copyright Notice

This program was prepared by the Regents of the University of California at Los Alamos National Laboratory (the University) under Contract No. W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has certain rights in the program pursuant to the contract and the program should not be copied or distributed outside your organization. All rights in the program are reserved by the DOE and the University. Neither the U.S. Government nor the University makes any warranty, express or implied, or assumes any liability or responsibility for the use of this software.

Acknowledgements

I am deeply thankful for a collaboration with Joel Berendzen that has produced many of the Bayesian approaches used by SOLVE. Many thanks to Richard Fisher who wrote the FFT and file manipulation routines used here and to the many others who write freeware that I have used. Finally, I am indebted to the users of HEAVY versions 1 to 4.5 who have provided me with valuable feedback that has greatly improved this new program.

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
setenv SYMOP /usr/local/lib/solve/symop.lib
#
solve <<EOD > solve.log
!command file to read in raw MAD data, scale, analyze and solve it----
checksolve                  ! compare solution with known h.a. sites
@solve.setup                ! get our standard information read in
logfile mad.logfile         ! write out most information to this file.
                            ! summary info will be written to solve.prt
readformatted               ! alternatives are readdenzo, readtrek
premerged                   ! alternative is unmerged
read_intensities            ! alternative is read_amplitudes
refscattfactors             ! alternative is fixscattfactors


mad_atomname se             ! anomalously scattering atom is Se

lambda 1                    ! info on wavelength #1 follows
label Wavelength #  1       ! a label for this wavelength
rawmadfile test_wva.fmt     ! datafile with h k l Intensity sigma or
                            ! h k l I+ sigma+ I- sigma-
wavelength 0.9000           ! wavelength value
fprimv_mad  -1.6            ! f' value at this wavelength
fprprv_mad  3.4             ! f doubleprime value at this wavelength

! input refined h.a. coordinates (used only for comparison in "checksolve")
! offset by 1/2 in y to match gvp.pdb 2-27-01.
atomname se
 XYZ   0.4813319     0.4972169     9.4140753E-02
 XYZ   0.9731338     0.7875228     0.9446641

lambda 2
rawmadfile test_wvb.fmt
wavelength 0.9794
fprimv_mad  -8.5
fprprv_mad  4.8
```

```
lambda 3
rawmadfile test_wvc.fmt
wavelength 0.9797
fprimv_mad  -9.85
fprprv_mad  2.86
premerged
readformatted
nres 100                [approx # of residues in protein molecule]
nanomalous 2            [approx # of anomalously scattering atoms per protein]
SCALE_MAD               ! read in and localscale the data
ANALYZE_MAD             ! run MADMRG and MADBST and analyze all the Pattersons
SOLVE                   ! Solve the structure
EOD
#
# Now run Resolve to do density modification
# (You can download it from http://resolve.lanl.gov
# if you do not have it yet)
#
resolve << EOD > resolve.log
!solvent_content 0.40       !    solvent fraction
seq_file gvp.seq
compare_file coords.pdb
EOD
#
#  That's it! Now resolve.mtz has your updated phases
#
echo 'Here are your SOLVE and resolve files:'
#
ls -l solve.prt solve.mtz solve.ezd resolve.mtz
#
echo 'All done.'
```

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*****************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*****************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 6)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: SOLVE 06-Apr-05

Space group number is: 5
Space group name from file name is: c2
Rescaling standard dataset to put it on approximate absolute scale. NRES =
100; <F**2> expected = 98000.00
; <F**2> observed in lowest resolution shell = 107053.9
... Scale factor = 0.9154266

*** Analysis of this scaled MAD data set ***

Fbar,sigma,Delano,sigma for 3 wavelengths written to:
mad_fbar.scl

F+,sigma,F-,sigma for 3 wavelengths written to:
mad_fpfm.scl

** Completeness of Fbar data at each wavelength: **

Completeness of dataset 1 (F > 2.000000 * sigma)
Wavelength # 1 ! a label for this wavelength

Reflections observed:

shell	dmin	Possible	Found	% complete
1	5.200	364	360	98.9
2	3.900	486	484	99.6
3	3.640	189	188	99.5
4	3.445	173	172	99.4
5	3.250	237	237	100.0
6	3.120	184	183	99.5
7	2.990	217	216	99.5
8	2.860	256	252	98.4
9	2.730	312	308	98.7
10	2.600	362	267	73.8
total		2780	2667	95.9

Completeness of dataset 2 (F > 2.000000 * sigma)
set 2

Reflections observed:

shell	dmin	Possible	Found	% complete
1	5.200	364	357	98.1
2	3.900	486	482	99.2
3	3.640	189	188	99.5
4	3.445	173	172	99.4
5	3.250	237	237	100.0
6	3.120	184	183	99.5
7	2.990	217	210	96.8
8	2.860	256	248	96.9
9	2.730	312	288	92.3
10	2.600	362	228	63.0
total		2780	2593	93.3

Completeness of dataset 3 (F > 2.000000 * sigma)
set 3

Reflections observed:

shell	dmin	Possible	Found	% complete
1	5.200	364	355	97.5
2	3.900	486	481	99.0
3	3.640	189	188	99.5
4	3.445	173	170	98.3
5	3.250	237	237	100.0
6	3.120	184	183	99.5
7	2.990	217	211	97.2

8	2.860	256	247	96.5
9	2.730	312	286	91.7
10	2.600	362	225	62.2
total		2780	2583	92.9

** R-factors for F-bar data dispersive differences **

Dispersive differences lambda 2 - lambda 1
(Delta f-prime = 6.900000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.200	354	250.947	0.037	1.000	11.11	3.71	2.99
2	3.900	478	289.474	0.021	1.000	6.64	3.64	1.82
3	3.640	187	256.106	0.018	1.000	4.44	4.06	1.09
4	3.445	172	212.002	0.022	1.000	4.11	4.03	1.02
5	3.250	235	205.104	0.021	1.000	3.29	3.95	0.83
6	3.120	181	188.273	0.025	1.000	4.81	3.68	1.31
7	2.990	210	160.278	0.027	1.000	3.72	4.19	0.89
8	2.860	247	144.703	0.028	0.999	3.20	3.87	0.83
9	2.730	287	127.549	0.035	1.000	3.22	4.59	0.70
10	2.600	226	117.926	0.038	0.999	2.87	4.74	0.60
Total:		2577	204.305	0.026	1.000	5.86	4.03	1.35

Recommended resolution cut-off = 2.60

Dispersive differences lambda 3 - lambda 1
(Delta f-prime = 8.250000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.200	351	251.323	0.039	0.999	11.66	3.72	3.13
2	3.900	477	289.185	0.023	1.000	7.44	3.57	2.09
3	3.640	187	255.567	0.021	1.000	5.46	4.03	1.35
4	3.445	169	212.785	0.023	1.000	4.55	3.93	1.16
5	3.250	236	204.307	0.021	1.000	3.94	3.80	1.04
6	3.120	183	188.150	0.029	1.000	5.92	3.64	1.63
7	2.990	210	157.911	0.031	1.000	4.42	4.07	1.08
8	2.860	245	145.425	0.031	0.999	4.27	3.63	1.18
9	2.730	285	127.739	0.036	0.999	3.18	4.62	0.69
10	2.600	223	118.708	0.036	0.999	3.12	4.26	0.73
Total:		2566	204.296	0.028	1.000	6.45	3.91	1.54

Recommended resolution cut-off = 2.60

Dispersive differences lambda 3 - lambda 2
(Delta f-prime = 1.350000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.200	353	254.151	0.010	1.000	1.75	2.79	0.63
2	3.900	471	287.779	0.008	1.000	1.13	2.64	0.43
3	3.640	183	253.214	0.009	1.000	0.56	2.88	0.20
4	3.445	167	213.508	0.012	1.000	1.61	2.87	0.56
5	3.250	234	205.151	0.011	1.000	0.00	2.94	0.00
6	3.120	181	189.134	0.012	1.000	1.50	2.71	0.55
7	2.990	204	160.405	0.017	1.000	1.97	3.02	0.65
8	2.860	243	144.802	0.017	1.000	1.53	2.86	0.53
9	2.730	280	128.364	0.024	1.000	1.78	3.49	0.51
10	2.600	219	119.348	0.023	1.000	0.00	3.57	0.00
Total:		2535	204.923	0.012	1.000	1.34	2.97	0.42

Recommended resolution cut-off = 2.68

Anomalous differences lambda 1 (f" = 3.400000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.200	356	249.568	0.021	1.000	5.86	4.56	1.29
2	3.900	476	286.584	0.016	1.000	4.25	4.24	1.00
3	3.640	184	257.083	0.016	1.000	2.07	4.83	0.43
4	3.445	172	212.104	0.021	1.000	3.33	4.71	0.71
5	3.250	235	206.109	0.017	1.000	1.85	4.24	0.44
6	3.120	181	187.386	0.022	0.999	3.58	3.98	0.90
7	2.990	210	154.747	0.026	0.999	2.18	4.74	0.46
8	2.860	250	142.195	0.030	0.999	3.39	4.38	0.77
9	2.730	301	125.378	0.033	0.999	0.65	5.34	0.12
10	2.600	256	116.327	0.037	1.000	0.00	5.69	0.00
Total:		2621	201.054	0.022	1.000	3.42	4.68	0.66

Recommended resolution cut-off = 2.67

Anomalous differences lambda 2 (f" = 4.800000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.200	356	252.146	0.042	0.999	13.71	5.54	2.47
2	3.900	476	288.261	0.030	1.000	9.81	5.46	1.80
3	3.640	184	255.315	0.027	1.000	6.91	5.76	1.20
4	3.445	167	212.632	0.032	1.000	6.32	5.87	1.08
5	3.250	232	204.717	0.031	1.000	6.37	5.53	1.15
6	3.120	177	188.925	0.032	1.000	5.70	5.27	1.08
7	2.990	203	157.884	0.037	0.999	3.97	6.24	0.64
8	2.860	244	143.025	0.040	0.999	4.35	6.08	0.71
9	2.730	283	127.971	0.048	0.999	3.07	7.11	0.43
10	2.600	220	119.988	0.049	1.000	0.00	7.16	0.00
Total:		2542	204.547	0.035	1.000	7.78	6.00	1.19

Recommended resolution cut-off = 2.60

Anomalous differences lambda 3 (f" = 2.860000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.200	351	253.965	0.014	1.000	1.50	4.54	0.33
2	3.900	473	287.310	0.012	1.000	1.18	4.33	0.27
3	3.640	184	253.025	0.014	1.000	0.00	4.88	0.00
4	3.445	168	210.803	0.017	0.999	0.00	4.93	0.00
5	3.250	228	204.891	0.015	0.999	0.00	4.42	0.00
6	3.120	178	191.177	0.017	1.000	0.00	4.24	0.00
7	2.990	205	158.395	0.024	0.999	0.00	4.98	0.00
8	2.860	242	143.961	0.026	0.999	0.97	4.67	0.21
9	2.730	277	127.379	0.030	0.999	0.00	5.80	0.00
10	2.600	220	119.936	0.035	1.000	0.00	5.52	0.00
Total:		2526	204.567	0.017	1.000	0.00	4.82	0.12

Recommended resolution cut-off = 3.90

ANALYZE_MAD: Run MADMRG and MADBST on MAD data to get ready for SOLVE

Correlation of anomalous differences at different wavelengths.
(You should probably cut your data off at the resolution where
this drops below about 0.3. A good dataset has correlation
between peak and remote of at least 0.7 overall. Data with

correlations below about 0.5 probably are not contributing much.)

DMIN	CORRELATION FOR WAVELENGTH PAIRS 1 VS 2	1 VS 3	2 VS 3
5.20	0.90	0.71	0.80
3.90	0.76	0.55	0.68
3.64	0.69	0.44	0.69
3.44	0.64	0.44	0.50
3.25	0.59	0.27	0.50
3.12	0.57	0.34	0.47
2.99	0.58	0.18	0.39
2.86	0.42	0.30	0.46
2.73	0.33	0.17	0.31
2.60	0.19	0.09	0.32
ALL	0.66	0.39	0.55

Final refined values of fprime and fdoubleprime

Form factors at lambda = 0.9000
f-prime = -2.58
f" = 3.71

Form factors at lambda = 0.9794
f-prime = -8.26
f" = 6.52

Form factors at lambda = 0.9797
f-prime = -9.01
f" = 2.26

Fa Patterson from MADBST to be written to:
patterson.patt

Script file suitable for running SOLVE written to:
solve_mad.script

Datafile for SOLVE with MADMRG-compressed dataset
("Fnat",sig,"Fder",sig,"Delano",sig,iso diffs, ano diffs,
<Fhcos>, <Fhsin> from MADBST) is:
solve.data

**** SOLVE: Solutions to MIR or SIR datasets ******

Derivatives considered: 3 (NSET)

```
Cross-vectors tested in HASSP:      20  (ICRMAX, DEFAULT=20)
HASSP solutions saved per deriv:    30  (NTOPHASSP, DEFAULT=30)
Fourier peaks saved per map:        30  (NTOPFOUR, DEFAULT=10)
Sites per derivative:               2  (NSOLSITE, DEFAULT=20)
Derivative solutions per seed:      5  (NTOPDERIV, DEFAULT=5)
Seeds per derivative tested:        10  (NSEEDTEST,DEFAULT=10)
Sorted seeds to use                 5  (NSEEDSOLVE, DEFAULT=5)
Number of final solutions saved:    5  (NTOPSOLVE, DEFAULT=5)


Solutions obtained will be compared to input solution (ICHECKSOLVE)
Correlated phasing used (CORRELPHASE)
Patterson map for derivative           2 will be read directly from:
patterson.patt

For derivative           2 the heavy atom
 structure factor components parallel to and
perpendicular to the native structure factor will be read from columns
         9 and          10

Datafile with          10 columns of data:
Title:MADMRG output (cols 1 to  8) and  MADBST fh cos,sin theta (c
Data: madmrg: MOCK FNAT
Data: madmrg: MOCK sig FNAT
Data: madmrg: MOCK FDER
Data: madmrg: MOCK sig FDER
Data: madmrg: MOCK DEL ANO
Data: madmrg: MOCK sig DEL ANO
Data: madmrg: Del iso for Patterson
Data: madmrg: Sigma of del iso for Patterson
Data: <Fa cos theta> = Fa component along Fo weighted by fom
Data: <Fa sin theta> = weighted Fa component perpendicular to Fo

Fnat,sigma taken from columns          1          2
Fder,sig,Delano,sig deriv           2 from cols:           3          4
         5          6

Check solution to be compared to all solutions found:

Derivative           1:
 Site    X       Y      Z

Derivative           2:
 Site    X       Y      Z
   1   0.481   0.497   0.094
   2   0.973   0.788   0.945

Derivative           3:
 Site    X       Y      Z

*********************************************************

ANALYZE_SOLVE:  analysis of top          1 solutions
```

****

Solution 1

********** Analysis of this solution **********

****** Analysis of non-randomness of native Fourier map ******

A. Maps with distinct solvent regions havea high standard deviation
of local r.m.s. electron density.
For this map the SD of this local r.m.s. is 0.2883532

B. Maps with distinct solvent regions also have a high correlation
of local r.m.s. electron density with
density at neighboring locations. Typical
values for poor maps in this
structure solution are 0.2778453 +/- 5.9644025E-02
For this map the correlation of r.m.s. density
in neighboring boxes is 0.3177819
The correlation coefficient is used here in scoring.

Skew of the map is: 0.2887446

****** Analysis of derivative solutions with the difference Patterson ******
and with cross-validation difference Fouriers

Derivative # 2

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.484	0.500	0.097	87.322
2	0.526	0.278	0.056	38.273

Evaluation of this test soln with 2 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.969	0.000	-0.194	14896.8	15250.4	2

Cross-vectors for sites 2 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.042	-0.222	-0.042	4232.84	3342.12	1
2	-1.010	-0.222	-0.153	5438.59	3342.12	1

Cross-vectors for sites 2 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.052	0.000	-0.111	1223.88	2929.69	2

Overall quality of this Patterson soln = 7488.57
Overall quality of the fit to patterson = 1.28245
Avg normalized peak height = 3348.99

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.482	0.497	0.094	0.758	55.365	4.49
2	0.528	0.283	0.057	0.458	60.000	4.07

Summary of scoring for this solution:

	-- over many solutions--		-- this solution --	
Criteria	MEAN	SD	VALUE	Z-SCORE
Pattersons:	2.09	1.95	5.58	1.79
Cross-validation Fourier:	3.72	2.33	7.13	1.46
NatFourier CCx100:	27.8	5.96	31.8	0.670
Mean figure of meritx100:	0.000E+00	10.2	59.7	5.86
Correction for Z-scores:				-2.36

Overall Z-score value: 7.42

****** Overall analysis of phasing for solution 1**********

*** Re-estimation of scattering factors by refinement of occupancies using ***
dispersive and anomalous differences.

Estimation of scattering factors at each wavelength
by refinement of occupancies relative to those found from the
initial refinement carried out with data from MADMRG.

Refining iso occupancies for iso diffs lambda 2 - lambda 1

Results of refinement:
Ratio of occupancies to standard refinement: 1.055 +/- 0.221
Delta f-prime based on input f-prime values: 5.684
New estimate of delta f-prime: 5.995 +/- 1.257
with sign of: -1. and Z of 33.0

Refining iso occupancies for iso diffs lambda 3 - lambda 1

Results of refinement:

Ratio of occupancies to standard refinement: 1.063 +/- 0.189
Delta f-prime based on input f-prime values: 6.427
New estimate of delta f-prime: 6.832 +/- 1.216
 with sign of: -1. and Z of 33.6

Refining iso occupancies for iso diffs lambda 3 - lambda 2

Results of refinement:
Ratio of occupancies to standard refinement: 2.606 +/- 0.945
Delta f-prime based on input f-prime values: 0.743
New estimate of delta f-prime: 1.935 +/- 0.702
 with sign of: -1. and Z of 9.8

Refining ano occupancies for ano diffs lambda 1

Results of refinement:
Ratio of occupancies to standard refinement: 0.609 +/- 0.300
f" value based on input values: 3.712
New estimate of f": 2.261 +/- 1.112

Refining ano occupancies for ano diffs lambda 2

Results of refinement:
Ratio of occupancies to standard refinement: 0.703 +/- 0.329
f" value based on input values: 6.525
New estimate of f": 4.584 +/- 2.147

Refining ano occupancies for ano diffs lambda 3

Results of refinement:
Ratio of occupancies to standard refinement: 0.618 +/- 0.301
f" value based on input values: 2.259
New estimate of f": 1.396 +/- 0.681

Fitting f-prime values.

Restraints:

Lambda	Target f-prime	final f-prime	weight
1	-2.579	-2.579	0.001
2	-8.263	-8.263	0.001
3	-9.006	-9.006	0.001

Delta-fprime targets:

Lambda i j target delta-fprime final delta-fprime wgt

```
  1  2          5.995      5.684      32.97
  1  3          6.832      6.427      33.62
  2  3          1.935      0.743       9.78

Residual for restraints:    0.00000E+00
Residual for targets:     4.7541



            Final refined values of f-prime and f"

  Wavelength  ------- f-prime --------      --------f"--------------
     last refinement      Refined     last refinement      Refined

   1        -2.579       -2.579          3.712        2.261

   2        -8.263       -8.263          6.525        4.584

   3        -9.006       -9.006          2.259        1.396


*** Done with re-estimation of scattering factors ***



HEAVY:  Refine heavy atom parameters

File title: SOLVE     06-Apr-05
CRYSTALLOGRAPHIC PARAMETERS

   A =  76.08     B =  27.97     C =  42.36     alpha =  90.00     beta = 103.20     gamma =  90.00

PHASES CALCULATED EVERY   5 DEGREES
RESIDUALS CALCULATED ON EXTRA ZEROTH CYCLE ONLY
SIGMAS FROM data FILE WILL BE USED
STATISTICS WILL BE PRINTED ON ZEROTH CYCLE, SHIFTS ON LAST
PHASING WILL BE DONE TAKING INTO ACCOUNT THE CORRELATIONS AMONG DERIVATIVES
THE GROUPS OF DERIVATIVES WITH CORRELATIONS WILL BE UPDATED
THE BETA VALUES FOR EACH DERIV  WILL BE SET TO 1.0
PHASE-AVERAGED RESIDUALS WILL BE USED FOR PHASING

TYPE OF REFINEMENT SELECTED:
UNPHASED ORIGIN-REMOVED  PATTERSON REFINEMENT ONLY

Bayesian correlated MAD phasing will be used with wavelength          2
 as the reference wavelength.

RESOLUTION LIMITS IN ANGSTROMS:     2.603    20.000
MINIMUM RATIO OF FDER TO RMS LACK-OF-CLOSURE FOR  INCLUSION IN REFINEMENT OR PHASING=     0.000
MINIMUM NATIVE F:     0.000
MINIMUM FIGURE OF MERIT FOR PHASED REFINEMENT:     0.000
MINIMUM ALLOWED ISOTROPIC B:     0.000
PARAMETER SHIFTS GREATER THAN     0.0000 TIMES SIGMA  WILL BE SCALED BY     0.5000
```

```
MINIMUM RATIO OF FNAT/SIGMA OR FDER/SIGMA TO INCLUDE:      1.000

NUMBER OF REFINEMENT CYCLES IS      2
DERIVATIVES REFINED DURING THESE CYCLES ARE :  0  0
TYPE OF OUTPUT SELECTED IS: +10 COLUMNS OF HENDRICKSON-LATTMAN COEFFICIENTS
1 INPUT data FILE WITH  12 COLUMNS IS:
mad_fpfm.scl
COLUMN  0 : mad_fpfm.scl Fnat,sig,(F+,sig,F-,sig)n
COLUMN  1 : F from I_TO_F      Wavelength #  1        ! a label for this wavelengt
COLUMN  2 : SIGMA of F         Wavelength #  1        ! a label for this wavelengt
COLUMN  3 : F from I_TO_F      Wavelength #  1        ! a label for this wavelengt
COLUMN  4 : SIGMA of F         Wavelength #  1        ! a label for this wavelengt
COLUMN  5 : F from I_TO_F      set 2
COLUMN  6 : SIGMA of F         set 2
COLUMN  7 : F from I_TO_F      set 2
COLUMN  8 : SIGMA of F         set 2
COLUMN  9 : F from I_TO_F      set 3
COLUMN 10 : SIGMA of F         set 3
COLUMN 11 : F from I_TO_F      set 3
COLUMN 12 : SIGMA of F         set 3


DERIVATIVE INFORMATION FOR  3 COMPOUNDS


COMPOUND 1     TEST REFINEMENT LAMBDA 3  (ANO ONLY)
COLUMNS FOR F+, SIGMA, F-, SIGMA     1    2    3    4
THIS DERIVATIVE WILL NOT BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
ONLY ANO DIFFERENCES WILL BE USED IN REFINEMENT AND PHASING FOR THIS DERIVATIVE.
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
NO PARAMETERS REFINED FOR ATOM LAM1 WITH ZERO OCCUPANCY

COMPOUND 2     set 2
COLUMNS FOR F+, SIGMA, F-, SIGMA     5    6    7    8
THIS DERIVATIVE WILL BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 3     set 3
COLUMNS FOR F+, SIGMA, F-, SIGMA     9   10   11   12
THIS DERIVATIVE WILL NOT BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
NO PARAMETERS REFINED FOR ATOM LAM3 WITH ZERO OCCUPANCY
CARRYING OUT STANDARD REFINEMENT
Total of           2 cycles will be done
Derivs refined will be           0          0
```

SUMMARY OF RESULTS ON FINAL CYCLE:

NUMBER OF REFLECTIONS READ = 2668
NUMBER OF F .GT. FMIN = 2668
NUMBER OF F IN RES. LIMITS = 2668
NUMBER OF F .GT. MIN <M> = 2634
NUMBER OF F USED TO REFINE = 0

FIGURE OF MERIT <	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
# OF REFLECTIONS	322	291	317	255	238	204	228	242	295	275

FIGURE OF MERIT WITH RESOLUTION

DMIN:	TOTAL	8.82	5.76	4.56	3.89	3.45	3.13	2.88	2.69
N:	2667	153	233	299	338	389	417	449	389
MEAN FIG MERIT:	0.48	0.75	0.69	0.68	0.61	0.48	0.45	0.31	0.24

RMS ANOMALOUS FH/E [f" PART OF FH / RMS ANO ERROR]:

LAMBDA: 1	0.5	0.9	0.9	0.7	0.7	0.5	0.3	0.2	0.1
LAMBDA: 2	0.7	1.0	0.9	0.8	0.8	0.6	0.5	0.4	0.3
LAMBDA: 3	0.3	0.6	0.7	0.5	0.3	0.3	0.2	0.1	0.1

RMS DISPERSIVE FH/E [Delta-f-prime PART OF FH / RMS DISPERSIVE ERROR]:

L1 VS L2:	0.8	1.1	1.2	1.0	0.9	0.7	0.5	0.4	0.3
L1 VS L3:	0.9	1.3	1.3	1.0	0.9	0.8	0.5	0.4	0.3
L2 VS L3:	0.2	0.3	0.3	0.2	0.2	0.1	0.1	0.1	0.0

RMS ANOMALOUS FH [f" PART OF FH] AS % of F:

LAMBDA: 1	0.9	1.3	1.4	0.8	0.8	0.7	0.7	0.7	0.6
LAMBDA: 2	1.8	2.7	2.9	1.6	1.6	1.4	1.4	1.4	1.3
LAMBDA: 3	0.6	0.8	0.9	0.5	0.5	0.4	0.4	0.4	0.4

RMS DISPERSIVE FH [Delta-f-prime PART OF FH] AS % of F:

L1 VS L2:	2.3	3.3	3.6	2.1	1.9	1.7	1.7	1.7	1.6
L1 VS L3:	2.6	3.7	4.0	2.3	2.2	2.0	1.9	1.9	1.8
L2 VS L3:	0.3	0.4	0.5	0.3	0.3	0.2	0.2	0.2	0.2

RMS ANOMALOUS ERRORS [CALC - OBS VALUE OF (F+ - F-)/2], IN % OF RMS F:

LAMBDA: 1	1.8	1.5	1.6	1.2	1.2	1.5	2.0	2.9	4.3
LAMBDA: 2	2.7	2.8	3.2	2.1	2.1	2.4	2.8	3.7	4.1
LAMBDA: 3	1.7	1.3	1.3	1.0	1.4	1.6	2.0	2.9	3.9

RMS DISPERSIVE ERRORS [CALC - OBS VALUE OF (F(i) - F(j))], IN % OF RMS F:

L1 VS L2:	3.0	3.0	3.0	2.1	2.2	2.5	3.4	4.4	5.9
L1 VS L3:	3.0	2.9	3.0	2.3	2.4	2.6	3.6	4.6	5.7
L2 VS L3:	1.9	1.5	1.4	1.2	1.4	1.7	2.4	3.2	4.7

CORRELATED ANOMALOUS ERRORS BY WAVELENGTH (%):

```
LAMBDA:  1       1.1   1.1   1.3   0.9   0.9   1.1   1.2   1.7   1.6
LAMBDA:  2       2.3   2.2   2.7   1.8   1.9   2.3   2.4   3.4   3.2
LAMBDA:  3       0.7   0.7   0.8   0.5   0.6   0.7   0.7   1.1   1.0

RMS F BY WAVELENGTH:

LAMBDA:  1     239.6 328.3 275.9 344.1 299.6 245.4 198.5 153.8 132.4
LAMBDA:  2     241.4 328.2 276.3 347.9 294.0 245.3 199.3 153.2 135.0
LAMBDA:  3     242.3 329.7 277.0 347.7 298.2 246.5 199.4 151.9 135.6


PARAMETER SHIFTS FOR DERIV  2 : set 2

                    SCALE FACTOR     OVERALL B
CURRENT VALUES:         1.0000         0.0000



                SITE  ATOM      OCCUP     X       Y       Z        B


CURRENT VALUES:     1    Se      0.7575  0.4821  0.4972  0.0938   55.3655

CURRENT VALUES:     2    Se      0.4584  0.5277  0.2832  0.0570   60.0000



*************************************************************
*************************************************************


*** Summary of solutions and their relationships
 to each other and to check solution ***


-------------------------------------------------------
solution #          1 with overall quality =   7.417267
Derivative           2 with           2 sites.
Overall scale =    1.000000     and  overall b of   0.0000000E+00
 0.4820956      0.4972169      9.3780220E-02  0.7575110       55.36546
 0.5276953      0.2831880      5.6996226E-02  0.4584486       60.00000
Best match of solution 1 -> solution 2:

-------- solution 1 --------    -------------solution 2 ------
site     x        y        z    site     x        y        z     DIST (A)
Derivative  2
 1    0.482    0.497    0.094    1    0.481    0.497    0.094    0.06
 2    0.528    0.283    0.057    2    0.527    0.288    0.055    0.15
Comparison of this solution with check solution:
Number of sites in this solution matching check=          2
... and number not matching =           0
by derivative, this is...
Deriv          nsame          ndifferent
```

1	0	0
2	2	0
3	0	0

All sites in this solution are contained in check soln

```
#!/bin/csh
#
#  set CCP4 and SOLVETMPDIR variables:
#
setenv CCP4_OPEN UNKNOWN
setenv SOLVETMPDIR /var/tmp
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
setenv SYMOP /usr/local/lib/solve/symop.lib
#
#
solve_giant <<EOD > solve.log

!command file to read in raw MAD data, scale, analyze and solve it----
title armadillo repeat of beta catenin 4-wavelength MAD data
logfile mad.logfile            ! write out most information to this file.
                          ! summary info will be written to "solve.prt"
@solve.setup                   ! get our standard information read in
readformatted                   ! or: readdenzo, readtrek, readccp4_unmerged
unmerged                       ! or; premerged

mad_atom se

refscattfactors                ! do not refine scattering factors (you can if
                          ! you want though)

       !  Comment out next line if you don't know any sites
checksolve                     ! compare solutions to the one input below

lambda 1                       ! info on wavelength #1 follows
label Wavelength #  1          ! a label for this wavelength
rawmadfile l1.int
wavelength 0.9000              ! wavelength value
fprimv_mad  -1.6               ! f' value at this wavelength
fprprv_mad  3.4                ! f'' value at this wavelength

! list of all SE positions in refined beta-catenin
! structure (offset by 0.5 in y from PDB file)

atomname se
xyz  0.2631041     0.6633824     2.8978506E-02
xyz  0.4166300     0.6113137     7.7325497E-03
xyz  0.4765674     0.7249608     2.5320712E-02
```

```
xyz  0.4591554      0.7427059      0.4719517
xyz  0.4083922      0.7455686      0.1403100
xyz  0.4416372      0.8393628      7.6342024E-02
xyz  0.1327285      0.4970000      0.4364171
xyz  9.6379094E-02  0.5855882      0.3802352
xyz  7.6066948E-02  0.6245000      0.3974865
xyz  0.1150683      0.7795883      0.3715025
xyz  0.1385160      0.7238529      0.4098982
xyz  9.2073016E-02  0.7063529      0.4022779
xyz  0.2152710      0.8265882      0.3764597
xyz  0.3304202      0.6161765      0.2311389
xyz  0.1806852      0.8512745      0.1618233

lambda 2
rawmadfile l2.int
wavelength 0.9794
fprimv_mad  -11.44
fprprv_mad  8.74

lambda 3
rawmadfile l3.int
wavelength 0.9797
fprimv_mad  -12.83
fprprv_mad  2.56

lambda 4
rawmadfile l4.int
wavelength 0.9897
fprimv_mad -2.42
fprprv_mad 1.13

nres 700                [approx # of residues in protein molecule]
nanomalous 15           [approx # of anomalously scattering atoms per protein]
acceptance 0.10
SCALE_MAD               ! read in and localscale the data
ANALYZE_MAD             ! run MADMRG and MADBST and analyze all the Pattersons
SOLVE                   ! Solve the structure
EOD
#
# Now run Resolve to do density modification
# (You can download it from http://resolve.lanl.gov
# if you do not have it yet)
#
```

```
resolve_giant << EOD > resolve.log
!solvent_content 0.40        !    solvent fraction
seq_file weis.seq
compare_file coords_met.pdb
EOD
#
#  That's it! Now resolve.mtz has your updated phases
#
echo 'Here are your SOLVE and resolve files:'
#
ls -l solve.prt solve.mtz solve.ezd resolve.mtz
#
echo 'All done.'
```

---------- COPYRIGHT NOTICE ----------
Los Alamos National Laboratory
This program was prepared by the Regents of the University of California at
Los Alamos National Laboratory (the University) under Contract No.
W-7405-ENG-36 with the U.S. Department of Energy (DOE). The University has
certain rights in the program pursuant to the contract and the program should
not be copied or distributed outside your organization. All rights in the
program are reserved by the DOE and the University. Neither the U.S.
Government nor the University makes any warranty, express or implied, or
assumes any liability or responsibility for the use of this software.

```
*****************************************************
*              --- SOLVE ---                  *
*                                              *
*   Automated structure solution for MAD and MIR     *
*                                              *
*  Please type "solvehelp" for on-line help   *
*     or see "http://solve.lanl.gov"        *
*****************************************************
```

(version 2.09 of 02-Apr-2005 / Size = 12)
Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

Dataset title: armadillo repeat of beta catenin 4-wavelength MAD data

Space group number is: 20
Space group name from file name is: c2221
Rescaling standard dataset to put it on approximate absolute scale. NRES =
700; <F**2> expected = 1372000.
; <F**2> observed in lowest resolution shell = 130033.9
... Scale factor = 10.55110

*** Analysis of this scaled MAD data set ***

Fbar,sigma,Delano,sigma for 4 wavelengths written to:
mad_fbar.scl

F+,sigma,F-,sigma for 4 wavelengths written to:
mad_fpfm.scl

** Completeness of Fbar data at each wavelength: **

Completeness of dataset 1 (F > 2.000000 * sigma)
Wavelength # 1 ! a label for this wavelength

Reflections observed:

shell	dmin	Possible	Found	% complete
1	5.400	2224	2196	98.7
2	4.050	2978	2958	99.3
3	3.780	1168	1155	98.9
4	3.578	1128	1118	99.1
5	3.375	1402	1384	98.7
6	3.240	1156	1141	98.7
7	3.105	1357	1324	97.6
8	2.970	1588	1528	96.2
9	2.835	1909	1833	96.0
10	2.700	2307	2142	92.8
total		17217	16779	97.5

Completeness of dataset 2 (F > 2.000000 * sigma)
set 2

Reflections observed:

shell	dmin	Possible	Found	% complete
1	5.400	2224	2203	99.1
2	4.050	2978	2966	99.6
3	3.780	1168	1163	99.6
4	3.578	1128	1121	99.4
5	3.375	1402	1383	98.6
6	3.240	1156	1136	98.3
7	3.105	1357	1326	97.7
8	2.970	1588	1547	97.4
9	2.835	1909	1830	95.9
10	2.700	2307	2157	93.5
total		17217	16832	97.8

Completeness of dataset 3 (F > 2.000000 * sigma)
set 3

Reflections observed:

shell	dmin	Possible	Found	% complete
1	5.400	2224	2201	99.0
2	4.050	2978	2949	99.0
3	3.780	1168	1159	99.2
4	3.578	1128	1121	99.4
5	3.375	1402	1380	98.4
6	3.240	1156	1129	97.7
7	3.105	1357	1322	97.4

8	2.970	1588	1540	97.0
9	2.835	1909	1836	96.2
10	2.700	2307	2169	94.0
total		17217	16806	97.6

Completeness of dataset 4 (F > 2.000000 * sigma)
set 4

Reflections observed:

shell	dmin	Possible	Found	% complete
1	5.400	2224	2194	98.7
2	4.050	2978	2960	99.4
3	3.780	1168	1163	99.6
4	3.578	1128	1121	99.4
5	3.375	1402	1383	98.6
6	3.240	1156	1140	98.6
7	3.105	1357	1337	98.5
8	2.970	1588	1551	97.7
9	2.835	1909	1855	97.2
10	2.700	2307	2213	95.9
total		17217	16917	98.3

** R-factors for F-bar data dispersive differences **

Dispersive differences lambda 2 - lambda 1
(Delta f-prime = 9.839999)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	2174	816.187	0.054	1.000	51.13	22.88	2.23
2	4.050	2926	830.537	0.039	1.000	33.38	23.27	1.43
3	3.780	1142	875.459	0.033	1.000	25.63	25.46	1.01
4	3.578	1113	778.337	0.036	1.001	23.92	25.32	0.94
5	3.375	1363	648.302	0.041	1.000	21.56	25.59	0.84
6	3.240	1120	573.931	0.045	1.000	17.92	26.41	0.68
7	3.105	1304	530.735	0.047	0.999	17.85	25.94	0.69
8	2.970	1500	484.570	0.048	1.000	10.11	27.45	0.37
9	2.835	1791	435.663	0.054	1.000	10.86	27.70	0.39
10	2.700	2076	390.779	0.062	1.001	0.01	30.33	0.00
Total:		16509	642.529	0.045	1.000	27.08	25.98	0.93

Recommended resolution cut-off = 2.78

Dispersive differences lambda 3 - lambda 1
(Delta f-prime = 11.23000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	2174	816.710	0.061	1.001	58.10	24.56	2.37
2	4.050	2924	831.337	0.043	1.000	37.92	24.82	1.53
3	3.780	1144	876.187	0.036	1.000	29.73	27.29	1.09
4	3.578	1109	776.335	0.039	1.001	26.98	26.64	1.01
5	3.375	1363	648.632	0.044	1.000	23.27	27.48	0.85
6	3.240	1121	574.021	0.047	1.000	19.84	27.84	0.71
7	3.105	1304	531.127	0.050	1.000	18.70	27.04	0.69
8	2.970	1501	484.179	0.053	1.000	15.34	28.35	0.54
9	2.835	1793	434.658	0.056	1.000	9.34	28.97	0.32
10	2.700	2078	390.331	0.064	1.001	2.99	31.06	0.10
Total:		16511	642.422	0.049	1.000	30.66	27.32	0.99

Recommended resolution cut-off = 2.75

Dispersive differences lambda 4 - lambda 1
(Delta f-prime = 0.8200001)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	2119	781.402	0.015	1.000	0.00	16.53	0.00
2	4.050	2908	824.047	0.014	1.000	0.00	17.97	0.00
3	3.780	1136	875.562	0.014	1.000	0.00	20.47	0.00
4	3.578	1099	778.232	0.016	1.000	0.00	19.72	0.00
5	3.375	1354	645.728	0.022	1.000	0.00	19.53	0.00
6	3.240	1108	577.205	0.025	1.000	0.00	19.41	0.00
7	3.105	1300	531.364	0.028	1.000	0.00	19.29	0.00
8	2.970	1488	487.445	0.029	1.000	0.00	19.71	0.00
9	2.835	1779	435.592	0.035	1.000	0.00	19.84	0.00
10	2.700	2077	389.602	0.043	1.000	0.01	21.57	0.00
Total:		16368	636.370	0.022	1.000	0.00	19.28	0.00

Recommended resolution cut-off = 2.70

Dispersive differences lambda 3 - lambda 2
(Delta f-prime = 1.390000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	2157	811.115	0.015	1.000	0.00	16.98	0.00
2	4.050	2908	828.386	0.014	1.000	0.00	16.72	0.00
3	3.780	1142	873.439	0.012	1.000	0.00	17.74	0.00
4	3.578	1095	777.296	0.015	1.000	0.00	16.99	0.00
5	3.375	1351	648.868	0.019	1.000	0.00	17.46	0.00
6	3.240	1101	578.541	0.022	1.000	0.00	16.94	0.00
7	3.105	1293	532.831	0.027	1.000	6.25	17.10	0.37
8	2.970	1500	485.711	0.030	1.000	5.81	17.93	0.32
9	2.835	1772	436.868	0.033	1.000	2.80	18.12	0.15
10	2.700	2066	389.851	0.040	1.000	0.01	20.02	0.00
Total:		16385	641.917	0.020	1.000	0.00	17.66	0.08

Recommended resolution cut-off = 2.70

Dispersive differences lambda 4 - lambda 2
(Delta f-prime = 9.020000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	2172	803.461	0.053	0.999	48.36	23.19	2.08
2	4.050	2932	826.525	0.037	1.000	30.61	23.49	1.30
3	3.780	1150	867.979	0.031	1.000	21.78	25.55	0.85
4	3.578	1112	777.724	0.033	1.000	20.65	24.78	0.83
5	3.375	1362	644.660	0.037	1.000	18.35	23.74	0.77
6	3.240	1122	576.081	0.040	1.000	16.82	23.24	0.72
7	3.105	1314	527.028	0.043	1.000	15.53	23.61	0.66
8	2.970	1519	482.101	0.045	1.000	11.86	24.48	0.48
9	2.835	1793	433.858	0.049	1.000	11.23	24.13	0.47
10	2.700	2098	386.547	0.056	0.999	0.00	26.85	0.00
Total:		16574	637.721	0.042	1.000	25.02	24.31	0.88

Recommended resolution cut-off = 2.81

Dispersive differences lambda 4 - lambda 3
(Delta f-prime = 10.41000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	2168	802.834	0.060	0.999	55.31	24.27	2.28
2	4.050	2927	829.553	0.042	1.000	35.98	24.38	1.48
3	3.780	1147	870.461	0.034	1.000	25.96	26.71	0.97
4	3.578	1114	777.550	0.037	1.000	25.38	25.64	0.99
5	3.375	1361	645.221	0.040	1.000	21.23	24.68	0.86
6	3.240	1120	575.033	0.043	1.000	19.55	23.69	0.83
7	3.105	1307	527.948	0.045	1.000	18.06	23.54	0.77
8	2.970	1510	480.822	0.048	1.000	14.77	24.19	0.61
9	2.835	1804	431.343	0.052	1.000	13.72	24.16	0.57
10	2.700	2121	383.763	0.057	0.999	7.37	26.50	0.28
Total:		16579	637.168	0.046	1.000	29.23	24.78	1.03

Recommended resolution cut-off = 2.70

Anomalous differences lambda 1 (f'' = 3.400000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	1725	804.284	0.051	1.000	43.09	27.31	1.58
2	4.050	2560	837.861	0.040	1.000	29.44	29.85	0.99
3	3.780	1008	879.864	0.035	1.000	17.57	33.97	0.52
4	3.578	992	782.253	0.040	1.000	20.85	33.11	0.63
5	3.375	1217	658.389	0.045	1.000	15.24	34.19	0.45
6	3.240	1004	575.517	0.052	1.001	14.30	34.37	0.42
7	3.105	1161	534.522	0.056	1.000	15.62	33.17	0.47
8	2.970	1342	492.107	0.057	1.000	4.57	34.42	0.13
9	2.835	1620	435.326	0.068	1.000	7.78	35.92	0.22
10	2.700	1878	394.082	0.077	1.000	0.01	37.85	0.00
Total:		14507	641.111	0.050	1.000	22.06	33.29	0.58

Recommended resolution cut-off = 3.02

Anomalous differences lambda 2 (f'' = 8.740000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	1740	803.429	0.083	1.001	75.20	34.94	2.15
2	4.050	2560	838.090	0.060	0.999	52.09	34.95	1.49
3	3.780	1019	876.695	0.049	1.000	38.86	36.78	1.06
4	3.578	994	778.083	0.056	1.000	40.16	35.62	1.13

5	3.375	1221	655.198	0.058	1.001	31.53	36.10	0.87
6	3.240	1015	572.173	0.064	1.001	27.41	35.53	0.77
7	3.105	1164	530.065	0.067	1.000	26.40	35.13	0.75
8	2.970	1377	486.713	0.071	1.000	23.40	36.02	0.65
9	2.835	1619	436.749	0.079	1.001	21.95	36.32	0.60
10	2.700	1896	392.176	0.088	1.001	14.51	40.45	0.36
Total:		14605	638.744	0.067	1.000	41.08	36.28	1.03

Recommended resolution cut-off = 2.70

Anomalous differences lambda 3 (f" = 2.560000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	1734	790.368	0.028	1.000	8.45	26.54	0.32
2	4.050	2546	834.436	0.024	1.000	0.00	26.73	0.00
3	3.780	1005	877.528	0.022	1.000	0.00	28.28	0.00
4	3.578	993	776.060	0.026	1.000	0.00	27.10	0.00
5	3.375	1217	652.855	0.030	1.001	0.00	27.28	0.00
6	3.240	1018	572.682	0.035	1.001	0.00	26.65	0.00
7	3.105	1158	533.230	0.038	1.000	5.24	25.43	0.21
8	2.970	1371	487.252	0.043	1.000	2.37	26.37	0.09
9	2.835	1626	435.299	0.049	1.000	4.92	26.58	0.18
10	2.700	1904	390.397	0.058	1.000	0.00	28.31	0.00
Total:		14572	635.578	0.033	1.000	0.00	26.94	0.08

Recommended resolution cut-off = 4.05

Anomalous differences lambda 4 (f" = 1.130000)

Differences by shell:

shell	dmin	nobs	Fbar	R	scale	SIGNAL	NOISE	S/N
1	5.400	1729	787.618	0.017	1.000	0.00	23.53	0.00
2	4.050	2539	828.206	0.018	1.000	0.00	24.79	0.00
3	3.780	1012	868.856	0.019	1.000	0.00	27.88	0.00
4	3.578	986	777.892	0.020	1.000	0.00	26.18	0.00
5	3.375	1221	652.180	0.025	1.000	0.00	24.08	0.00
6	3.240	1014	574.581	0.028	1.000	0.00	22.70	0.00
7	3.105	1171	528.892	0.030	1.000	0.00	22.22	0.00
8	2.970	1377	483.181	0.034	0.999	0.00	22.22	0.00
9	2.835	1646	432.946	0.040	1.000	3.84	21.62	0.18
10	2.700	1952	385.809	0.048	1.001	0.00	23.45	0.00

Total: 14647 630.861 0.025 1.000 0.00 23.82 0.02

Recommended resolution cut-off = 2.70

ANALYZE_MAD: Run MADMRG and MADBST on MAD data to get ready for SOLVE

Correlation of anomalous differences at different wavelengths.
(You should probably cut your data off at the resolution where
this drops below about 0.3. A good dataset has correlation
between peak and remote of at least 0.7 overall. Data with
correlations below about 0.5 probably are not contributing much.)

CORRELATION FOR
WAVELENGTH PAIRS

DMIN	1 VS 2	1 VS 3	1 VS 4	2 VS 3	2 VS 4	3 VS 4
5.40	0.84	0.66	0.42	0.79	0.41	0.36
4.05	0.75	0.53	0.36	0.69	0.35	0.33
3.78	0.65	0.43	0.21	0.60	0.23	0.19
3.58	0.67	0.38	0.24	0.58	0.27	0.22
3.38	0.56	0.31	0.19	0.50	0.19	0.17
3.24	0.53	0.28	0.12	0.40	0.14	0.14
3.11	0.48	0.21	0.14	0.36	0.18	0.16
2.97	0.44	0.25	0.11	0.38	0.18	0.11
2.84	0.41	0.21	0.08	0.32	0.13	0.06
2.70	0.33	0.11	0.10	0.25	0.13	0.12
ALL	0.63	0.37	0.22	0.52	0.24	0.19

Final refined values of fprime and fdoubleprime

Form factors at lambda = 0.9000
f-prime = -3.20
f" = 4.77

Form factors at lambda = 0.9794
f-prime = -12.20
f" = 10.92

Form factors at lambda = 0.9797
f-prime = -13.32
f" = 2.86

Form factors at lambda = 0.9897
f-prime = -4.63

```
f"      =   1.28

Fa Patterson from MADBST to be written to:
patterson.patt


Script file suitable for running SOLVE written to:
solve_mad.script

Datafile for SOLVE with MADMRG-compressed dataset
 ("Fnat",sig,"Fder",sig,"Delano",sig,iso diffs, ano diffs,
 <Fhcos>, <Fhsin> from MADBST) is:
solve.data


 ****   SOLVE:  Solutions to MIR or SIR datasets  ******

Derivatives considered:              4  (NSET)
Cross-vectors tested in HASSP:      20  (ICRMAX, DEFAULT=20)
HASSP solutions saved per deriv:    30  (NTOPHASSP, DEFAULT=30)
Fourier peaks saved per map:        40  (NTOPFOUR, DEFAULT=10)
Sites per derivative:               15  (NSOLSITE, DEFAULT=20)
Derivative solutions per seed:       5  (NTOPDERIV, DEFAULT=5)
Seeds per derivative tested:        10  (NSEEDTEST,DEFAULT=10)
Sorted seeds to use                  5  (NSEEDSOLVE, DEFAULT=5)
Number of final solutions saved:     5  (NTOPSOLVE, DEFAULT=5)


Solutions obtained will be compared to input solution (ICHECKSOLVE)
Correlated phasing used (CORRELPHASE)
Patterson map for derivative            2 will be read directly from:
patterson.patt

For derivative           2 the heavy atom
 structure factor components parallel to and
perpendicular to the native structure factor will be read from columns
         9 and          10

Datafile with          10 columns of data:
Title:MADMRG output (cols 1 to  8) and  MADBST fh cos,sin theta (c
Data: madmrg: MOCK FNAT
Data: madmrg: MOCK sig FNAT
Data: madmrg: MOCK FDER
Data: madmrg: MOCK sig FDER
Data: madmrg: MOCK DEL ANO
Data: madmrg: MOCK sig DEL ANO
Data: madmrg: Del iso for Patterson
Data: madmrg: Sigma of del iso for Patterson
Data: <Fa cos theta> = Fa component along Fo weighted by fom
Data: <Fa sin theta> = weighted Fa component perpendicular to Fo

Fnat,sigma taken from columns          1          2
Fder,sig,Delano,sig deriv           2 from cols:           3          4
         5          6
```

Check solution to be compared to all solutions found:

Derivative 1:
Site X Y Z

Derivative 2:

Site	X	Y	Z
1	0.263	0.663	0.029
2	0.417	0.611	0.008
3	0.477	0.725	0.025
4	0.459	0.743	0.472
5	0.408	0.746	0.140
6	0.442	0.839	0.076
7	0.133	0.497	0.436
8	0.096	0.586	0.380
9	0.076	0.624	0.397
10	0.115	0.780	0.372
11	0.139	0.724	0.410
12	0.092	0.706	0.402
13	0.215	0.827	0.376
14	0.330	0.616	0.231
15	0.181	0.851	0.162

Derivative 3:
Site X Y Z

Derivative 4:
Site X Y Z

****

ANALYZE_SOLVE: analysis of top 1 solutions

****

Solution 1

********** Analysis of this solution **********

****** Analysis of non-randomness of native Fourier map ******

A. Maps with distinct solvent regions havea high standard deviation
of local r.m.s. electron density.
For this map the SD of this local r.m.s. is 0.3484668

B. Maps with distinct solvent regions also have a high correlation
of local r.m.s. electron density with
density at neighboring locations. Typical
values for poor maps in this

structure solution are 0.1171082 +/- 5.5471703E-02
For this map the correlation of r.m.s. density
in neighboring boxes is 0.6060044
The correlation coefficient is used here in scoring.

Skew of the map is: 0.4625264

****** Analysis of derivative solutions with the difference Patterson ******
and with cross-validation difference Fouriers

Derivative # 2

List of sites analyzed for compatibility with difference Patterson

PEAK	X	Y	Z	OPTIMIZED RELATIVE OCCUPANCY
1	0.833	0.118	0.231	66.639
2	0.424	0.125	0.102	60.432
3	0.944	0.337	0.076	57.477
4	0.868	0.497	0.062	56.229
5	0.681	0.351	0.162	50.416
6	0.403	0.083	0.118	54.187
7	0.042	0.243	0.028	45.631
8	0.979	0.226	0.025	49.770
9	0.389	0.281	0.130	39.266
10	0.285	0.326	0.125	32.344
11	0.410	0.212	0.097	30.225
12	0.361	0.222	0.090	35.002
13	0.910	0.243	0.139	26.396
14	0.889	0.104	0.012	19.209
15	0.764	0.167	0.028	24.865

Evaluation of this test soln with 15 sites after optimizing
occupancy of each site

Cross-vectors for sites 1 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.667	-0.236	0.500	7762.94	8881.45	2
2	-1.667	0.000	0.037	6621.59	8881.45	2
3	0.000	-0.236	-0.463	6340.11	8881.45	2

Cross-vectors for sites 2 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.410	0.007	-0.130	7287.10	4027.09	1
2	-1.257	-0.243	0.370	7227.19	4027.09	1
3	-1.257	0.007	0.167	8512.62	4027.09	1
4	-0.410	-0.243	-0.333	5953.13	4027.09	1

Cross-vectors for sites 2 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

1	-0.847	0.000	0.296	4840.62	7303.97	2
2	0.000	-0.250	-0.204	5029.59	7303.97	2

Cross-vectors for sites 3 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.111	0.219	-0.155	5636.24	3830.21	1
2	-1.778	-0.455	0.345	5094.47	3830.21	1
3	-1.778	0.219	0.192	3562.32	3830.21	1
4	0.111	-0.455	-0.308	5183.66	3830.21	1

Cross-vectors for sites 3 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.521	0.212	-0.025	4484.42	3473.44	1
2	-1.368	-0.462	0.475	4281.19	3473.44	1
3	-1.368	0.212	0.322	4576.54	3473.44	1
4	0.521	-0.462	-0.178	4893.72	3473.44	1

Cross-vectors for sites 3 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.889	-0.674	0.500	4699.89	6607.25	2
2	-1.889	0.000	0.347	5774.54	6607.25	2
3	0.000	-0.674	-0.153	3937.11	6607.25	2

Cross-vectors for sites 4 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.035	0.378	-0.169	3412.38	3747.03	1
2	-1.701	-0.615	0.331	6116.88	3747.03	1
3	-1.701	0.378	0.206	7368.03	3747.03	1
4	0.035	-0.615	-0.294	3544.71	3747.03	1

Cross-vectors for sites 4 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.444	0.372	-0.039	3604.90	3398.01	1
2	-1.292	-0.622	0.461	3151.94	3398.01	1
3	0.444	-0.622	-0.164	4100.05	3398.01	1

Cross-vectors for sites 4 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.076	0.160	-0.014	3997.58	3231.89	1
2	-1.812	-0.833	0.486	3929.86	3231.89	1
3	-1.812	0.160	0.361	4952.37	3231.89	1
4	-0.076	-0.833	-0.139	4119.74	3231.89	1

Cross-vectors for sites 4 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.736	-0.993	0.500	6925.96	6323.41	2
2	-1.736	0.000	0.375	1726.50	6323.41	2
3	0.000	-0.993	-0.125	6625.76	6323.41	2

Cross-vectors for sites 5 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.153	0.233	-0.069	4355.24	3359.64	1
2	-1.514	-0.469	0.431	6269.49	3359.64	1
3	-1.514	0.233	0.106	7573.38	3359.64	1

4	-0.153	-0.469	-0.394	4687.07	3359.64	1

Cross-vectors for sites 5 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.257	0.226	0.060	4052.52	3046.70	1
2	-1.104	-0.476	0.560	2630.22	3046.70	1
3	-1.104	0.226	0.236	3893.40	3046.70	1
4	0.257	-0.476	-0.264	3472.42	3046.70	1

Cross-vectors for sites 5 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.264	0.014	0.086	2084.01	2897.75	1
2	-1.625	-0.688	0.586	5054.75	2897.75	1
3	-1.625	0.014	0.262	3451.61	2897.75	1
4	-0.264	-0.688	-0.238	3180.23	2897.75	1

Cross-vectors for sites 5 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.188	-0.146	0.100	2246.14	2834.82	1
2	-1.549	-0.847	0.600	3705.33	2834.82	1
3	-1.549	-0.146	0.275	2381.94	2834.82	1
4	-0.188	-0.847	-0.225	1948.49	2834.82	1

Cross-vectors for sites 5 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.361	-0.701	0.500	871.956	5083.48	2
2	-1.361	0.000	0.176	5352.87	5083.48	2
3	0.000	-0.701	-0.324	3443.16	5083.48	2

Cross-vectors for sites 6 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.236	-0.201	0.387	4207.56	3610.95	1
2	-0.431	-0.201	-0.350	4478.70	3610.95	1

Cross-vectors for sites 6 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.021	-0.042	0.016	-673.853	3274.60	1
2	-0.826	-0.042	0.280	3291.93	3274.60	1
3	-0.021	-0.208	-0.220	3112.12	3274.60	1

Cross-vectors for sites 6 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.542	-0.253	0.042	5079.83	3114.51	1
2	-1.347	-0.420	0.542	4274.16	3114.51	1
3	-1.347	-0.253	0.306	4081.66	3114.51	1
4	-0.542	-0.420	-0.194	2685.58	3114.51	1

Cross-vectors for sites 6 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.465	-0.413	0.056	4000.08	3046.87	1
2	-1.271	-0.580	0.556	4452.05	3046.87	1

Cross-vectors for sites 6 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

1	-1.083	-0.434	0.456	5414.77	2731.87	1
2	-1.083	-0.267	0.220	2516.87	2731.87	1

Cross-vectors for sites 6 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.000	-0.167	-0.236	3159.05	5872.43	2

Cross-vectors for sites 7 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.875	-0.361	0.296	3599.32	3040.81	1
2	-0.875	0.125	0.241	3969.16	3040.81	1
3	-0.792	-0.361	-0.259	3355.28	3040.81	1

Cross-vectors for sites 7 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.382	0.118	-0.074	2364.84	2757.57	1
2	-0.465	-0.368	0.426	3749.95	2757.57	1
3	-0.465	0.118	0.370	1423.26	2757.57	1
4	-0.382	-0.368	-0.130	3835.31	2757.57	1

Cross-vectors for sites 7 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.903	-0.094	-0.049	3369.54	2622.75	1
2	-0.986	-0.580	0.451	3206.65	2622.75	1
3	-0.986	-0.094	0.396	4351.27	2622.75	1
4	-0.903	-0.580	-0.104	2023.07	2622.75	1

Cross-vectors for sites 7 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.826	-0.253	-0.035	1088.48	2565.80	1
2	-0.910	-0.740	0.465	3162.91	2565.80	1
3	-0.910	-0.253	0.410	1789.85	2565.80	1
4	-0.826	-0.740	-0.090	2355.53	2565.80	1

Cross-vectors for sites 7 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.639	-0.108	-0.134	2542.93	2300.53	1
2	-0.722	-0.594	0.366	2348.91	2300.53	1
3	-0.722	-0.108	0.310	3219.99	2300.53	1
4	-0.639	-0.594	-0.190	2362.56	2300.53	1

Cross-vectors for sites 7 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.361	0.160	-0.090	3156.54	2472.61	1
2	-0.444	-0.326	0.410	2106.12	2472.61	1
3	-0.444	0.160	0.354	2426.81	2472.61	1
4	-0.361	-0.326	-0.146	2154.48	2472.61	1

Cross-vectors for sites 7 and 7 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.083	-0.486	0.500	4286.02	4164.42	2
2	-0.083	0.000	0.444	2470.77	4164.42	2
3	0.000	-0.486	-0.056	2406.42	4164.42	2

Cross-vectors for sites 8 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.812	-0.344	0.294	3207.76	3316.62	1
2	-1.812	0.108	0.243	4100.44	3316.62	1
3	0.146	-0.344	-0.257	3973.69	3316.62	1

Cross-vectors for sites 8 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.556	0.101	-0.076	2417.78	3007.69	1
2	-1.403	0.101	0.373	3341.48	3007.69	1

Cross-vectors for sites 8 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.924	-0.111	0.398	2481.87	2860.64	1
2	0.035	-0.562	-0.102	2663.22	2860.64	1

Cross-vectors for sites 8 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.111	-0.271	-0.037	3541.12	2798.52	1
2	-1.847	-0.722	0.463	3921.51	2798.52	1
3	-1.847	-0.271	0.412	4156.23	2798.52	1
4	0.111	-0.722	-0.088	3698.03	2798.52	1

Cross-vectors for sites 8 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.299	-0.125	-0.137	1984.82	2509.19	1
2	-1.660	-0.576	0.363	2601.68	2509.19	1
3	0.299	-0.576	-0.188	3273.56	2509.19	1

Cross-vectors for sites 8 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.382	0.142	0.356	2557.58	2696.88	1

Cross-vectors for sites 8 and 7 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.021	-0.469	0.498	6745.58	4542.14	2
2	0.938	-0.469	-0.053	2310.14	2271.07	1

Cross-vectors for sites 8 and 8 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.000	-0.451	-0.051	2375.81	4954.12	2

Cross-vectors for sites 9 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.444	0.163	-0.102	3544.70	2616.63	1
2	-1.222	-0.399	0.398	4240.80	2616.63	1
3	-1.222	0.163	0.139	3683.57	2616.63	1
4	-0.444	-0.399	-0.361	3122.46	2616.63	1

Cross-vectors for sites 9 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.812	-0.406	0.528	1913.22	2372.90	1
2	-0.812	0.156	0.269	1825.17	2372.90	1
3	-0.035	-0.406	-0.231	3399.44	2372.90	1

Cross-vectors for sites 9 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.556	-0.618	-0.206	3349.93	2256.89	1

Cross-vectors for sites 9 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.479	-0.215	0.067	3264.66	2207.88	1
2	-1.257	-0.778	0.567	1449.56	2207.88	1
3	-1.257	-0.215	0.308	1814.21	2207.88	1

Cross-vectors for sites 9 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.292	-0.069	-0.032	2424.97	1979.61	1
2	-1.069	-0.632	0.468	2465.55	1979.61	1
3	-0.292	-0.632	-0.292	3204.55	1979.61	1

Cross-vectors for sites 9 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.014	0.198	0.012	3177.71	2127.69	1
2	-0.792	-0.365	0.512	3044.27	2127.69	1
3	-0.014	-0.365	-0.248	3243.12	2127.69	1

Cross-vectors for sites 9 and 7 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.347	0.038	0.102	1103.84	1791.75	1
2	-0.431	-0.524	0.602	2170.75	1791.75	1
3	-0.431	0.038	0.343	2164.14	1791.75	1
4	0.347	-0.524	-0.157	1256.26	1791.75	1

Cross-vectors for sites 9 and 8 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.368	-0.507	0.604	2676.42	1954.26	1
2	-1.368	0.056	0.345	2152.03	1954.26	1
3	-0.590	-0.507	-0.155	3131.40	1954.26	1

Cross-vectors for sites 9 and 9 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.778	-0.562	0.500	2170.29	3083.61	2
2	-0.778	0.000	0.241	269.253	3083.61	2
3	0.000	-0.562	-0.259	1890.65	3083.61	2

Cross-vectors for sites 10 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.549	-0.444	-0.356	1569.29	2155.39	1

Cross-vectors for sites 10 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.139	0.201	0.023	3570.39	1954.62	1
2	-0.708	-0.451	0.523	2580.55	1954.62	1
3	-0.708	0.201	0.273	1941.11	1954.62	1
4	-0.139	-0.451	-0.227	3095.37	1954.62	1

Cross-vectors for sites 10 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.229	-0.663	0.549	1361.23	1859.06	1
2	-1.229	-0.010	0.299	812.953	1859.06	1
3	-0.660	-0.663	-0.201	1162.24	1859.06	1

Cross-vectors for sites 10 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.583	-0.170	0.062	1525.18	1818.69	1
2	-1.153	-0.823	0.562	1713.38	1818.69	1
3	-1.153	-0.170	0.312	1447.40	1818.69	1
4	-0.583	-0.823	-0.188	2177.62	1818.69	1

Cross-vectors for sites 10 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.965	-0.677	0.463	804.070	1630.66	1
2	-0.965	-0.024	0.213	2398.00	1630.66	1
3	-0.396	-0.677	-0.287	2307.83	1630.66	1

Cross-vectors for sites 10 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.118	0.243	0.007	2537.54	1752.64	1
2	-0.688	-0.410	0.507	3170.78	1752.64	1
3	-0.688	0.243	0.257	2190.48	1752.64	1
4	-0.118	-0.410	-0.243	1582.98	1752.64	1

Cross-vectors for sites 10 and 7 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.243	0.083	0.097	1496.05	1475.91	1
2	-0.326	-0.569	0.597	2023.81	1475.91	1
3	0.243	-0.569	-0.153	1806.99	1475.91	1

Cross-vectors for sites 10 and 8 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.694	0.101	0.100	2272.88	1609.78	1
2	-1.264	-0.552	0.600	1632.16	1609.78	1

Cross-vectors for sites 10 and 9 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.104	0.045	-0.005	-53.2615	1270.03	1

Cross-vectors for sites 10 and 10 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.569	-0.653	0.500	271.114	2092.32	2

Cross-vectors for sites 11 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.424	0.094	-0.134	3204.95	2014.15	1
2	-1.243	-0.330	0.366	2067.18	2014.15	1
3	-0.424	-0.330	-0.329	2210.86	2014.15	1

Cross-vectors for sites 11 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.014	0.087	-0.005	2411.42	1826.54	1
2	-0.833	0.087	0.301	1640.39	1826.54	1

3	-0.014	-0.337	-0.199	2126.26	1826.54	1

Cross-vectors for sites 11 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.535	-0.125	0.021	2621.83	1737.24	1
2	-1.354	-0.125	0.326	2123.10	1737.24	1

Cross-vectors for sites 11 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.458	-0.285	0.035	2330.40	1699.52	1
2	-1.278	-0.708	0.535	2280.89	1699.52	1
3	-1.278	-0.285	0.340	1625.96	1699.52	1
4	-0.458	-0.708	-0.160	1764.26	1699.52	1

Cross-vectors for sites 11 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.271	-0.139	-0.065	1080.50	1523.81	1
2	-1.090	-0.562	0.435	2161.71	1523.81	1
3	-0.271	-0.562	-0.259	994.003	1523.81	1

Cross-vectors for sites 11 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.007	-0.295	-0.215	3955.88	3275.59	2

Cross-vectors for sites 11 and 7 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.368	-0.031	0.069	1773.23	1379.20	1
2	-0.451	-0.031	0.375	1904.71	1379.20	1
3	0.368	-0.455	-0.125	2211.70	1379.20	1

Cross-vectors for sites 11 and 8 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.389	-0.438	0.572	1137.69	1504.30	1
2	-0.569	-0.438	-0.123	961.247	1504.30	1

Cross-vectors for sites 11 and 9 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.799	-0.493	0.468	1078.95	1186.81	1
2	0.021	-0.493	-0.227	1321.44	1186.81	1

Cross-vectors for sites 11 and 10 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.125	-0.115	-0.028	1329.81	977.607	1

Cross-vectors for sites 11 and 11 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.000	-0.424	-0.194	-1639.13	1827.10	2

Cross-vectors for sites 12 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 12 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 12 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.583	-0.559	-0.167	1751.71	2011.80	1

Cross-vectors for sites 12 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 12 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.042	-0.128	0.248	1565.15	1764.63	1

Cross-vectors for sites 12 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 12 and 7 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 12 and 8 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 12 and 9 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.750	-0.059	0.280	1772.06	1374.37	1

Cross-vectors for sites 12 and 10 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.646	-0.104	0.285	514.471	1132.11	1
2	0.076	-0.549	-0.215	1131.18	1132.11	1

Cross-vectors for sites 12 and 11 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 12 and 12 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.722	0.000	0.319	2689.29	2450.24	2

Cross-vectors for sites 13 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.076	0.125	-0.093	1274.29	1758.97	1
2	-1.743	-0.361	0.407	1256.19	1758.97	1

Cross-vectors for sites 13 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 13 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 13 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.042	-0.253	0.076	1318.54	1484.20	1
2	-1.778	-0.253	0.299	1020.55	1484.20	1

Cross-vectors for sites 13 and 5 (excluding origin; 1000 = 1 sigma):

| # | U | V | W | HEIGHT | PRED HEIGHT | SYMM# |

1	0.229	-0.108	-0.023	1840.21	1330.75	1
2	-1.590	-0.594	0.477	1631.21	1330.75	1
3	-1.590	-0.108	0.199	1009.35	1330.75	1

Cross-vectors for sites 13 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.507	0.160	0.021	2704.85	2860.59	2
2	0.507	-0.326	-0.257	3481.26	2860.59	2

Cross-vectors for sites 13 and 7 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.868	0.000	0.111	2352.91	2408.93	2
2	-0.951	0.000	0.333	2739.52	2408.93	2

Cross-vectors for sites 13 and 8 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.069	0.017	0.113	1114.51	1313.71	1
2	-0.069	-0.469	-0.164	1548.05	1313.71	1

Cross-vectors for sites 13 and 9 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.521	-0.038	0.009	1660.51	1036.45	1
2	-1.299	-0.038	0.231	1359.61	1036.45	1
3	0.521	-0.524	-0.269	2189.55	1036.45	1

Cross-vectors for sites 13 and 10 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.625	-0.083	0.014	892.150	853.749	1

Cross-vectors for sites 13 and 11 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.500	-0.455	-0.236	2301.77	1595.61	2

Cross-vectors for sites 13 and 12 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.271	-0.465	0.549	549.616	923.892	1

Cross-vectors for sites 13 and 13 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.000	-0.486	-0.278	1383.03	1393.46	2

Cross-vectors for sites 14 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 14 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.465	-0.021	-0.090	108.051	1160.86	1

Cross-vectors for sites 14 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.833	-0.441	0.435	1267.20	1104.10	1
2	-1.833	-0.233	0.412	1682.35	1104.10	1

Cross-vectors for sites 14 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.021	-0.392	-0.051	831.384	1080.13	1

Cross-vectors for sites 14 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.208	-0.247	-0.150	1125.05	968.457	1
2	0.208	-0.455	-0.174	678.068	968.457	1

Cross-vectors for sites 14 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.486	-0.188	-0.130	1833.41	1040.90	1

Cross-vectors for sites 14 and 7 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.931	-0.139	0.461	1826.99	876.551	1
2	0.847	-0.347	-0.039	-428.749	876.551	1

Cross-vectors for sites 14 and 8 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.090	-0.330	-0.037	1844.90	956.056	1

Cross-vectors for sites 14 and 9 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.500	-0.385	-0.141	2399.63	1508.55	2

Cross-vectors for sites 14 and 10 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.604	-0.222	-0.113	786.134	621.317	1
2	-1.174	-0.431	0.387	856.146	621.317	1
3	-1.174	-0.222	0.363	1371.82	621.317	1

Cross-vectors for sites 14 and 11 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.299	-0.316	0.414	1643.83	580.605	1
2	0.479	-0.316	-0.109	1386.51	580.605	1

Cross-vectors for sites 14 and 12 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 14 and 13 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 14 and 14 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.778	-0.208	0.500	-655.210	738.006	2
2	-1.778	0.000	0.477	-581.052	738.006	2

Cross-vectors for sites 15 and 1 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.597	-0.285	0.296	1900.63	1656.99	1
2	-0.069	-0.285	-0.259	569.216	1656.99	1

Cross-vectors for sites 15 and 2 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

1	-1.188	0.042	0.370	2017.74	1502.65	1
2	0.340	-0.292	-0.130	1191.82	1502.65	1

Cross-vectors for sites 15 and 3 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.181	-0.170	-0.049	840.911	1429.18	1
2	-1.708	-0.503	0.451	4159.17	2858.36	2
3	-1.708	-0.170	0.396	1630.48	1429.18	1

Cross-vectors for sites 15 and 4 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.632	-0.663	0.465	572.442	1398.15	1

Cross-vectors for sites 15 and 5 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 15 and 6 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.361	-0.250	-0.146	1426.51	1347.37	1

Cross-vectors for sites 15 and 7 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.722	-0.076	0.000	2793.05	2269.26	2
2	0.722	-0.410	-0.056	2868.65	1134.63	1

Cross-vectors for sites 15 and 8 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.215	-0.059	0.002	1500.61	2475.09	2
2	-0.215	-0.392	-0.053	908.491	1237.54	1

Cross-vectors for sites 15 and 9 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	0.375	-0.115	-0.102	-118.429	976.354	1

Cross-vectors for sites 15 and 10 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.049	-0.493	0.403	534.750	804.249	1
2	0.479	-0.493	-0.153	1660.95	804.249	1

Cross-vectors for sites 15 and 11 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.174	-0.378	0.431	-359.155	751.550	1

Cross-vectors for sites 15 and 12 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#

Cross-vectors for sites 15 and 13 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-0.146	-0.076	-0.111	279.595	656.333	1
2	-0.146	-0.410	-0.167	-1006.17	656.333	1

Cross-vectors for sites 15 and 14 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.653	0.062	0.461	-241.305	477.647	1

Cross-vectors for sites 15 and 15 (excluding origin; 1000 = 1 sigma):

#	U	V	W	HEIGHT	PRED HEIGHT	SYMM#
1	-1.528	0.000	0.444	963.123	1236.56	2
2	0.000	-0.333	-0.056	1983.23	1236.56	2

Total of 209 of 465 patterson peaks used more than once.

Overall quality of this Patterson soln = 27059.9
Overall quality of the fit to patterson = 0.896753
Avg normalized peak height = 1254.87

Cross-validation fouriers calculated with all heavy atoms in
all derivs except the site being evaluated and any sites equivalent to it.

Site	x	y	z	occ	B	-- PEAK HEIGHT --
1	0.830	0.116	0.231	0.619	26.457	31.27
2	0.423	0.125	0.103	0.725	37.223	26.57
3	0.942	0.338	0.076	0.712	38.619	26.42
4	0.868	0.497	0.063	0.527	24.602	25.46
5	0.680	0.352	0.162	0.661	51.242	21.48
6	0.404	0.084	0.119	0.665	43.483	22.98
7	0.045	0.243	0.028	0.600	60.000	18.57
8	0.977	0.225	0.026	0.686	60.000	19.28
9	0.388	0.281	0.129	0.466	42.962	16.04
10	0.288	0.328	0.125	0.450	58.596	12.68
11	0.409	0.211	0.097	0.455	55.181	12.94
12	0.363	0.223	0.091	0.275	29.266	11.76
13	0.911	0.243	0.140	0.407	60.000	9.35
14	0.892	0.106	0.011	0.515	60.000	7.60
15	0.766	0.166	0.028	0.463	60.000	8.12

Summary of scoring for this solution:

	-- over many solutions--		-- this solution --	
Criteria	MEAN	SD	VALUE	Z-SCORE
Pattersons:	5.12	0.500	13.7	17.2
Cross-validation Fourier:	22.1	4.43	197.	39.4
NatFourier CCx100:	11.7	5.55	60.6	8.81
Mean figure of meritx100:	0.000E+00	5.00	68.8	13.8
Correction for Z-scores:				-14.5

Overall Z-score value: 64.6

****** Overall analysis of phasing for solution 1**********

*** Re-estimation of scattering factors by refinement of occupancies using ***
dispersive and anomalous differences.

Estimation of scattering factors at each wavelength

by refinement of occupancies relative to those found from the
initial refinement carried out with data from MADMRG.

Refining iso occupancies for iso diffs lambda 2 - lambda 1

Results of refinement:
Ratio of occupancies to standard refinement: 0.972 +/- 0.131
Delta f-prime based on input f-prime values: 8.996
New estimate of delta f-prime: 8.741 +/- 1.179
with sign of: -1. and Z of 52.5

Refining iso occupancies for iso diffs lambda 3 - lambda 1

Results of refinement:
Ratio of occupancies to standard refinement: 1.009 +/- 0.117
Delta f-prime based on input f-prime values: 10.120
New estimate of delta f-prime: 10.211 +/- 1.182
with sign of: -1. and Z of 60.6

Refining iso occupancies for iso diffs lambda 4 - lambda 1

Results of refinement:
Ratio of occupancies to standard refinement: 0.706 +/- 0.582
Delta f-prime based on input f-prime values: 1.421
New estimate of delta f-prime: 1.003 +/- 0.827
with sign of: -1. and Z of 6.1

Refining iso occupancies for iso diffs lambda 3 - lambda 2

Results of refinement:
Ratio of occupancies to standard refinement: 2.338 +/- 1.167
Delta f-prime based on input f-prime values: 1.124
New estimate of delta f-prime: 2.628 +/- 1.311
with sign of: -1. and Z of 39.8

Refining iso occupancies for iso diffs lambda 4 - lambda 2

Results of refinement:
Ratio of occupancies to standard refinement: 1.121 +/- 0.153
Delta f-prime based on input f-prime values: 7.575
New estimate of delta f-prime: 8.492 +/- 1.158
with sign of: 1. and Z of 52.8

Refining iso occupancies for iso diffs lambda 4 - lambda 3

Results of refinement:
Ratio of occupancies to standard refinement: 1.149 +/- 0.109
Delta f-prime based on input f-prime values: 8.699
New estimate of delta f-prime: 9.993 +/- 0.948

with sign of: 1. and Z of 61.4

Refining ano occupancies for ano diffs lambda 1

Results of refinement:
Ratio of occupancies to standard refinement: 0.876 +/- 0.199
f'' value based on input values: 4.772
New estimate of f'': 4.178 +/- 0.951

Refining ano occupancies for ano diffs lambda 2

Results of refinement:
Ratio of occupancies to standard refinement: 0.628 +/- 0.217
f'' value based on input values: 10.918
New estimate of f'': 6.861 +/- 2.367

Refining ano occupancies for ano diffs lambda 3

Results of refinement:
Ratio of occupancies to standard refinement: 0.643 +/- 0.204
f'' value based on input values: 2.857
New estimate of f'': 1.838 +/- 0.583

Refining ano occupancies for ano diffs lambda 4

Results of refinement:
Ratio of occupancies to standard refinement: 0.429 +/- 0.285
f'' value based on input values: 1.277
New estimate of f'': 0.548 +/- 0.364

Fitting f-prime values.

Restraints:

Lambda	Target f-prime	final f-prime	weight
1	-3.205	-3.205	0.001
2	-12.200	-11.723	0.001
3	-13.324	-13.711	0.001
4	-4.625	-3.554	0.001

Delta-fprime targets:

Lambda	i	j	target delta-fprime	final delta-fprime	wgt
1	2		8.741	8.519	52.48

```
  1  3          10.211      10.506      60.63
  1  4           1.003       0.349       6.05
  2  3           2.628       1.988      39.81
  4  2           8.492       8.170      52.84
  4  3           9.993      10.157      61.36

Residual for restraints:     0.39059E-01
Residual for targets:      5.8275



              Final refined values of f-prime and f"

  Wavelength  ------- f-prime --------       --------f"--------------
     last refinement       Refined     last refinement       Refined

   1        -3.205        -3.205            4.772         4.178

   2       -12.200       -11.723           10.918         6.861

   3       -13.324       -13.711            2.857         1.838

   4        -4.625        -3.554            1.277         0.548


*** Done with re-estimation of scattering factors ***



HEAVY:  Refine heavy atom parameters

File title: armadillo repeat of beta catenin 4-wavelength MAD data
CRYSTALLOGRAPHIC PARAMETERS

   A =  64.10     B = 102.00     C = 187.00     alpha =  90.00     beta =  90.00     gamma =  90.00

PHASES CALCULATED EVERY   5 DEGREES
RESIDUALS CALCULATED ON EXTRA ZEROTH CYCLE ONLY
SIGMAS FROM data FILE WILL BE USED
STATISTICS WILL BE PRINTED ON ZEROTH CYCLE, SHIFTS ON LAST
PHASING WILL BE DONE TAKING INTO ACCOUNT THE CORRELATIONS AMONG DERIVATIVES
THE GROUPS OF DERIVATIVES WITH CORRELATIONS WILL BE UPDATED
THE BETA VALUES FOR EACH DERIV  WILL BE SET TO 1.0
PHASE-AVERAGED RESIDUALS WILL BE USED FOR PHASING

TYPE OF REFINEMENT SELECTED:
UNPHASED ORIGIN-REMOVED  PATTERSON REFINEMENT ONLY

Bayesian correlated MAD phasing will be used with wavelength          2
 as the reference wavelength.

RESOLUTION LIMITS IN ANGSTROMS:      2.700    20.000
MINIMUM RATIO OF FDER TO RMS LACK-OF-CLOSURE FOR  INCLUSION IN REFINEMENT OR PHASING=      0.000
MINIMUM NATIVE F:      0.000
```

```
MINIMUM FIGURE OF MERIT FOR PHASED REFINEMENT:      0.000
MINIMUM ALLOWED ISOTROPIC B:      0.000
PARAMETER SHIFTS GREATER THAN     0.0000 TIMES SIGMA  WILL BE SCALED BY     0.5000
MINIMUM RATIO OF FNAT/SIGMA OR FDER/SIGMA TO INCLUDE:      1.000

NUMBER OF REFINEMENT CYCLES IS      2
DERIVATIVES REFINED DURING THESE CYCLES ARE :  0  0
TYPE OF OUTPUT SELECTED IS: +10 COLUMNS OF HENDRICKSON-LATTMAN COEFFICIENTS
1 INPUT data FILE WITH  16 COLUMNS IS:
mad_fpfm.scl
COLUMN  0 : mad_fpfm.scl Fnat,sig,(F+,sig,F-,sig)n
COLUMN  1 : F+ from separateano Wavelength #  1         ! a label for this wavelengt
COLUMN  2 : sig of F+           Wavelength #  1         ! a label for this wavelengt
COLUMN  3 : F- from separateano Wavelength #  1         ! a label for this wavelengt
COLUMN  4 : sig of F-           Wavelength #  1         ! a label for this wavelengt
COLUMN  5 : F+ from separateano set 2
COLUMN  6 : sig of F+           set 2
COLUMN  7 : F- from separateano set 2
COLUMN  8 : sig of F-           set 2
COLUMN  9 : F+ from separateano set 3
COLUMN 10 : sig of F+           set 3
COLUMN 11 : F- from separateano set 3
COLUMN 12 : sig of F-           set 3
COLUMN 13 : F+ from separateano set 4
COLUMN 14 : sig of F+           set 4
COLUMN 15 : F- from separateano set 4
COLUMN 16 : sig of F-           set 4


DERIVATIVE INFORMATION FOR  4 COMPOUNDS


COMPOUND 1     TEST REFINEMENT LAMBDA 4  (ANO ONLY)
COLUMNS FOR F+, SIGMA, F-, SIGMA     1    2    3    4
THIS DERIVATIVE WILL NOT BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
ONLY ANO DIFFERENCES WILL BE USED IN REFINEMENT AND PHASING FOR THIS DERIVATIVE.
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
NO PARAMETERS REFINED FOR ATOM LAM1 WITH ZERO OCCUPANCY

COMPOUND 2     set 2
COLUMNS FOR F+, SIGMA, F-, SIGMA     5    6    7    8
THIS DERIVATIVE WILL BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000

COMPOUND 3     set 3
COLUMNS FOR F+, SIGMA, F-, SIGMA     9   10   11   12
THIS DERIVATIVE WILL NOT BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
```

```
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
NO PARAMETERS REFINED FOR ATOM LAM3 WITH ZERO OCCUPANCY

COMPOUND 4     set 4
COLUMNS FOR F+, SIGMA, F-, SIGMA    13   14   15   16
THIS DERIVATIVE WILL NOT BE USED IN PHASING
ANOMALOUS DIFFERENCES WILL BE USED IN PHASING FOR THIS DERIVATIVE
AFTER OVERALL SCALING OBSERVED STRUCTURE FACTORS AND SIGMAS WILL BE
DIVIDED BY :     1.000*EXP(     0.000*(SIN theta/LAMBDA)**2 )
THEN SIGMAS WILL BE MULTIPLIED BY      1.000
NO PARAMETERS REFINED FOR ATOM LAM4 WITH ZERO OCCUPANCY
CARRYING OUT STANDARD REFINEMENT
Total of           2 cycles will be done
Derivs refined will be           0          0

SUMMARY OF RESULTS ON FINAL CYCLE:

NUMBER OF REFLECTIONS READ =   17155
NUMBER OF F .GT. FMIN      =   17155
NUMBER OF F IN RES. LIMITS =   17155
NUMBER OF F .GT. MIN <M>   =   17024
NUMBER OF F USED TO REFINE =       0



FIGURE OF MERIT <       0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9   1.0
# OF REFLECTIONS      1760  1608  1509  1359  1358  1325  1397  1755  2323  2760

FIGURE OF MERIT WITH RESOLUTION
DMIN:          TOTAL    9.09   5.96   4.72   4.03   3.57   3.24   2.99   2.79
N:             17154    946   1466   1815   2122   2386   2623   2798   2998
MEAN FIG MERIT:   0.55   0.73   0.75   0.67   0.60   0.57   0.51   0.44   0.39

RMS ANOMALOUS FH/E  [f" PART OF FH / RMS ANO ERROR]:

LAMBDA:  1        0.6    1.2    1.3    0.9    0.7    0.6    0.5    0.4    0.3
LAMBDA:  2        1.0    1.3    1.3    1.2    1.1    1.0    0.9    0.7    0.6
LAMBDA:  3        0.3    0.5    0.6    0.4    0.3    0.3    0.2    0.2    0.2
LAMBDA:  4        0.1    0.1    0.2    0.1    0.1    0.1    0.1    0.1    0.1

RMS DISPERSIVE FH/E  [Delta-f-prime PART OF FH / RMS DISPERSIVE ERROR]:

L1 VS L2:         1.0    1.5    1.6    1.3    1.1    0.9    0.8    0.6    0.5
L1 VS L3:         1.2    1.8    1.8    1.5    1.2    1.1    0.9    0.8    0.6
L1 VS L4:         0.1    0.1    0.1    0.1    0.1    0.1    0.0    0.0    0.0
L2 VS L3:         0.4    0.7    0.7    0.5    0.5    0.3    0.3    0.2    0.2
L2 VS L4:         1.1    1.4    1.6    1.3    1.1    1.0    0.8    0.7    0.6
L3 VS L4:         1.3    1.6    1.9    1.5    1.3    1.2    1.0    0.9    0.7

RMS ANOMALOUS FH  [f" PART OF FH] AS % of F:

LAMBDA:  1        2.3    2.6    3.7    2.3    1.9    2.0    2.1    2.1    2.2
```

LAMBDA: 2	3.8	4.2	6.0	3.8	3.1	3.2	3.4	3.4	3.6
LAMBDA: 3	1.0	1.1	1.6	1.0	0.8	0.9	0.9	0.9	1.0
LAMBDA: 4	0.3	0.3	0.5	0.3	0.3	0.3	0.3	0.3	0.3

RMS DISPERSIVE FH [Delta-f-prime PART OF FH] AS % of F:

L1 VS L2:	4.8	5.5	7.5	4.8	3.9	4.0	4.3	4.2	4.5
L1 VS L3:	5.9	6.7	9.2	5.9	4.8	4.9	5.3	5.2	5.6
L1 VS L4:	0.2	0.2	0.3	0.2	0.2	0.2	0.2	0.2	0.2
L2 VS L3:	1.1	1.3	1.7	1.1	0.9	0.9	1.0	1.0	1.1
L2 VS L4:	4.6	5.1	7.2	4.6	3.7	3.8	4.1	4.1	4.3
L3 VS L4:	5.6	6.4	8.9	5.6	4.6	4.7	5.1	5.1	5.4

RMS ANOMALOUS ERRORS [CALC - OBS VALUE OF (F+ - F-)/2], IN % OF RMS F:

LAMBDA: 1	3.6	2.2	2.8	2.5	2.7	3.3	4.3	5.3	6.8
LAMBDA: 2	3.8	3.3	4.6	3.2	2.9	3.2	4.0	4.9	6.4
LAMBDA: 3	3.3	2.5	2.7	2.6	2.6	3.1	3.8	4.6	5.8
LAMBDA: 4	3.0	2.3	2.4	2.4	2.6	2.9	3.4	3.9	4.9

RMS DISPERSIVE ERRORS [CALC - OBS VALUE OF (F(i) - F(j))], IN % OF RMS F:

L1 VS L2:	4.7	3.5	4.7	3.7	3.6	4.2	5.7	6.6	8.5
L1 VS L3:	4.9	3.8	5.0	4.0	3.9	4.4	5.8	6.9	8.6
L1 VS L4:	3.4	2.6	2.5	2.3	2.3	2.9	4.4	5.1	6.8
L2 VS L3:	3.1	1.7	2.6	2.1	1.9	2.7	3.9	5.0	6.4
L2 VS L4:	4.3	3.6	4.4	3.6	3.4	3.9	4.9	5.9	7.5
L3 VS L4:	4.5	3.9	4.7	3.8	3.7	4.1	5.0	6.0	7.3

CORRELATED ANOMALOUS ERRORS BY WAVELENGTH (%):

LAMBDA: 1	2.2	2.0	2.6	2.1	2.0	2.0	2.3	2.7	3.1
LAMBDA: 2	3.7	3.3	4.4	3.4	3.3	3.4	3.9	4.4	5.0
LAMBDA: 3	1.0	0.9	1.2	0.9	0.9	0.9	1.0	1.2	1.4
LAMBDA: 4	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.4	0.4

RMS F BY WAVELENGTH:

LAMBDA: 1	747.0	1196.4	716.6	936.9	977.0	807.2	623.1	526.1	438.3
LAMBDA: 2	746.0	1207.4	713.2	936.0	977.1	804.2	620.4	523.6	437.3
LAMBDA: 3	745.0	1198.7	716.6	938.3	978.3	801.6	622.1	521.7	435.9
LAMBDA: 4	739.5	1183.2	716.0	931.6	969.6	802.9	620.5	519.3	432.7

PARAMETER SHIFTS FOR DERIV 2 : set 2

	SCALE FACTOR	OVERALL B
CURRENT VALUES:	1.0000	0.0000

	SITE	ATOM	OCCUP	X	Y	Z	B
CURRENT VALUES:	1	Se	0.6194	0.8300	0.1164	0.2313	26.4572

```
CURRENT VALUES:     2    Se      0.7246  0.4229  0.1245  0.1025   37.2234

CURRENT VALUES:     3    Se      0.7122  0.9424  0.3376  0.0763   38.6189

CURRENT VALUES:     4    Se      0.5266  0.8678  0.4974  0.0634   24.6016

CURRENT VALUES:     5    Se      0.6607  0.6800  0.3524  0.1621   51.2424

CURRENT VALUES:     6    Se      0.6652  0.4036  0.0844  0.1189   43.4833

CURRENT VALUES:     7    Se      0.6001  0.0449  0.2433  0.0284   60.0000

CURRENT VALUES:     8    Se      0.6859  0.9768  0.2254  0.0264   60.0000

CURRENT VALUES:     9    Se      0.4663  0.3883  0.2809  0.1287   42.9621

CURRENT VALUES:    10    Se      0.4500  0.2878  0.3281  0.1254   58.5963

CURRENT VALUES:    11    Se      0.4548  0.4085  0.2108  0.0973   55.1809

CURRENT VALUES:    12    Se      0.2751  0.3633  0.2230  0.0905   29.2657

CURRENT VALUES:    13    Se      0.4073  0.9114  0.2432  0.1395   60.0000

CURRENT VALUES:    14    Se      0.5146  0.8919  0.1059  0.0113   60.0000

CURRENT VALUES:    15    Se      0.4633  0.7665  0.1660  0.0285   60.0000


*************************************************************
*************************************************************


*** Summary of solutions and their relationships
 to each other and to check solution ***


-------------------------------------------------------
solution #          1 with overall quality =   64.60204
Derivative           2 with          15 sites.
Overall scale =    1.000000     and  overall b of   0.0000000E+00
 0.8299618      0.1164444      0.2312984      0.6194327       26.45717
 0.4228720      0.1245035      0.1025112      0.7245885       37.22337
 0.9423922      0.3375553      7.6348968E-02  0.7122003       38.61888
 0.8677520      0.4974284      6.3432172E-02  0.5266423       24.60162
 0.6799953      0.3523577      0.1620754      0.6606563       51.24240
 0.4035540      8.4412403E-02  0.1188622      0.6651538       43.48333
 4.4850446E-02  0.2433184      2.8359637E-02  0.6000957       60.00000
 0.9767606      0.2253628      2.6405444E-02  0.6859065       60.00000
 0.3882656      0.2808683      0.1287173      0.4662510       42.96208
 0.2878491      0.3280861      0.1253817      0.4499981       58.59629
 0.4085205      0.2108281      9.7288467E-02  0.4547521       55.18093
 0.3632535      0.2230348      9.0507492E-02  0.2751453       29.26567
```

0.9113981	0.2432471	0.1395056	0.4073112	60.00000
0.8918978	0.1058941	1.1270198E-02	0.5145602	60.00000
0.7664632	0.1660018	2.8497651E-02	0.4632579	60.00000

Best match of solution 1 -> solution 2:

-------- solution 1 -------- ----------solution 2 ------

site	x	y	z	site	x	y	z	DIST (A)

Derivative 2

1	0.830	0.116	0.231	14	0.830	0.116	0.231	0.05
2	0.423	0.125	0.103	9	0.424	0.125	0.103	0.07
3	0.942	0.338	0.076	6	0.942	0.339	0.076	0.19
4	0.868	0.497	0.063	7	0.867	0.497	0.064	0.06
5	0.680	0.352	0.162	15	0.681	0.351	0.162	0.13
6	0.404	0.084	0.119	8	0.404	0.086	0.120	0.21
7	0.045	0.243	0.028	4	0.041	0.243	0.028	0.27
8	0.977	0.225	0.026	3	0.977	0.225	0.025	0.21
9	0.388	0.281	0.129	10	0.385	0.280	0.128	0.25
10	0.288	0.328	0.125	13	0.285	0.327	0.124	0.43
11	0.409	0.211	0.097	12	0.408	0.206	0.098	0.47
12	0.363	0.223	0.091	11	0.361	0.224	0.090	0.16
13	0.911	0.243	0.140	5	0.908	0.246	0.140	0.34
14	0.892	0.106	0.011					
15	0.766	0.166	0.028	1	0.763	0.163	0.029	0.35

Comparison of this solution with check solution:

Number of sites in this solution matching check= 14

... and number not matching = 1

by derivative, this is...

Deriv	nsame	ndifferent
1	0	0
2	14	1
3	0	0
4	0	0

This solution is not the same as check soln

Contents	Index

The routine SOLVE: the core of automated structure determination by SOLVE

The SOLVE routine is an exceptionally powerful feature of this package that can find and evaluate the quality of heavy-atom sites in a MIR, SIR, or MIR-like dataset. The SOLVE routine treats MAD data almost exactly like MIR data, beginning with the output from MADMRG and MADBST.

Ordinarily SOLVE is called after SCALE_MAD and ANALYZE_MAD or SCALE_MIR and ANALYZE_MIR as part of automated structure determination. In this case you don't have to worry about all the keywords because the previous routines figure them out for you and write them to the script file solve_mad.script (or solve_mir.script).

You can, however, control much of what SOLVE does by setting keywords before running it. SOLVE can also be called using the solve_mad.script or solve_mir.script file written out by ANALYZE_MAD or an edited version of this file.

For MAD datasets, SOLVE uses a "compressed" form of MAD data that can be analyzed much more rapidly than the full n-wavelength data. This compressed dataset is generated by MADMRG in ANALYZE_MAD . The compressed dataset essentially consists of the SIR+anomalous scattering equivalent to the full MAD dataset. This dataset can be used to refine heavy atom parameters and generate native phases more quickly than a MAD dataset can. At the conclusion of SOLVE, phases are calculated with full Bayesian correlated MAD phasing.

The SOLVE routine operates by using a new version of HASSP to generate a few or many possible "seed" solutions for the anomalously scattering atoms in the structure. The heavy-atom parameters in each seed are first refined using the very fast refinement procedure in HEAVY (origin-removed patterson refinement). The refined seed is then used in self-difference Fouriers to suggest possible additional sites. A number of solutions are scored based on each seed, each solution being evaluated based on both the difference Patterson and a "free" difference Fourier. Additionally, the non-randomness of the native Fourier is used to judge the quality of a solution and to identify the correct hand of the structure if anomalous data is present. The figure of merit of phasing is the final scoring criterion.

If desired, a solution may be read in and evaluated directly with ANALYZE_SOLVE. Also, a solution may be read in and used as a seed in generating additional sites and a more complete solution with ADDSOLVE.

Using SOLVE is quite easy, particularly since ANALYZE_MAD or ANALYZE_MIR writes out a script file (usually solve_mad.script or solve_mir.script) that has everything you need to run SOLVE.

The only really non-obvious thing you need to know about running SOLVE on MAD data is that it requires 2 input data files. One is the compressed datafile from MADMRG, usually called "solve.data". The other is the full MAD dataset, usually called "mad_fpfm.scl". SOLVE uses "solve.data" for most of its analyses, then switches to the full MAD dataset at the very end.

The way you enter information on scattering factors is a little different in the SOLVE routine from the way it was entered in SCALE_MAD and ANALYZE_MAD . In the SOLVE routine you define atom types for each wavelength and specify the scattering factors for that atom type. Then you tell SOLVE what atom type goes with which wavelength. In SCALE_MAD, in contrast, you specified scattering factors directly for each wavelength. The reason for the difference is that SOLVE has to deal with both MAD and MIR data and defining atom types is a simple way to do that.

The solve_mad.script control file for MAD data

A sample SOLVE script file that will give you an idea of what you need to specify and what other things you can specify follows. This script is an edited version of a script file written out by the ANALYZE_MAD routine.

This script file is written out during automated SOLVE operation. You may wish to edit the one SOLVE has written out for you and use it if:

- you want to change how the core SOLVE routine solves your structure
- you want to use ANALYZE_SOLVE or ADDSOLVE to add sites to a solution you have found or to analyze a solution you have found

Sample MAD script file for SOLVE routine

```
!------------------solve_mad.script: solve a MAD problem----------------------
@solve.setup
 LOGFILE solve.logfile

 INFILE solve.data              !input file with MADMRG-compressed data
 MADFPFMFILE mad_fpfm.scl       !input file with full MAD dataset

 JSTD  1                        ! Lambda 1 is reference wavelength used in MADMRG
 IMADPHASE  1                   ! this is a MAD dataset, reference
                                !   wavelength is #1 (should match jstd)

 NNATF  1                       ! Pseudo-native F is column 1 of solve.data
 NNATS  2                       ! sigma is column 2

 ! Atom definitions with f' and f" values for the 3 wavelengths:
 NEWATOMTYPE LAM1
 AVAL  17.0006 5.8196 3.9731 4.3543
 BVAL  2.4098 .2726 15.2372 43.8163
 CVAL  2.8409
 FPRIMV  -1.6
 FPRPRV  3.4
 NEWATOMTYPE LAM2
 AVAL  17.0006 5.8196 3.9731 4.3543
 BVAL  2.4098 .2726 15.2372 43.8163
 CVAL  2.8409
 FPRIMV  -8.5
 FPRPRV  4.8
 NEWATOMTYPE LAM3
 AVAL  17.0006 5.8196 3.9731 4.3543
 BVAL  2.4098 .2726 15.2372 43.8163
 CVAL  2.8409
 FPRIMV  -9.85
 FPRPRV  2.86


 LAMBDA  1                          !  This is wavelength #1
 LABEL Wavelength  1 from MADMRG    !  label for lambda 1
```

```
NCOLFBAR  3                          ! Ncolfbar...ncolsdelf are column #'s
NCOLSFBAR  4                         ! in solve.data (MADMRG-compressed)
NCOLDELF  5                          ! datafile
NCOLSDELF  6
INPHASE
INANO
NOREFINESCALE                        ! Don't refine overall scale factor
                                     ! because this is MADMRG data

! Information for MADPHASE:
NCOLFPLUS  1                         ! these 4 column numbers refer to the
NCOLSIGPLUS  2                       ! full MAD datafile (mad_fpfm.scl)
NCOLFMINUS  3
NCOLSIGMINUS  4

! Heavy atoms for this wavelength:
ATOMNAME LAM1                        ! "LAM1" tells the program to use
OCCUPANCY  .1                        ! the scattering factors input above for
BVALUE  35.0                         ! LAM1
REFINEALL                            ! the occupancy and b values are guesses

LAMBDA  2
LABEL Wavelength  2 from MADMRG
NCOLFBAR  3
NCOLSFBAR  4
NCOLDELF  5
NCOLSDELF  6
INPHASE
INANO

! Information for MADPHASE:
NCOLFPLUS  5
NCOLSIGPLUS  6
NCOLFMINUS  7
NCOLSIGMINUS  8

! Heavy atoms for this derivative/wavelength:
ATOMNAME LAM2

LAMBDA  3
LABEL Wavelength  3 from MADMRG
NCOLFBAR  3
NCOLSFBAR  4
NCOLDELF  5
NCOLSDELF  6
INPHASE
INANO

! Information for MADPHASE:
NCOLFPLUS  9
NCOLSIGPLUS  10
```

```
 NCOLFMINUS  11
 NCOLSIGMINUS  12

 ! Heavy atoms for this derivative/wavelength:
 ATOMNAME LAM3

 ! Information for HASSP and SOLVE
 NCOLFHCOS  9                      ! column #s for <fh cos theta>
 NCOLFHSIN  10                     ! and <fh sin theta> in solve.data
 PATTFFTFILE patterson.patt        ! name of Bayesian patterson calculated
                                      !  by MADBST

 SOLVE                                ! run SOLVE
!-----------------------------------------------------------------------------
```

The solve_mir.script control file for MIR data

Using SOLVE is quite easy with MIR data too, particularly since ANALYZE_MIR writes out a script file that has everything you need to run SOLVE. A sample SOLVE script file that will give you an idea of what you need to specify and what other things you can specify follows. This script is an edited version of a script file written out by the ANALYZE_MIR routine.

This script file is written out during automated SOLVE operation. You may wish to edit the one SOLVE has written out for you and use it if:

- you want to change how the core SOLVE routine solves your structure
- you want to use ANALYZE_SOLVE or ADDSOLVE to add sites to a solution you have found or to analyze a solution you have found

Sample script file for SOLVE (MIR data)

```
!------------------solve_mir.script: solve an MIR problem----------------------
@solve.setup

LOGFILE solve.logfile

INFILE mir_fbar.scl                !input file with Fnat,sig, and
                                   !(fbar,sig,delano,sig) for each derivative..

NNATF 1                            ! Native F is column 1 of mir_fbar.scl
NNATS 2                            ! sigma is column 2

Derivative 1                       ! begin information about derivative 1
LABEL deriv 1 HG                   ! label for deriv 1
NCOLFBAR 3                         ! Ncolfbar...ncolsdelf are column #'s
NCOLSFBAR 4                        ! in mir_fbar.scl datafile
NCOLDELF 5
NCOLSDELF 6
```

```
INANO                              ! include anomalous differences

! Heavy atoms for this derivative:

ATOMNAME HG                        ! the atom type is "HG"
OCCUPANCY .1                       ! guess for occupancy
BVALUE 35.0                        ! guess for bvalue
REFINEALL                          ! refine everything that is reasonable

Derivative 2                       ! begin information about derivative 2
LABEL deriv 2 Iodine               ! label for deriv 2
NCOLFBAR 7                         ! Ncolfbar...ncolsdelf are column #'s
NCOLSFBAR 8                        ! in mir_fbar.scl datafile
NCOLDELF 9
NCOLSDELF 10
INANO                              ! include anomalous differences

ATOMNAME I-                        ! the atom type is "I-"
OCCUPANCY .1                       ! guess for occupancy
BVALUE 35.0                        ! guess for bvalue
REFINEALL                          ! refine everything that is reasonable

SOLVE                              ! run SOLVE
!-----------------------------------------------------------------------------
```

Keywords for the solve_mad.script and solve_mir.script files

There are a lot of keywords that can affect what SOLVE does. Ordinarily you do not have to worry about most of these because they are all set for you in ANALYZE_MAD. The solve_mad.script file written out by ANALYZE_MAD or the solve_mir.script file written by ANALYZE_MIR will have most of these keywords set for you. The keywords are listed here so that you can understand what they do and so that you can set them if you want to.

Most of these keywords can be specified at the beginning of automated data analysis to control what happens when SOLVE is called. For example, typing "ntopsolve 2" in the keywords before running SCALE_MAD and ANALYZE_MAD will affect SOLVE when it is called by restricting the number of solutions analyzed at the end of the routine to 2.

SOLVE treats MAD phasing and MIR phasing in almost exactly the same way except at the very end of the routine. Consequently "derivative" and "lambda" have the same meaning to SOLVE. You can enter information about lambda 1 by typing "lambda 1" or "derivative 1". The keywords that are specific to MAD phasing are listed at the top of the list.

Keywords that have a meaning for MAD data but not for MIR data:

```
INFILE  xxx.data    Principal input dorgbn-style file with compressed MAD data
                    from MADMRG and optional additional columns
                    of data. (usual file name = "solve.data").  This file
                    is usually produced by ANALYZE_MAD.


MADFPFMFILE yyy.scl Additional input file with (F+,sigma,F-
```

```
                    ,sigma) for each wavelength will be yyy.scl.
                    This file is used at the very end of SOLVE
                    for Bayesian correlated MAD phasing if the
                    keyword "bayes" is set in ANALYZE_MAD or the
                    keyword "imadphase n" is set in SOLVE.  All the
                    wavelengths have "inphase" specified for this
                    work. (DEFAULT="mad_fpfm.scl")

JSTD n              wavelength to be used as reference (default = lowest wavelength)

IMADPHASE n         This is a MAD dataset, n should match JSTD n

NOREFINESCALE      include this for all wavelengths usually because the
                    refinements in SOLVE are based on MADMRG output which
                    should not be further refined.
                    If xx is not recognized by SOLVE you need to
                    specify instead:
```

Keywords that apply to both MAD and MIR data:

```
NNATF n             column # in "infile" for native F (pseudo-native for MAD)
NNATS n             column # in "infile" for sigma of native f

 gotoderiv n        go to derivative (wavelength) n and get ready to read some
                    modifications of the parameters for this wavelength

 gotoatom n         go to the n'th atom in this wavelength/derivative  and get
                     ready to read some modifications of its parameters

 LABEL xxxxxx       label for this wavelength/derivative

 NCOLFBAR  n        column # for Fbar for this wavelength/derivative
                     For MAD data, this and the next three values are only
                       needed for the one wavelength defined by JSTD
                     For MIR data, they are needed for all derivatives
 NCOLSFBAR  n       column # for sigma of Fbar
 NCOLDELF  n        column # for delAno (if INANO is specified)
 NCOLSDELF  n       column # for sig of delAno
```

```
NCOLFHCOS  xx     column # in "infile" for estimated heavy atom structure
                    factor component along native structure
                    factor. (Output from MADBST for MAD data). This will be
                    used in calculation of heavy atom difference
                    Fouriers if ncolfhsin is also specified.
                    For MIR data, you can specify which derivative this applies
                    to by replacing the "1" in "ncolfhcos(1)" with another
                    derivative number. NCOLFHCOS is equivalent to NCOLFHCOS(1)

NCOLFHSIN  xx      column # in "infile" for estimated of heavy atom structure
                    factor perpendicular to native structure
                    factor. See ncolfhcos. NCOLFHSIN is equivalent to
                    NCOLFHSIN(1)

PATTFFTFILE xxxxxx   MAD data: Use previously calculated Patterson FFT
                      xxxxxx as the patterson map for the anomalously
scattering
                      atoms in this MAD structure.
                      MIR data: use patterson FFT xxxxx as patterson map for
                      derivative #1. PATTFFTFILE is equivalent to
                      PATTFFTFILE(1)(For other derivs, change the "1" to the
                      appropriate derivative number).
```

Also see all the commonly-used keywords for SOLVE.

Contents	Index

SOLVE scoring

SOLVE uses four scoring criteria to evaluate each solution:

- Analysis of difference Pattersons
- "Free" self-difference Fourier analyses
- Non-randomness test on native fourier
- Figure of merit of phasing

SOLVE uses the 10-30 seeds generated from the analyses of Patterson functions as trial solutions and scores each seed. From this set of scores, an average score and the standard deviation of this average score is obtained for each criteria. A Z-score (number of standard deviations above the mean) is then calculated for each trial solution for each criteria. The overall Z-score for a solution is the sum of the individual Z-scores, corrected for any large deviation among the scores.

Analysis of difference Pattersons

The analysis of the difference Pattersons for a MAD dataset are carried out in two steps. First all Patterson vectors that should result from all the sites in a derivative are calculated. Then the difference Patterson is examined at these positions and the relative occupancies of all the sites are refined so as to match the Patterson as closely as possible. The comparison of observed and calculated peak heights*1000/rms of the map is printed out along with an overall quality of the solution and evaluation of the fit to the difference Patterson.

"Free" self-difference Fourier analyses

The difference fouriers calculated here use all anomalously scattering atoms except just one atom to be examined (and all that are equivalent to it) to calculate native phases. These native phases are used to calculate a difference Fourier; the Fourier is examined at the coordinates of the test site and the peak height/rms of the map is noted. This is repeated for each anomalously-scattering atom in the structure and the heights are printed out.

Non-randomness test on native fourier

Most protein crystals have solvent regions (with low density and low variation in electron density) and protein regions (with high variation of density and moderate average density). This program examines the standard deviation (SD) of the local electron density in various locations around the unit cell and determines if the variation of this SD is high or low. If the phases are random, then you get a low

variation of the SD (the whole unit cell is rather uniform and looks like noise). If they are good, you get a high variation of the SD (part of the unit cell is smoother than another part). The program also analyzes the correlation of local r.m.s. density to see whether adjacent regions in the map tend to have similar r.m.s. variation. This is essentially a measure of how contiguous the regions of high and low variation are. This measure is reported back as the "correlation coefficient" (CC) in SOLVE. In versions 1.11 and higher the correlation coefficient is used in the actual scoring procedure (and not the standard deviation).

Figure of merit of phasing

The figure of merit of phasing the native structure proves to be a useful criteria in scoring a solution. Solutions with low figures of merit are unlikely to be correct, while those with a high figure of merit are likely to be right if the other criteria are also favorable.

Correction for uneven scores

SOLVE makes a correction on the score to reflect the discrepancy between the various scores. The effect is to score a solution that has a uniformly good score for all criteria higher than a solution with a very good score on just one criteria.

Back to RESOLVE table of contents

Quick Start

How to use RESOLVE

You can run RESOLVE right after SOLVE (using solve.mtz as input), or you can run it using an mtz-format file written by another program.

Usually if you have MAD/SAD/MIR data, you will want to (1) edit and run one of the standard SOLVE scripts(to solve your structure and calculate an initial map, and (2) edit and run the RESOLVE_BUILD script to density modify the map and build an atomic model. With fast model-building, the whole process only takes a few hours for a small protein.

If you have an MR model and FOBS data, you will want to run the RESOLVE_BUILD script to remove model bias and build a new atomic model.

If you have non-crystallographic symmetry, you will want to include that in your resolve scripts, either by specifying the NCS matrices (see ***Keywords*** for resolve) or by specifying the coordinates of some of the atoms in each monomor with a PDB file "ha.pdb" (this can be the heavy-atom file written by SOLVE for MIR/MAD/SAD data, or a file you create from some of the CA atoms in your starting PDB file for your MR solution).

Running RESOLVE right after SOLVE

It is really easy to run RESOLVE right after SOLVE. Just go to the directory where you ran SOLVE and type (or put in a command file):

```
#!/bin/csh
# Here is a very minimal script to run RESOLVE:
# Set CCP4 variables for symmetry information and
# for file handling:
setenv SYMOP /usr/local/lib/solve/symop.lib
setenv SYMINFO /usr/local/lib/solve/syminfo.lib
setenv CCP4_OPEN UNKNOWN
# Now run RESOLVE:
resolve<<EOD
solvent_content 0.4             ! your solvent content goes here.
!             Next line is protein sequence file
seq_file protein.seq
EOD
# Now "resolve.mtz" has the output amplitudes, weighted F, phases,
# figure of merit and HL-coeffs in columns labelled: FP PHIM FOMM HLAM HLBM HLCM HLDM
# Also resolve.pdb contains your model and the heavy-atom sites from SOLVE.
#
```

That's it! (Sure, there are a few other keywords, but usually you don't need them.) (More sample scripts are available too.)

- RESOLVE will read from "solve.mtz" and write out new structure factor amplitudes and phases to "resolve.mtz."
- You can use "resolve.mtz" in the same way as you use any mtz file in the CCP4 suite.
- The new phases are called PHIM and the figure of merit is FOMM (M is for modified).
- If there is a file called "ha.pdb" in your working directory, RESOLVE will assume it contains heavy-atom sites in PDB format and will try to find NCS in them and to apply it if it exists. It will ignore the NCS if the correlation is very low.
- RESOLVE will build a model of your structure into the density after doing statistical density modification. If you specify seq_file, then it expects a file with the sequence of your protein (1-letter code); if your protein has more than one unique chain, separate them with a line starting with the characters ">>>". No need to enter multiple copies of the same chain.
- Resolve will initially use only data in the resolution range where the starting figure-of-merit is significant, then will carry out phase extension.

Back to RESOLVE table of contents

RESOLVE output

The output from RESOLVE follows. The data are for the molecular replacement example using initiation factor 5A shown in the *example* section of the manual. You can run this data through yourself using the data and command files in the SOLVE/RESOLVE library installation directory, usually located at the place referred to by $SOLVEDIR/examples_resolve/

```
 ------------------------  COPYRIGHT NOTICE  --------------------------------
                   Los Alamos National Laboratory
 This program was prepared by the Regents of the University of California at
 Los Alamos National Laboratory (the University) under  Contract No.
 W-7405-ENG-36 with the U.S. Department of Energy (DOE).  The University has
 certain rights in the program pursuant to the contract and the program should
 not be copied or distributed outside your organization.  All rights in the
 program are reserved by the DOE and the University.  Neither the U.S.
 Government nor the University makes any warranty, express or implied, or
 assumes any liability or responsibility for the use of this software.

        ******************************************************
        *                --- Resolve---                      *
        *                                                    *
        *    Reciprocal-space solvent flattening and         *
        *    Maximum-likelihood density modification         *
        *                                                    *
        *     Type "resolvehelp" for on-line help            *
        *      or see "http://resolve.lanl.gov"              *
        ******************************************************

              (version 2.00 of 22-Mar-2001)
   Tom Terwilliger, Los Alamos National Laboratory, "terwilliger@LANL.gov"

>hklin sigmaa_2EIF.mtz
 Data to be read from mtz file:
 sigmaa_2EIF.mtz


>labin FP=FP FC=FC PHIC=PHIC FOM=WCMB
 Current value of "LABIN" is:
 labin FP=FP FC=FC PHIC=PHIC FOM=WCMB


>mask_cycles 10
 Number of mask/image cycles:          10

>minor_cycles 10
```

```
 Number of minor cycles per mask/image   cycle:          10

>solvent_content 0.60        !    solvent fraction
 Fraction of unit cell assumed to be solvent is:   0.6000000

>prime_and_switch
 Prime-and-switch phasing will be used (requires FC PHIC FP to be input).

>hklout resolve_ps.mtz
 Data to be written to mtz file:
 resolve_ps.mtz


 All done with inputs

       The date today is 12-apr-01.  Your license is good until 15-jul-07.

 User:   terwill              Logical Name: sigmaa_2EIF.mtz
 Status: READONLY   Filename: sigmaa_2EIF.mtz
 HEADER INFORMATION FROM INPUT MTZ FILE ON INDEX  1

 * Title:

 ...

 * Number of Columns =  11

 * Number of Reflections =  12490

 * Missing value set to NaN in input mtz  file

 * Column Labels :

 H K L FC PHIC FOMM FP SIGFP DELFWT FWT WCMB

 * Column Types :

 H H H F P W F Q F F W

 * Cell Dimensions :

   113.95  113.95   32.47   90.00   90.00   90.00

 *  Resolution Range :

      0.00262     0.22672      (   19.537 -     2.100 A )

 * Sort Order :

      1     2     3     0     0

 * Space group = I4  (number    79)
```

```
 * Input Program Labels :

 H K L FP PHIB FOM HLA HLB HLC HLD FC PHIC SIGFP FWT

 * Input File Labels :

 H K L FC PHIC FOMM FP SIGFP DELFWT FWT WCMB

 * Lookup Table : the number indicates the input column no.
 * Array element n corresponds to the nth program label

     0    0    0    7    0   11    0    0    0    0    4    5    0    0

 FP taken from column           7
 FOM taken from column          11
 Reading in FC and PHIC from columns            4           5
 FOM will be applied to PHIC
 Total of        12490 reflections read from file
 Mean FOM of input data =   0.2678500
 Adding F000 term (0.0) to this list
 Closed mtz file
 Space group is          79

 FORMATTED      OLD     file opened on unit  11
 Logical name: SYMOP, Full name: /usr/local/lib/solve/symop.lib

   Centric Zone   1 Reflections of Type  hk0

 Cycles of mask generation:          10
 Sub-cycles per mask cycle:          10


 Warning -- Total of            1 centric phases off by >   0.5000000
  degrees

 Expected I corrected for centering by factor of            2
 Results of wilson scaling:
 Scale on I = 1.89
 B-value = 9.66


 Setting up small boxes for cross-validation
Subdivisions along X Y Z:    6   3   1 for total of   18 boxes.

 Results of wilson scaling of model Fc to Fo :
 Scale on I to apply to Fc = 3.02
 B-value to apply to Fc= .128E-01


 Estimates of overall B and scale for fitting Model map to Observed intensities
 Overall B =     25.6 Corresponding to rmsd of about      0.6 A.
```

Overall scale, corresponding to fraction of atoms modelled with this error = 0.28

CC of Observed, Model Intensities vs resolution
(CALC is fit to the observed CC with a pseudo-B)

ACENTRIC			CENTRIC			
N	CC	CALC	N	CC	CALC	DMIN
20	0.42	0.27	13	-0.36	0.27	14.00
53	0.35	0.27	23	0.20	0.27	10.11
111	0.33	0.26	37	0.16	0.26	7.78
196	0.10	0.25	44	0.13	0.25	6.22
211	0.30	0.23	41	0.41	0.23	5.44
365	0.39	0.22	67	0.01	0.22	4.67
457	0.39	0.20	63	0.27	0.20	4.12
511	0.24	0.19	66	0.25	0.19	3.73
606	0.10	0.17	71	0.29	0.17	3.42
621	0.13	0.16	64	0.22	0.16	3.19
817	0.13	0.14	82	0.23	0.14	2.96
738	0.10	0.13	64	0.14	0.13	2.80
893	0.06	0.12	83	0.17	0.12	2.64
1123	0.03	0.11	92	-0.02	0.11	2.49
1484	0.06	0.09	114	-0.07	0.09	2.33
1395	0.09	0.08	99	0.07	0.08	2.22
1751	0.09	0.07	115	0.15	0.07	2.10

Scaling model reflections with overall scale and B

Resolution cutoff recommended based on CC of FP and FC is: 2.100175

Overall R-factor for FC vs FP: 0.45 for 12491 reflections

Reset-resolution with adjusted_resolution of 2.10 and NREFL = 12491

Total of 0 reflections < 2.10 tossed, leaving a total of 12491

Getting starting map using current phases

Mask cycle 1

big_cycle
get_mask_wang
New Wang averaging radius = 11.57

Mean +/- SD of density in protein region : 0.00 +/- 0.30
Mean +/- SD of density in solvent region : 0.00 +/- 0.13

Plot of probability that a grid point is part of protein region
vs percentiles of grid points

All points to the left of the "+" signs are in solvent masked region
those to right are in protein masked region.

The values of p(protein) should change from low to high approximately at the value of the fraction of solvent indicated by the "+" signs.
The sharper the transition the better.

Note: the mask is only used to make an estimate of the p(protein)

The values of p(protein) are used to weight the contribution of each grid point to the likelihood of the map:

p(rho) = p(rho|protein) p(protein) + p(rho|solvent) (1-p(protein))

This says that the probability that we would observe the value "rho" of electron density at this point is the probability that we would observe "rho" if this were really protein times the probability that this is protein, plus the probability that we would observe "rho" if it were really solvent, times the probability that it is solvent.

```
                              Probability that grid points are in protein region


                        1.0  ........................................xxxxxxx
                             .                            +         xxx      .
                             .                            +       xx         .
                             .                            +     xx           .
                             .                            +   xx             .
                             .                            + xx               .
                             .                            +x                 .
               p(protein)    .                           xx                  .
                        0.5  .                          x +                  .
                             .                         x  +                  .
                             .                        x   +                  .
                             .                       x    +                  .
                             .                     xx     +                  .
                             .                    x       +                  .
                             .                  xxx       +                  .
                        0.0  .xxxxxxxxxxxxxxxxxx...........+...................

                             0       20        40        60        80       100

                                             Percentile of grid points
-------------------------------------------------------------------------------
```

Range of P(protein) and percentiles
used for histograms of protein and solvent:

P(protein) Percentile

```
             Low      High              Low      High
Protein:    0.85      1.00              80.      100.
Solvent:    0.00      0.12               0.       44.


Minor cycle    1



Plot of Observed (o) and model (x) electron density distributions for protein region
where the model distribution is given by,  p_model(beta*(rho+offset)) = p_ideal(rho)
and then convoluted with a gaussian with width of sigma
where sigma, offset and beta are given below under "Error estimate."


                        0.05.................................................
                            .                   .                           .
                            .                   .                           .
                            .                xxoo                           .
                            .               xooxo                           .
                            .              xo   xo                          .
                            .             xx    .xo                         .
               p(rho)       .             x     . xo                        .
                            .            xo     .  x                        .
                            .            x      .  xx                       .
                            .          ox       .    xx                     .
                            .          x        .     xxx                   .
                            .         ox        .      ooxxxx               .
                            .        ox         .        oooxxxx            .
                            .      oxx          .             ooxxxxooo     .
                        0.0 xxxxxxxx...............................xxxxxxxxxx

                            -2        -1         0         1         2        3

                                 normalized rho (0 = mean of solvent region)
-------------------------------------------------------------------------------




Plot of Observed (o) and model (x)
electron density distributions for solvent region


                        0.09.................................................
                            .                   .                           .
                            .                   .                           .
                            .                  oxo                          .
                            .                  x.xx                         .
                            .                  x. x                         .
                            .                 x . o                         .
               p(rho)       .                 x .  x                        .
```

```
                              .                x  .  x                      .
                              .                x  .  x                      .
                              .               oo  .  ox                     .
                              .               x   .   ox                    .
                              .              xx   .    x                    .
                              .              x    .    ox                   .
                              .            xx     .     xx                  .
                         0.0  xxxxxxxxxxxxxx.............xxxxxxxxxxxxxxxxxxxxx

                             -2        -1         0         1         2        3

                                  normalized rho (0 = mean of solvent region)
-----------------------------------------------------------------------------
```

Error estimate for map on this cycle

The estimated error in this map is 0.11 based on an analysis of the electron density distributions in the protein and solvent regions with rms values of 0.31 and 0.13 respectively.
The value of the scale factor beta relating idealized density distributions P_ideal(rho) to observed ones is:

obs_rho = beta * (ideal_rho + offset) + sigma

where beta = 0.63, offset is -0.09 and sigma is a random variable with rms value of 0.11

Mean starting figure of merit this cycle

Overall	Centric	Acentric
0.27	0.27	0.27
12490	1138	11352

Input phase probabilities weighted by factor of 0.00

Mean normalized structure factor changes this cycle

Acentric	Centric
0.24	0.27
11350	1136

New figure of merit () of phasing based on:
(1) experimental phase information, and
(2) likelihood of the resulting map

These are the 2 sources of phase information used in resolve. The phase

information from them should be correlated and the phase information from the map should increase during maximum- likelihood density modification.

Correlation between prior and map phase information is measured by
, the mean cosine of the phase difference.

Best estimate of true figure of merit of map-likelihood phasing is
ratio of correlation to of prior information

Fraction of phase information from prior is estimated from of prior, map

Acentric reflections only:

DMIN	--Figure of merit-- Prior	Map	Total	CC Prior-Map	Fraction from Prior	Total (corrected)	N
ALL:	0.33	0.40	0.40	0.72	0.00	0.40	11352
6.0	0.45	0.73	0.73	0.76	0.00	0.73	431
3.8	0.50	0.80	0.80	0.82	0.00	0.80	1473
3.0	0.36	0.62	0.62	0.75	0.00	0.62	1892
2.6	0.30	0.36	0.36	0.72	0.00	0.36	1932
2.3	0.26	0.22	0.22	0.69	0.00	0.22	3459
2.1	0.31	0.17	0.17	0.65	0.00	0.17	2165
WGT:	0.00	1.00					

Centric reflections only:

DMIN	--Figure of merit-- Prior	Map	Total	CC Prior-Map	Fraction from Prior	Total (corrected)	N
ALL:	0.27	0.46	0.46	0.56	0.00	0.46	1138
6.0	0.34	0.65	0.65	0.68	0.00	0.65	130
3.8	0.38	0.67	0.67	0.54	0.00	0.67	222
3.0	0.28	0.59	0.59	0.55	0.00	0.59	197
2.6	0.24	0.40	0.40	0.56	0.00	0.40	179
2.3	0.16	0.26	0.26	0.52	0.00	0.26	262
2.1	0.27	0.25	0.25	0.57	0.00	0.25	148
WGT:	0.00	1.00					

All reflections:

DMIN	--Figure of merit-- Prior	Map	Total	CC Prior-Map	Fraction from Prior	Total (corrected)	N
ALL:	0.33	0.40	0.40	0.70	0.00	0.40	12490
6.0	0.42	0.71	0.71	0.74	0.00	0.71	561
3.8	0.49	0.78	0.78	0.78	0.00	0.78	1695
3.0	0.35	0.62	0.62	0.73	0.00	0.62	2089
2.6	0.29	0.36	0.36	0.70	0.00	0.36	2111
2.3	0.25	0.23	0.23	0.68	0.00	0.23	3721
2.1	0.31	0.17	0.17	0.65	0.00	0.17	2313
WGT:	0.00	1.00					

```
*********************************************************************
*                                                                   *
*     CORRECTED OVERALL FIGURE OF MERIT OF PHASING:   0.40          *
*     BIAS RATIO:  5.39                                             *
*                                                                   *
*     Bias ratio is given by /(FOMA*FOMB),           *
*     where phiA and FOMA are phase and  from prior (model or     *
*     experiment) and phiB and FOMB are from map.                   *
*     Bias ratio is about 1 if phiA and phiB are independent,       *
*                         >1 if phiB is biased by phiA,             *
*                         <1 if FOMA or FOMB are overestimated      *
*                                                                   *
*     If BIAS RATIO <1, CORRECTED FOM = ESTIMATED FOM * BIAS RATIO  *
*********************************************************************
```

Cumulative phase change from start to end of this cycle []

DMIN	-----acentric-------- N	N	N	--------centric-------			---------all---------		
0.0	0.72	0.82	11352.	0.56	0.74	1138.	0.70	0.81	12490.
6.0	0.76	0.81	431.	0.67	0.78	130.	0.74	0.80	561.
3.8	0.82	0.85	1473.	0.54	0.71	222.	0.78	0.84	1695.
3.0	0.75	0.81	1892.	0.55	0.71	197.	0.73	0.80	2089.
2.6	0.72	0.81	1932.	0.56	0.73	179.	0.70	0.80	2111.
2.3	0.69	0.81	3459.	0.52	0.75	262.	0.68	0.80	3721.
2.1	0.65	0.79	2165.	0.57	0.83	148.	0.65	0.79	2313.

End of first cycle

```
Minor cycle    2



     Error estimate for map on this cycle

The estimated error in this map is    0.10 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.44 and    0.13 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.00, offset is   -0.07   and sigma is a
random variable with rms value of    0.10


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.40       0.46       0.40
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.17         0.17
     11351         1137


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.11 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.48 and    0.13 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.05, offset is   -0.08   and sigma is a
random variable with rms value of    0.11


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
```

```
      0.45       0.52       0.45
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.12         0.11
     11351         1137


Minor cycle    4



     Error estimate for map on this cycle

The estimated error in this map is    0.11 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.48 and    0.13 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.05, offset is   -0.09   and sigma is a
random variable with rms value of    0.11


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.45       0.52       0.45
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.10         0.08
     11351         1137


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.11 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.49 and    0.12 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:
```

```
     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.06, offset is   -0.10   and sigma is a
random variable with rms value of    0.11


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.46       0.53       0.45
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.08         0.07
     11351         1137


Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.11 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.49 and    0.12 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.06, offset is   -0.10   and sigma is a
random variable with rms value of    0.11


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.46       0.53       0.46
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.07         0.07
     11351         1136


Minor cycle    7
```

```
     Error estimate for map on this cycle

The estimated error in this map is    0.11 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.49 and    0.12 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.06, offset is   -0.11   and sigma is a
random variable with rms value of    0.11


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.47       0.54       0.46
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.06         0.06
     11351         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.11 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.49 and    0.12 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.06, offset is   -0.11   and sigma is a
random variable with rms value of    0.11


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.47       0.54       0.46
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
```

```
      0.06         0.06
     11351         1136


Minor cycle    9



     Error estimate for map on this cycle

The estimated error in this map is    0.11 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.50 and    0.11 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.06, offset is   -0.11   and sigma is a
random variable with rms value of    0.11


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.47       0.54       0.46
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.05         0.05
     11351         1136


Minor cycle   10



     Error estimate for map on this cycle

The estimated error in this map is    0.11 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.50 and    0.11 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.06, offset is   -0.11   and sigma is a
random variable with rms value of    0.11
```

```
Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.47       0.54       0.46
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.04         0.05
     11351         1136


                        Mask cycle            2

big_cycle
get_mask_wang
New Wang averaging radius =  6.47

Mean +/- SD of density in protein region :   0.01 +/-   0.50
Mean +/- SD of density in solvent region :   0.00 +/-   0.10


Minor cycle    1



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.50 and    0.10 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.07, offset is   -0.12   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.47       0.55       0.46
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.05         0.05
     11351         1136
```

```
Minor cycle    2



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.50 and    0.10 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.07, offset is   -0.15   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.55       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.04         0.04
     11351         1136


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.50 and    0.09 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.07, offset is   -0.16   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352
```

```
Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.03         0.03
     11351         1136


Minor cycle    4



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.09 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.07, offset is   -0.17   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.55       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.03         0.03
     11351         1136


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.09 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma
```

```
where beta =    1.07, offset is   -0.17   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.03         0.03
     11351         1136


Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.09 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.07, offset is   -0.18   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.02         0.03
     11351         1136


Minor cycle    7



     Error estimate for map on this cycle
```

```
The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.09 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.18   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.02         0.03
     11351         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.09 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.18   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.02         0.02
     11351         1136
```

```
Minor cycle    9



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.09 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.19   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.02         0.02
     11351         1136


Minor cycle   10



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.09 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.18   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
```

```
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.02         0.02
     11351         1136


                        Mask cycle            3

big_cycle
get_mask_wang
New Wang averaging radius =  6.37

Mean +/- SD of density in protein region :   0.00 +/-   0.51
Mean +/- SD of density in solvent region :   0.00 +/-   0.08


Minor cycle    1



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.20   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.02         0.02
     11351         1136


Minor cycle    2
```

```
     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.20   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.02         0.02
     11351         1136


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.20   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352
```

```
Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.02         0.02
     11351         1136


Minor cycle    4



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.20   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.20   and sigma is a
random variable with rms value of    0.12
```

```
Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136


Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.20   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136


Minor cycle    7



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
```

```
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136
```

```
Minor cycle    9



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136


Minor cycle   10



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352
```

```
Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136


                       Mask cycle            4

big_cycle
get_mask_wang
New Wang averaging radius =  6.35

Mean +/- SD of density in protein region :   0.00 +/-   0.51
Mean +/- SD of density in solvent region :   0.00 +/-   0.08


Minor cycle    1



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1135


Minor cycle    2
```

```
     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
```

```
      0.01         0.02
     11351         1136


Minor cycle    4



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.21   and sigma is a
random variable with rms value of    0.12
```

```
Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.02
     11351         1136


Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    7



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
```

```
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    9
```

```
     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle   10



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.08 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.21   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.56       0.47
     12490       1138      11352
```

```
Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


                        Mask cycle            5

big_cycle
get_mask_wang
New Wang averaging radius =  6.34

Mean +/- SD of density in protein region :   0.00 +/-   0.51
Mean +/- SD of density in solvent region :   0.00 +/-   0.07


Minor cycle    1



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11350         1136


Minor cycle    2



     Error estimate for map on this cycle
```

```
The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136
```

```
Minor cycle    4



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
```

```
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    7



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:
```

```
     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    9
```

```
     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle   10



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
```

```
      0.01         0.01
     11351         1136


                        Mask cycle           6

big_cycle
get_mask_wang
New Wang averaging radius =  6.33

Mean +/- SD of density in protein region :   0.00 +/-   0.51
Mean +/- SD of density in solvent region :   0.00 +/-   0.07


Minor cycle    1



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    2



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
```

```
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136
```

```
Minor cycle    4



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352
```

```
Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11350         1135


Minor cycle    7



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma
```

```
where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11350         1136


Minor cycle    9



     Error estimate for map on this cycle
```

```
The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11350         1136


Minor cycle   10



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11350         1136
```

```
                        Mask cycle           7

big_cycle
get_mask_wang
New Wang averaging radius =  6.32

Mean +/- SD of density in protein region :   0.00 +/-   0.51
Mean +/- SD of density in solvent region :   0.00 +/-   0.07


Minor cycle    1



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    2



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
```

```
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    4
```

```
     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352
```

```
Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    7



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12
```

```
Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.01         0.01
     11351         1136


Minor cycle    9



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
```

```
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.08, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.01
     11351         1136


Minor cycle   10



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11351         1136
```

```
                        Mask cycle            8

big_cycle
get_mask_wang
New Wang averaging radius =  6.32

Mean +/- SD of density in protein region :   0.00 +/-   0.51
Mean +/- SD of density in solvent region :   0.00 +/-   0.07


Minor cycle    1



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11351         1136


Minor cycle    2



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:
```

```
     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11351         1136


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11351         1136


Minor cycle    4
```

```
     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11351         1136


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
```

```
      0.00         0.00
     11350         1136


Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    7



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12
```

```
Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    9



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
```

```
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle   10



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


                        Mask cycle            9
```

```
big_cycle
get_mask_wang
New Wang averaging radius =  6.32

Mean +/- SD of density in protein region :   0.00 +/-   0.51
Mean +/- SD of density in solvent region :   0.00 +/-   0.07


Minor cycle    1



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    2



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma
```

```
where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    4



     Error estimate for map on this cycle
```

```
The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136
```

```
Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    7



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
```

```
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    9



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:
```

```
     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle   10



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.22   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


                         Mask cycle           10

big_cycle
get_mask_wang
```

```
New Wang averaging radius =  6.31

Mean +/- SD of density in protein region :   0.00 +/-   0.51
Mean +/- SD of density in solvent region :   0.00 +/-   0.07



-------------------------------------------------------------------------------

Plot of probability that a grid point is part of protein region
vs percentiles of grid points

All points to the left of the "+" signs are in solvent masked region
those to right are in protein masked region.

The values of p(protein) should change from low to high approximately at the value
of the fraction of solvent indicated by the "+" signs.
The sharper the transition the better.

Note: the mask is only used to make an estimate of the p(protein)

The values of p(protein) are used to weight the contribution of each grid
 point to the likelihood of the map:

 p(rho) = p(rho|protein) p(protein) +  p(rho|solvent) (1-p(protein))

This says that the probability that we would observe  the value "rho"  of electron
density at this point is the probability that we would observe "rho" if this were
really protein times the probability that this is  protein, plus the probability
that we would observe "rho" if it were really solvent, times the probability that
it is solvent.

                             Probability that grid points are in protein region


                      1.0  ....................................xxxxxxxxxxxxxx
                           .                            +    x             .
                           .                            +   x              .
                           .                            +  x               .
                           .                            + x                .
                           .                            +                  .
                           .                            +x                 .
             p(protein)    .                            +                  .
                      0.5  .                            x                  .
                           .                           x+                  .
                           .                            +                  .
                           .                          x +                  .
                           .                         x  +                  .
                           .                        x   +                  .
                           .                      xx    +                  .
                      0.0  .xxxxxxxxxxxxxxxxxxxxxxx......+...................
```

```
                              0       20        40        60        80       100

                                            Percentile of grid points
-------------------------------------------------------------------------------



Range of P(protein)  and percentiles
used for histograms of protein and solvent:

             P(protein)                 Percentile
            Low      High             Low      High
Protein:    0.89      1.00             70.      100.
Solvent:    0.00      0.13              0.       52.


Minor cycle    1



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    2



     Error estimate for map on this cycle
```

```
The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    3



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136
```

```
Minor cycle    4



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    5



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
```

```
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11350         1136


Minor cycle    6



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11349         1136


Minor cycle    7



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:
```

```
     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11349         1136


Minor cycle    8



     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11349         1136


Minor cycle    9
```

```
     Error estimate for map on this cycle

The estimated error in this map is    0.12 based on an analysis of the electron
density distributions in the protein and solvent regions with rms
values of    0.51 and    0.07 respectively.
The value of the scale factor beta relating idealized density distributions
P_ideal(rho) to observed ones is:

     obs_rho = beta * (ideal_rho + offset) + sigma

where beta =    1.09, offset is   -0.23   and sigma is a
random variable with rms value of    0.12


Mean starting figure of merit this cycle
   Overall    Centric    Acentric
      0.48       0.57       0.47
     12490       1138      11352


Mean normalized structure factor changes this cycle
   Acentric       Centric
      0.00         0.00
     11349         1136


Minor cycle   10
```

```
                         Final cycle



Plot of Observed (o) and model (x) electron density distributions for protein region
where the model distribution is given by,  p_model(beta*(rho+offset)) = p_ideal(rho)
and then convoluted with a gaussian with width of sigma
where sigma, offset and beta are given below under "Error estimate."


                      0.03.................................................
                          .                   .                           .
                          .                   .                           .
                          .          xxxxxxoo .                           .
                          .         ox ooooX oo                           .
                          .        ox    o  xx.                           .
                          .       ox         xxoo                         .
              p(rho)      .       x           xxo                         .
                          .      xx           . xo                        .
```

```
                          .    o x            .  xx                       .
                          .    xx             .   oxxx                    .
                          .    x              .    oooxxxxxx              .
                          .   xo              .      oooooooxxxxxxx       .
                          . xx                .               oooooxxxxxxo  .
                          xxo                 .                       xxxxo
                     0.0  o...............................................x

                         -2        -1         0         1         2        3

                              normalized rho (0 = mean of solvent region)
-------------------------------------------------------------------------------
```

Plot of Observed (o) and model (x)
electron density distributions for solvent region

```
                     0.18...............................................
                          .                  .                          .
                          .                  .                          .
                          .                  oo                         .
                          .                  .                          .
                          .                  .                          .
                          .                 o.o                         .
              p(rho)      .                  .                          .
                          .                 o. o                        .
                          .                  .                          .
                          .                 xxxo                        .
                          .                x . xx                       .
                          .               x  .  xx                      .
                          .              xx o .    x                    .
                          .            xxx o  .  o  xx                  .
                     0.0  xxxxxxxxxxxxxoooo.......oooooxxxxxxxxxxxxxxxxxxxxx

                         -2        -1         0         1         2        3

                              normalized rho (0 = mean of solvent region)
-------------------------------------------------------------------------------
```

Error estimate for map on this cycle

The estimated error in this map is 0.12 based on an analysis of the electron density distributions in the protein and solvent regions with rms values of 0.51 and 0.07 respectively.
The value of the scale factor beta relating idealized density distributions P_ideal(rho) to observed ones is:

```
     obs_rho = beta * (ideal_rho + offset) + sigma

 where beta =    1.09, offset is   -0.23   and sigma is a
 random variable with rms value of    0.12


 Mean starting figure of merit this cycle
    Overall    Centric    Acentric
       0.48       0.57       0.47
      12490       1138      11352


Input phase probabilities weighted by factor of   0.00


 Mean normalized structure factor changes this cycle
    Acentric       Centric
       0.00         0.00
      11349         1136



 New figure of merit () of phasing based on:
      (1) experimental phase information, and
      (2) likelihood of the resulting map

 These are the 2 sources of phase information used in resolve.  The phase
 information from them should be correlated and the phase information from
 the map should increase during maximum- likelihood density modification.

 Correlation between prior and map phase information is measured by
 , the mean cosine of the phase difference.

 Best estimate of true figure of merit of map-likelihood phasing is
 ratio of correlation to  of prior information

 Fraction of phase information from prior is estimated from  of prior, map
```

Acentric reflections only:

DMIN	--Figure of merit-- Prior	Map	Total	CC Prior-Map	Fraction from Prior	Total (corrected)	N
ALL:	0.34	0.47	0.47	0.23	0.00	0.47	11352
6.0	0.45	0.79	0.79	0.46	0.00	0.79	431
3.8	0.51	0.83	0.83	0.42	0.00	0.82	1473

3.0	0.37	0.74	0.74	0.27	0.00	0.74	1892
2.6	0.30	0.49	0.49	0.21	0.00	0.49	1932
2.3	0.27	0.31	0.31	0.16	0.00	0.31	3459
2.1	0.32	0.19	0.19	0.13	0.00	0.19	2165
WGT:	0.00	1.00					

Centric reflections only:

	--Figure of merit--			CC	Fraction	Total	
DMIN	Prior	Map	Total	Prior-Map	from Prior	(corrected)	N
ALL:	0.27	0.57	0.57	0.24	0.00	0.57	1138
6.0	0.34	0.68	0.68	0.38	0.00	0.68	130
3.8	0.38	0.73	0.73	0.27	0.00	0.70	222
3.0	0.28	0.69	0.69	0.30	0.00	0.69	197
2.6	0.24	0.56	0.56	0.16	0.00	0.56	179
2.3	0.16	0.42	0.42	0.22	0.00	0.42	262
2.1	0.27	0.34	0.34	0.15	0.00	0.34	148
WGT:	0.00	1.00					

All reflections:

	--Figure of merit--			CC	Fraction	Total	
DMIN	Prior	Map	Total	Prior-Map	from Prior	(corrected)	N
ALL:	0.33	0.48	0.48	0.23	0.00	0.48	12490
6.0	0.42	0.76	0.76	0.44	0.00	0.76	561
3.8	0.49	0.81	0.81	0.40	0.00	0.81	1695
3.0	0.36	0.73	0.73	0.27	0.00	0.73	2089
2.6	0.30	0.50	0.50	0.20	0.00	0.50	2111
2.3	0.26	0.32	0.32	0.16	0.00	0.32	3721
2.1	0.31	0.20	0.20	0.13	0.00	0.20	2313
WGT:	0.00	1.00					

```
*********************************************************************
*                                                                   *
*     CORRECTED OVERALL FIGURE OF MERIT OF PHASING:   0.48          *
*     BIAS RATIO:  1.42                                             *
```

```
*                                                                    *
*     Bias ratio is given by /(FOMA*FOMB),           *
*     where phiA and FOMA are phase and  from prior (model or     *
*     experiment) and phiB and FOMB are from map.                    *
*     Bias ratio is about 1 if phiA and phiB are independent,        *
*                         >1 if phiB is biased by phiA,              *
*                         <1 if FOMA or FOMB are overestimated       *
*                                                                    *
*     If BIAS RATIO <1, CORRECTED FOM = ESTIMATED FOM * BIAS RATIO   *
**********************************************************************


Cumulative phase change from start to end of this cycle []

DMIN  -----acentric--------  --------centric-------  ---------all---------
           N        N        N
 0.0    0.25    0.36  11352.   0.25    0.40   1138.   0.25    0.36  12490.
 6.0    0.47    0.50    431.   0.39    0.48    130.   0.45    0.50    561.
 3.8    0.43    0.48   1473.   0.27    0.39    222.   0.41    0.47   1695.
 3.0    0.29    0.34   1892.   0.30    0.43    197.   0.29    0.35   2089.
 2.6    0.23    0.32   1932.   0.16    0.36    179.   0.22    0.32   2111.
 2.3    0.18    0.29   3459.   0.22    0.34    262.   0.18    0.29   3721.
 2.1    0.15    0.23   2165.   0.15    0.43    148.   0.15    0.25   2313.

             All done .. writing out new phases

 (Q)QOPEN allocated #  1
User:   terwill              Logical Name: resolve_ps.mtz
Status: UNKNOWN    Filename: resolve_ps.mtz
HEADER INFORMATION FOR OUTPUT MTZ FILE ON INDEX  3

* Title:



* Number of Columns =  10

* Number of Reflections =  12490

* Missing value set to NaN in output mtz  file

* Column Labels :

H K L FP PHIM FOMM HLAM HLBM HLCM HLDM

* Column Types :

H H H F P W A A A A

* Cell Dimensions :

  113.95  113.95   32.47   90.00   90.00   90.00
```

```
*  Resolution Range :

     0.00262     0.22672      (   19.542 -     2.100 A )

* There is no sort order recorded in the MTZ header

* Space group = I4  (number    79)



*********************************************************************
*                                                                   *
*     NOTE:  HIGH RESOLUTION LIMIT ADJUSTED TO   2.10 A             *
*   (Use the keyword resolution to set a different value)           *
*                                                                   *
*********************************************************************
```

Back to table of contents

RESOLVE FAQ page

Where can I find all the questions and answers that have been sent to the SOLVE/RESOLVE newsgroup?"

- *You can find them archived at Jiansheng Jiang's very nice site at http://asdp.bnl.gov. On that site, click the blue button "BBnML" (Bulletin Boards and Mailing Lists) and then click "solve" under the Archive column.*

What do I do if RESOLVE says:

- *Sorry, the protein region has no volume?* This means that the solvent content is 0 or 1. Try a value in between.
- *Sorry, this is an invalid access code* You need to check out your solve2.access file. If SOLVE runs, so should RESOLVE.
- *Segmentation fault (core dump, program never starts)* There can be several causes. On linux and SGI this can happen if your system does not match the system the binaries were compiled on. Solution: email terwilliger@lanl.gov and I'll send you instructions on how to compile on your own machine. Also on an SGI: This can happen if your machine doesn't have enough memory allocated to you. To get around this (at least on an SGI), use the command: "unlimit" . You can put this as one line in your .cshrc file if you use csh. The same works on an Alpha too. You may also need to increase your swap space in some cases.
- *Segmentation fault (core dump after "Writing peaks to.."), same fix as above: use the command: "unlimit"; may need to increase your swap space or use "small" version.*
- *killed This happens if you run the too big a version of resolve on an SGI or if you're already running a big job. See the above command "unlimit" to fix it or use a smaller version.*
- */sbin/loader: Fatal Error: Program datasize exceeds process datasize limit. This is an error message you get with an Alpha when you don't have enough memory allocated to yourself. To get around this, edit your ".cshrc" or similar file to include the line "unlimit" or use the "small" version.*
- *Sorry, nothing in bins of density? The electron density map you input probably has no reflections in the resolution range or is all zeros.*
- *Sorry, cannot add any more onto the "LABIN" line (max 132 characters). RESOLVE cannot use more than 132 characters in a labin line. Sorry!*
- *Sorry: uninterpretable value or keyword. RESOLVE could not parse the input. It stops when this happens. Usually a keyword it does not recognize was input. Note that RESOLVE does not recognize partial keywords, you have to type the whole word in.*
- *Sorry, need to make n_map_max at least [some number]. The version of RESOLVE you have is*

not big enough for your dataset. Try the "giant" (resolve_giant) or even the "huge" (resolve_huge) version. If that does not work, contact the author for a bigger version.

- *sorry, this map has no density. The electron density map you input probably has no reflections in the resolution range or is all zeros.*
- *Sorry, need to increase n_refl_max The version of RESOLVE you have is not big enough for your dataset. Try the giant or huge versions (resolve_giant or resolve_huge). If that does not work, contact the author for a bigger version.*
- *rms of this map is nan avg = nan This can happen if there are "NaN" (empty) entries in your mtz file. This version of RESOLVE does not always handle these correctly. If you remove them (mtzdump, remove lines with question marks, f2mtz) it should be ok.*
- *resolve: Open failed: File: SYMOP. RESOLVE needs the CCP4 symmetry data file "symop.lib". Usually this is located in /usr/local/lib/solve after you download resolve. If it is not there, then you'll need to find out where it is, then define the environmental variable SYMOP. Similarly, SYMINFO needs to be defined.*

  ```
  setenv SYMOP /usr/local/lib/solve/symop.lib
  setenv SYMINFO/usr/local/lib/solve/syminfo.lib
  ```

  [or wherever they are].
- *Sorry, cannot read segment library needed for automatic model-building. Or Sorry, cannot properly read fragment libraries .. RESOLVE model-building uses a library of helix/strand segments and a library of fragments. These are normally located in /usr/local/lib/solve/segments/ after standard installation of SOLVE/RESOLVE. If you define $SOLVEDIR then RESOLVE will expect to find these libraries in $SOLVEDIR/segments.*

Can RESOLVE..

- *Extend phases or fill in missing reflections? Yes, any reflections with non-zero F in your input file that are in your resolution limits are filled in by RESOLVE (version 1.04 or higher). Additionally with version 2.01 or higher, all reflections that are completely missing are filled in as well if you specify "fill"*
- *Use NC-symmetry? Yes, starting with version 2.0.*
- *Phase from a partial model? Yes, use iterative model-building for this one*
- *Use a MR starting model? Yes, use prime-and-switch phasing or use iterative model rebuilding*
- *Use a mask that I input? Yes, sort of. Make a model that covers the region you want masked. Read the model in with model mask_model.pdb. Then specify use_model_mask and that will be used. You can set the radius around each atom to be masked with rad_mask xxx*

Parameters to consider setting if SOLVE does not find a satisfactory solution

If SOLVE does not find a good solution atomatically, there are a few parameters that you might consider setting to tell it how to go about its search. Some of these are:

- THOROUGH. This keyword tells SOLVE to keep looking through all the seeds even if a good solution (figure of merit greater than 0.5, score greater than 10) has been seen. The default is QUICK, the opposite of THOROUGH.
- RATMIN. This keyword specifies the minimum F/sigma that will be used. The default is 2.0. SOLVE may be tossing too much of your data if it is weak.
- RESOLUTION. This keyword tells SOLVE the resolution limits to use. If they include high resolution but the data is very poor there, this could prevent SOLVE from finding a solution at high resolution.
- RESOLUTION_STEPS. This keyword (default = 1) tells SOLVE to cycle through shells of resolution. Sometimes SOLVE can find a solution at low resolution but not high resolution or vice-versa. Try a value of 3 if 1 does not work.
- CUTOFF_DERIV. This keyword can be used to give SOLVE more restrictive high- and low-resolution limits for a particular derivative, wavelength, or native dataset than the overall resolution limits specified with RESOLUTION.
- RES_PHASE. This keyword can be used to set a different high-resolution limit for phasing in SOLVE than the overall resolution limit.
- SN_MIN and SN_RATIO_MIN. These keywords can be used to automatically set a different high-resolution limit for phasing in SOLVE than the overall resolution limit. Default is signal-to-noise minimum of 0.1, minimum ratio to low-resolution signal-to-noise of 0.1.
- FP_OR_FM. This keyword tells SOLVE to keep reflections in MAD datasets where only F+ or only F- have been measured. If MAD data are very incomplete, SOLVE may be tossing too much data in its effort to only use matched F+/F- pairs. The default is FPFM_ONLY, only use matched pairs.
- FIXSCATTFACTORS vs REFSCATTFACTORS. These keywords tell SOLVE whether to fix or refine scattering factors f' and f". If your MAD data is pretty weak, refining scattering factors can lead to worse estimates than you started with.
- NANOMALOUS and NRES. These define the number of anomalously scattering atoms and protein residue equivalents in the asymmetric unit. NRES defines the overall scale of the data. The ratio of NANOMALOUS to NRES defines how big the expected scattering contribution from the anomalously scattering atoms should be. Additionally NANOMALOUS defines the maximum number of sites for the anomalously-scattering atoms (unless the keyword NSOLSITE is set, in which case the NSOLSITE value is used).
- NSOLSITE and NSOLSITE_DERIV. These keywords define the maximum number of heavy atom sites either for the dataset as a whole (NSOLSITE) or for a particular derivative

(NSOLSITE_DERIV). You can use them to control how much time SOLVE will spend in various derivatives.

- INANO. This keyword tells SOLVE to use anomalous differences for a derivative. If you leave it off it will not include them for MIR data. (For MAD data SOLVE will include anomalous differences unless you tell it not to with noinano).
- NOANOREFINE. This keyword tells SOLVE that even if INANO is specified for a particular derivative, the anomalous differences will not be used in heavy atom refinement. This is the default for MIR data and should be used for MIR or MAD data if the anomalous data are much weaker than isomorphous differences.
- ACCEPTANCE. This keyword tells solve how weak a site can be but still get accepted. Sites that have occupancies of about ACCEPTANCE times the average for other sites are typically accepted (but this is not a simple cutoff, it is a parameter in the overall scoring procedure). If you are not getting as many sites as you would like, try lowering the value of ACCEPTANCE (default =0.2).
- NSEEDSOLVE. This keyword tells solve how many seeds to try (default=5). If SOLVE isn't finding anything, you could try more.
- NTOPSOLVE. This keyword tells SOLVE how many solutions to keep track of at any one time. If SOLVE isn't finding anything, you could increase it (default=5). Note that this will slow things down a lot if you increase it by much.
- ICRMAX. This keyword tells solve how many peaks in the Patterson function to try as potential cross-vectors in a 2-site search. Try increasing it from 30 to a higher number.

```
#!/bin/ksh
#
#  bayesdiff -- does Bayesian differencing on two CNS output files
#             and writes the output to a third CNS input file.
#             The CNS output files should be written in the order
#               FOBS SIGMA FCALC TEST
#             where FCALC is complex (default for CNS).
#             The output of this script (and the input to the next
#             round of CNS) will be named bayesdiff.hkl unless you
#             specify otherwise.  The output defines the following
#             columns of "mutant" data that you should read in to CNS
#             and use:
#               FOBS SIGMA WEIGHT TEST
#             A log of the SOLVE run called bayesdiff.log will also
#             be created.
#
#  Usage: bayesdiff ../wt/wildtype.hkl  ../mut/mutant.hkl [output.hkl]
#
#  Author:  JB  25 Sep 99
#

#
# Check arguments
#
if [[ $# < 2 ]]; then
  echo "Usage: bayesdiff ../wt/wildtype.hkl  ../mut/mutant.hkl [output.hkl]"
  exit 0
fi
if [ -r $1 ] ; then
   wtfile=$1
else
   echo "Can't read wild-type input file $1"
   exit 0
fi
if [ -r $2 ] ; then
   intfile=$2
else
   echo "Can't read mutant input file $2"
   echo "Can't read mutant input file $2"
   exit 0
fi
outfile=${3:-bayesdiff.hkl}
```

```
#
# strip header lines at top and get rid of non-numeric parts of
# CNS file before reading it in
#
nskip=7 # number of lines to skip at the top of the file
tail +$nskip $wtfile | tr -d '[=][:alpha:]'| tr -s '[:blank:]' '[:blank:]'  > wtin.dat
tail +$nskip $intfile | tr -d '[=][:alpha:]'| tr -s '[:blank:]' '[:blank:]'  > intin.dat

#
# file up SOLVE, setting it to overwrite any output files that may exist
# from previous rounds
#
CCP4_OPEN=unknown; /usr/local/bin/solve <<EOF

logfile bayesdiff.log
SYMFILE /usr/local/lib/solve/p63.sym
CELL 61.08  61.08  110.40  90.0   90.0  120.0
RESOLUTION 20.0 2.25

IMPORT
wtin.dat
2,
wt.drg
conversion of ground-state data
5
Fobs
Sigma
Fcalc
Phi
Rtest
1.0,
y
n
y
1
4

IMPORT
intin.dat
2,
int.drg
conversion of intermediate data from X-PLOR
```

```
5
Fobs
Sigma
Fcalc
Phi
Rtest
1.0,
y
n
y
1
4

FILEMERGE
2
wt.drg
int.drg
combined.drg
ground state and intermediate data merged into same file
1
1,3
2
1,3
2
5,5
0


infile combined.drg
outfile tmpout.dat
ncolfowt 1
ncolswt 2
ncolfc 3
ncolfomut 4
ncolsmut 5
ncolfcmut 6
ncolrtest 7
fdiff
EOF

#
# fix up header of output file for CNS
#
```

```
set -A nreflection `wc -l tmpout.dat`
echo " NREFlection=    $nreflection">$outfile
cat <<EOF >>$outfile
 ANOMalous=FALSe { equiv. to HERMitian=TRUE}
 DECLare NAME=FOBS                DOMAin=RECIprocal   TYPE=REAL END
 DECLare NAME=SIGMA                DOMAin=RECIprocal   TYPE=REAL END
 DECLare NAME=WEIGHT                DOMAin=RECIprocal   TYPE=REAL END
 DECLare NAME=TEST                DOMAin=RECIprocal   TYPE=INTE END
EOF
cat tmpout.dat >>$outfile
rm -f wt.dat *.drg tmpout.dat intin.dat wtin.dat


#  END OF bayesdiff.ksh
```