Simple algorithms for macromolecular phasing

IUCr Computing School, Siena, August 2005

George M. Sheldrick

http://shelx.uni-ac.gwdg.de/SHELX/

Experimental phasing of macromolecules

Except in relatively rare cases where atomic resolution data permit the phase problem to be solved by *ab initio* direct methods, experimental phasing usually implies the presence of *heavy atoms* to provide *reference phases*. We then calculate the phases ϕ_T of the full structure by:

$$\phi_T = \phi_A + \alpha$$

Where ϕ_A is the calculated phase of the heavy atom substructure. As we will see, α can be estimated from the experimental data. The phase determination requires the following stages:

1. Location of the heavy atoms.
2. (Refinement of heavy atom parameters and) calculation of ϕ_A.
3. Calculation of starting protein phases using $\phi_T = \phi_A + \alpha$.
4. Improvement of these phases by density modification (and where appropriate NCS averaging).

SAD as a special case of MAD

$$|F_+|^2 = |F_T|^2 + a|F_A|^2 + b|F_T||F_A|\cos\alpha + c|F_T||F_A|\sin\alpha$$

$$|F_-|^2 = |F_T|^2 + a|F_A|^2 + b|F_T||F_A|\cos\alpha - c|F_T||F_A|\sin\alpha$$

where $a = (f''^2+f'^2)/f_0^2$, $b = 2f'/f_0$, $c = 2f''/f_0$ and $\alpha = \phi_T - \phi_A$

By subtracting the second equation from the first we obtain:

$$|F_+|^2 - |F_-|^2 = 2c|F_T||F_A|\sin\alpha$$

If we assume that the native structure factor $|F_T|$ is given by $|F_T| = \frac{1}{2}(|F_+| + |F_-|)$, this simplifies to:

$$|F_+| - |F_-| = c|F_A|\sin\alpha$$

where $|F_A|$ is the heavy atom structure factor) and $\phi_T = \phi_A + \alpha$. Amazingly, this is sufficient to find the heavy atoms and to use them to estimate the protein phases ϕ_T for some reflections.

SAD, SIR, SIRAS and MAD

For SAD, the reflections with the largest normalized anomalous differences $|E_A|$ will tend to have α close to 90 or 270°. These reflections are used to find the heavy atoms (only the largest $|E_A|$ are used by direct methods) and to start the phasing.

In the case of SIR, if we assume that the isomorphous difference is small compared to the native structure factor, we obtain the approximation:

$$|F_{deriv}| - |F_{nat}| = b|F_A|\cos\alpha$$

So reflections with large normalized isomorphous differences will tend to have α close to 0 or 180°. Although $||F_{deriv}|-|F_{nat}||$ will in general be larger than $||F_+|-|F_-||$, as we shall see α values of 0° or 180° are less useful than 90° or 270°, and there are problems with lack of isomorphism and scaling.

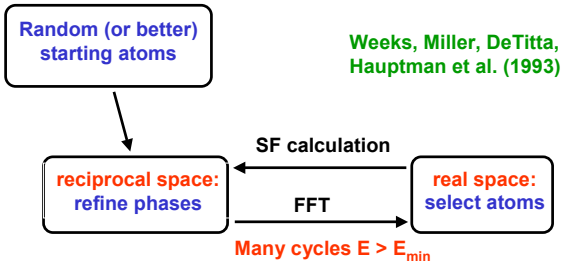
For MAD (and SIRAS) we have $F_A\sin\alpha$ and $F_A\cos\alpha$ and so we can derive both $|F_A|$ and α.

Substructure solution

The same methods used for *ab intio* all-atom structure solution from very high resolution native data turn out to be eminently suitable for the location of heavy atom sites from SIR, SAD ΔF or MAD F_A values. The resolution is then not so critical; 3.5Å is fine because it is normally still greater than the distance between the sites. In the case of sulfur-SAD, the two sulfurs in a disulfide bridge fuse into a single super-sulfur atom at this resolution.

The ΔF or F_A values are normalized to give *E*-values. The fact that direct methods use only the larger *E*-values is an advantage, especially for SIR or SAD, because the ΔF values represent lower limits on the heavy atom structure factors and so the weak ΔF are very unreliable anyway.

Dual space recycling



If the figures of merit indicate a solution, it can be expanded to the complete structure using all data

Implemented in SnB and (later) SHELXD

Random atom positions

```
JS=MOD(JS*3877+29573,139968)
JK=MOD(JK*3613+45289,214326)
JR=MOD(JR*1366+150889,714025)
X=7.1444902E-6*REAL(JS)
Y=4.6657895E-6*REAL(JK)
Z=1.4005112E-6*REAL(JR)
```

This Fortran code generates random coordinates x, y and z in the range 0…1. The use of three independent series ensures that the the repeat length is long (3.5×10^{15}). JS etc. can be given fixed starting values so that the same sequence is always generated (with luck also on different computers) or they can be made randomly random (e.g. by using the last few digits of the current time (expressed as a real number in seconds).

Probabilistic Patterson sampling (PATS in SHELXD)

Each unique general Patterson vector of suitable length is a potential HA-HA vector, and may be employed as a 2-atom search fragment in a translational search based on the *Patterson minimum function*. Alternatively a vector of known length – e.g. a S-S bond (2.03Å) – but random orientation can be used. For each position of the two atoms in the cell, the Patterson height P_j is found for all vectors between them and their equivalents, and the sum (PSUM) of the lowest (say) 35% of P_j calculated.

It would be easy to find the global maximum of PSUM using a fine 3D grid, but this often does NOT lead to the solution of the structure! A more effective approach is to generate many different starting positions by simply taking the best of a finite number of random trials each time.

The *full-symmetry Patterson superposition minimum function* is used to expand from the two atoms to a much larger number before entering the dual-space recycling.

Selecting vectors for the translational search

```
C
C Choose biased random starting vector (PATS +n)
C
        NJ=LM-JW+1
        IF(NJ.LE.0)GOTO 18
        N=JW
          DO 20 NW=1,JQ
          JR=MOD(JR*1366+150889,714025)
          N=MAX0(N,LM-MOD(JR,NJ))
  20      CONTINUE
```

This SHELXD Fortran code was cited by Ralf Grosse-Kunstleve as a typically cryptic piece of SHELX code (he diplomatically said that he found the comment useful). The general Patterson peaks are stored in XA(JW…LM) etc., JR is a random number and JQ is a 'bias factor' (third PATS parameter, usually 5) that causes (higher) peaks closer to LM to be chosen more often.

Calculating the Patterson minimum function

In its simplest form, the Patterson minimum function is simply the lowest Patterson density at a series of points in the Patterson, e.g. in the case of the symmetry minimum function these are all vectors between one site and its symmetry equivalents.

When more than one site is involved or the symmetry is high, the chance of an accidental low value is too high, so Schilling (1970) and Nordman (1980) improved the function by summing the lowest (say) 1/3 of the Patterson densities involved. Since this requires sorting the values, it can become rate determining.

Alternatively one can sum some suitable function of the Patterson densities ρ designed to put more weight on the lower values. Summing $s\sqrt{|\rho|}$ where s is the sign of ρ works quite well, even in high symmetry space groups and with noisy data, which can cause problems for the Schilling/Nordman method.

How to find the independent Patterson Vectors

Assume that we wish to find all unique non-origin vectors involving atoms x_1,y_1,z_1 and x_2,y_2,z_2 and their symmetry equivalents in order to calculate a Patterson superposition function and that there are N_S symmetry operations. Lattice operators are ignored because they will generate equivalent peaks (the Patterson has the same lattice type as the structure) but a center of symmetry should be included in the symmetry operators.

There are (N_S-1) unique Harker vectors for each of the two atoms. To find the unique cross-vectors we need to subtract x_1,y_1,z_1 from x_2,y_2,z_2 and all its symmetry equivalents. So the total number of unique vectors generated by a two-atom search fragment is $2(N_S-1) + N_S$. In the space group $P4_32_12$ this is 22. For an N_A-atom fragment the number is $N_A(N_S-1) + \frac{1}{2}(N_A-1)N_AN_S$.

Note that in high-symmetry cases some Harker vectors have multiplicities greater than one, and that in centrosymmetric space groups all non-Harker have multiplicities of at least two.

The full-symmetry Patterson superposition minimum function

SHELXD finds good (*but different*) positions for a two-atom fragment by trying (say) 10000 random translations and using the PSMF as a criterion. The full-symmetry PSMF is then calculated one pixel at a time. A dummy atom is placed on the pixel and the Patterson function values at all vectors involving it, the two atoms of the search fragment, and their symmetry equivalents found. The sum of the lowest (say 1/3) of these values (the PSMF) is stored at that pixel.

The resulting map is then peak-searched to find the starting atom sites in the dual-space recycling part of the SHELXD procedure for finding the heavy atom sites. Each overall trial generates a *different* starting set of sites that are relatively consistent with the Patterson. There is no limit to the number of starting sets that can be generated in this way.

Density modification

The heavy atoms can be used to calculate reference phases; initial estimates of the protein phases can then be obtained by adding the phase shifts α to the heavy atom phases as explained at the beginning of this talk.

These phases are then improved by density modification. Clearly, if we simply do an inverse Fourier transform of the unmodified density we get back the phases we put in. So we try to make a 'chemically sensible' modification to the density before doing the inverse FFT in the hope that this will lead to improved estimates for the phases.

Many such density modifications have been tried, some of them very sophisticated. Major contributions have been made by Kevin Cowtan and Tom Terwilliger. One of the simplest ideas, truncating negative density to zero, is actually not too bad (it is the basic idea behind the program ACORN).

The sphere of influence algorithm

The variance *V* of the density on a spherical surface of radius 2.42Å is calculated for each pixel in the map. The pixels with the highest *V* are most likely to correspond to real protein atomic positions.

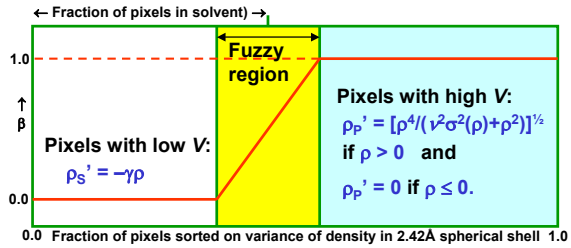
Pixels with low *V* are *flipped* ($\rho_S' = -\gamma\rho$ where γ is about one).

For pixels with high *V*, ρ is replaced by $[\rho^4/(\nu^2\sigma^2(\rho)+\rho^2)]^{½}$ (with ν usually 1.0) if positive and by zero if negative. This has a similar effect to the procedure used in the program ACORN.

A *fuzzy boundary* is used; in the *fuzzy region* ρ is set to a weighted sum of the two treatments. The *fuzzy boundary* is an attempt to avoid the *lock-in effect* of a binary mask.

The use of a spherical surface rather than a spherical volume was intended to add a little chemical information (2.42Å is a typical 1,3-distance in proteins and DNA). An empirical weighting scheme for phase recombination is used to combat model bias.

The fuzzy solvent boundary

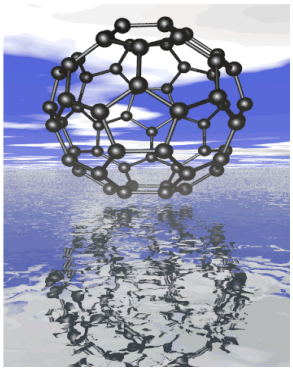


In the *fuzzy region*, the modified density is a weighted mean of the two treatments: $\rho' = \beta\rho_P' + (1-\beta)\rho_S'$

The parameters γ and ν are both usually set to 1.0

Calculating the sphere of influence

In SHELXE, the following method is used to generate the sphere. A C_{60} molecule consists of five and six-membered rings. If we make a vector from the center to each atom and also from the center to the center of each of the 32 faces, we define 92 directions that are well distributed in space. These 92 directions are stored in the form of 92 triplets of pixel offsets with vector lengths close to 2.42 Å. These are added to the coordinates of each pixel in turn to calculate the variance of the density in the *sphere of influence* of each pixel.



Graphic by Voita Jancik

The free lunch algorithm

In two recent papers (Caliandro *et al.,* Acta Cryst. D61 (2005) 556-565 and 1080-1087) the Bari group around Carmelo Giacovazzo used density modification to calculate phases for reflections that had not been measured, either completing the data to a given resolution or extending the resolution.

Their unexpected conclusion was that if these phases are now used to recalculate the density, using very rough estimates of the (unmeasured) amplitudes, the density actually improves! I have incorporated this into a test version of SHELXE and can completely confirm their observations, at least when the native data have been measured to a resolution of 2 Å or better.

Since one is apparently getting something for nothing, I propose that this algorithm be named the *free lunch algorithm*.

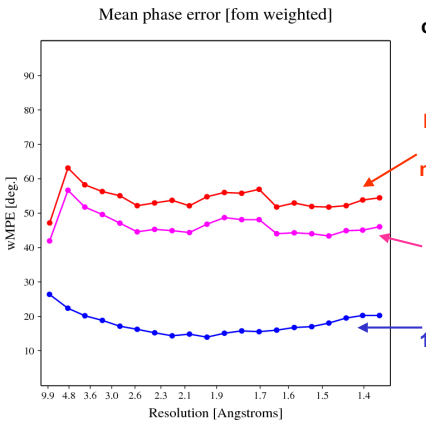
Solution of an unknown structure

The free lunch algorithm (FLA) clearly improved the density for a number of standard test structures, introducing real features that were not present in the original maps. However a particularly convincing example was the application to the solution of an unsolved structure by Isabel Usón using data collected by Clare Stevenson.

Data for this 262 amino-acid protein in space group P2 were almost complete to 1.9 Å and somewhat partial to 1.35 Å. Despite collecting six datasets the only phase information was a weak SIRAS signal to about 3.5 Å from a mercury acetate derivative. All attempts to improve these maps by interpretation or modification of the density and also molecular replacement on the native data failed.

At first we thought that the free lunch algorithm might be able to fill in the missing data, but Isabel was ambitious and expanded to 1.0 Å, much further than the crystals had ever diffracted.

Postmortem on a free lunch



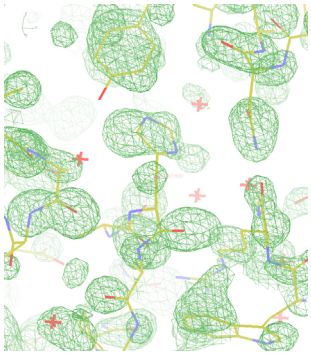
Phase errors calculated relative to the subsequently refined structure

Best experimental phases after den. mod. (MapCC 0.57)

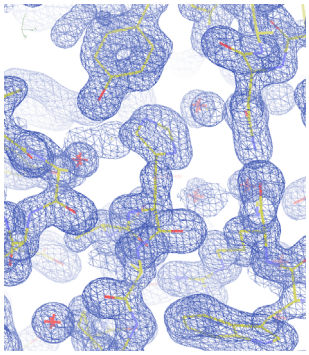
After FLA filling in to 1.34 Å (MapCC 0.66)

FLA expansion to 1.0 Å (MapCC 0.94)

Maps before and after a free lunch



Best experimental phases after den. mod. (MapCC 0.57)



After expansion to 1.0 Å with virtual data (MapCC 0.94)

Why do we get a free lunch?

It is not immediately obvious why inventing extra data improves the maps. Possible explanations are:

1. **The algorithm corrects Fourier truncation errors that may have had a more serious effect on the maps than we had realised.**
2. **Phases are more important than amplitudes (see Kevin's ducks and cats!), so as long as the extrapolated phases are OK any amplitudes will do.**
3. **Zero is a very poor estimate of the amplitude of a reflection that we did not measure.**

Acknowledgements

I am particularly grateful to Isabel Usón, Thomas R. Schneider, Stephan Rühl and Tim Grüne for many discussions.

SHELXD: Usón & Sheldrick (1999), *Curr. Opin. Struct. Biol.* 9, 643-648; Sheldrick, Hauptman, Weeks, Miller & Usón (2001), *International Tables for Crystallography Vol. F*, eds. Arnold & Rossmann, pp. 333-351; Schneider & Sheldrick (2002), *Acta Cryst.* D58, 1772-1779.

SHELXE: Sheldrick (2002), *Z. Kristallogr.* 217, 644-650; Debreczeni, Bunkóczi, Girmann & Sheldrick (2003), *Acta Cryst.* D59, 393-395; Debreczeni, Bunkóczi, Ma, Blaser & Sheldrick (2003), *Acta Cryst.* D59, 688-696; Debreczeni, Girmann, Zeeck, Krätzner & Sheldrick (2003), *Acta Cryst.* D59, 2125-2132.

http://shelx.uni-ac.gwdg.de/SHELX/