Journal of
Applied Crystallography

ISSN 0021-8898

Received 31 January 2007
Accepted 27 April 2007

Phaser crystallographic software

Airlie J. McCoy,[a]* Ralf W. Grosse-Kunstleve,[b] Paul D. Adams,[b] Martyn D. Winn,[c] Laurent C. Storoni[a]‡ and Randy J. Read[a]

[a]Department of Haematology, University of Cambridge, Cambridge Institute for Medical Research, Wellcome Trust/MRC Building, Hills Road, Cambridge CB2 0XY, UK, [b]Lawrence Berkeley National Laboratory, One Cyclotron Road, Bldg 64R0121, Berkeley, CA 94720-8118, USA, and [c]Daresbury Laboratory, Warrington WA4 4AD, UK. Correspondence e-mail: ajm201@cam.ac.uk

Phaser is a program for phasing macromolecular crystal structures by both molecular replacement and experimental phasing methods. The novel phasing algorithms implemented in *Phaser* have been developed using maximum likelihood and multivariate statistics. For molecular replacement, the new algorithms have proved to be significantly better than traditional methods in discriminating correct solutions from noise, and for single-wavelength anomalous dispersion experimental phasing, the new algorithms, which account for correlations between F^+ and F^-, give better phases (lower mean phase error with respect to the phases given by the refined structure) than those that use mean F and anomalous differences ΔF. One of the design concepts of *Phaser* was that it be capable of a high degree of automation. To this end, *Phaser* (written in C++) can be called directly from Python, although it can also be called using traditional *CCP4* keyword-style input. *Phaser* is a platform for future development of improved phasing methods and their release, including source code, to the crystallographic community.

1. Introduction

Improved crystallographic methods rely on both improved automation and improved algorithms. The software handling one part of structure solution must be automatically linked to software handling parts upstream and downstream of it in the structure solution pathway with (ideally) no user input, and the algorithms implemented in the software must be of high quality, so that the branching or termination of the structure solution pathway is minimized or eliminated. Automation allows all the choices in structure solution to be explored where the patience and job-tracking abilities of users would be exhausted, while good algorithms give solutions for poorer models, poorer data or unfavourable crystal symmetry. Both forms of improvement are essential for the success of high-throughput structural genomics (Burley *et al.*, 1999).

Macromolecular phasing by either of the two main methods, molecular replacement (MR) and experimental phasing, which includes the technique of single-wavelength anomalous dispersion (SAD), are key parts of the structure solution pathway that have potential for improvement in both automation and the underlying algorithms. MR and SAD are good phasing methods for the development of structure solution pipelines because they only involve the collection of a single data set from a single crystal and have the advantage of minimizing the effects of radiation damage. *Phaser* aims to facilitate automation of these methods through ease of scripting, and to facilitate the development of improved algorithms for these methods through the use of maximum likelihood and multivariate statistics.

Other software shares some of these features. For molecular replacement, *AMoRe* (Navaza, 1994) and *MOLREP* (Vagin & Teplyakov, 1997) both implement automation strategies, though they lack likelihood-based scoring functions. Likelihood-based experimental phasing can be carried out using *Sharp* (La Fortelle & Bricogne, 1997).

2. Algorithms

The novel algorithms in *Phaser* are based on maximum likelihood probability theory and multivariate statistics rather than the traditional least-squares and Patterson methods. *Phaser* has novel maximum likelihood phasing algorithms for the rotation functions and translation functions in MR and the SAD function in experimental phasing, but also implements other non-likelihood algorithms that are critical to success in certain cases. Summaries of the algorithms implemented in *Phaser* are given below. For completeness and for consistency of notation, some equations given elsewhere are repeated here.

2.1. Maximum likelihood

Maximum likelihood is a branch of statistical inference that asserts that the best model on the evidence of the data is the one that explains what has in fact been observed with the

‡ Present address: Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK.

highest probability (Fisher, 1922). The model is a set of parameters, including the variances describing the error estimates for the parameters. The introduction of maximum likelihood estimators into the methods of refinement, experimental phasing and, with *Phaser*, MR has substantially increased success rates for structure solution over the methods that they replaced. A set of thought experiments with dice (McCoy, 2004) demonstrates that likelihood agrees with our intuition and illustrates the key concepts required for understanding likelihood as it is applied to crystallography.

The likelihood of the model given the data is defined as the probability of the data given the model. Where the data have independent probability distributions, the joint probability of the data given the model is the product of the individual distributions. In crystallography, the data are the individual reflection intensities. These are not strictly independent, and indeed the statistical relationships resulting from positivity and atomicity underlie direct methods for small-molecule structures (reviewed by Giacovazzo, 1998). For macromolecular structures, these direct-methods relationships are weaker than effects exploited by density modification methods (reviewed by Kleywegt & Read, 1997); the presence of solvent means that the molecular transform is over-sampled, and if there is noncrystallographic symmetry then other correlations are also present. However, the assumption of independence is necessary to make the problem tractable and works well in practice.

To avoid the numerical problems of working with the product of potentially hundreds of thousands of small probabilities (one for each reflection), the log of the likelihood is used. This has a maximum at the same set of parameters as the original function.

$$\mathrm{LL}\big(\mathrm{model};\{\mathrm{data}_i\}\big) = \sum_i \ln\big[p(\mathrm{data}_i;\,\mathrm{model})\big]. \qquad (1)$$

Maximum likelihood also has the property that if the data are mathematically transformed to another function of the parameters, then the likelihood optimum will occur at the same set of parameters as the untransformed data. Hence, it is possible to work with either the structure-factor intensities or the structure-factor amplitudes. In the maximum likelihood functions in *Phaser*, the structure-factor amplitudes (*F*s), or normalized structure-factor amplitudes (*E*s, which are *F*s normalized so that the mean-square values are 1) are used.

The crystallographic phase problem means that the phase of the structure factor is not measured in the experiment. However, it is easiest to derive the probability distributions in terms of the phased structure factors and then to eliminate the unknown phase by integration, a process known as integrating out a nuisance variable (the nuisance variable being the introduced phase of the observed structure factor, or equivalently the phase difference between the observed structure factor and its expected value). The central limit theorem applies to structure factors, which are sums of many small atomic contributions, so the probability distribution for an acentric reflection, $\mathbf{F}_\mathrm{O}$, given the expected value of $\mathbf{F}_\mathrm{O}$ ($\langle\mathbf{F}_\mathrm{O}\rangle$) is a two-dimensional Gaussian with variance Σ centred on $\langle\mathbf{F}_\mathrm{O}\rangle$. (Note that here and in the following, bold font is used to represent complex or signed structure factors, and italics to represent their amplitudes.)

In applications to molecular replacement and structure refinement, $\langle\mathbf{F}_\mathrm{O}\rangle$ is the structure factor calculated from the model ($\mathbf{F}_\mathrm{C}$) multiplied by a fraction D (where $0 < D < 1$; Luzzati, 1952) that accounts for the effects of errors in the positions and scattering of the atoms that are correlated with the true structure factor. (If one works with E values, the factor D is replaced by σ_A and Σ is replaced by $1 - \sigma_\mathrm{A}^2$.) Integrating out the phase between $\mathbf{F}_\mathrm{O}$ and $\langle\mathbf{F}_\mathrm{O}\rangle$ gives

$$\begin{aligned} P\big(F_\mathrm{O};\langle F_\mathrm{O}\rangle\big) &= \frac{2F_\mathrm{O}}{\Sigma}\exp\left(-\frac{F_\mathrm{O}^2+\langle F_\mathrm{O}\rangle^2}{\Sigma}\right) I_0\left(\frac{2F_\mathrm{O}\langle F_\mathrm{O}\rangle}{\Sigma}\right) \\ &\equiv \Re\big(F_\mathrm{O},\langle F_\mathrm{O}\rangle,\Sigma\big), \end{aligned} \qquad (2)$$

where I_0 is the modified Bessel function of order 0 and $\langle F_\mathrm{O}\rangle$ represents the absolute value of $\langle\mathbf{F}_\mathrm{O}\rangle$. This is called the Rice distribution in statistical literature and is also known as the Sim (1959) distribution in crystallographic literature. The special case where $\langle F_\mathrm{O}\rangle = 0$ (*i.e.* nothing is known about the structure) is the Wilson (1949) distribution, which we denote as $\Re_0(F_\mathrm{O},\Sigma)$.

The probability distribution for a centric F_O given $\langle F_\mathrm{O}\rangle$ is the sum of two one-dimensional Gaussians:

$$\begin{aligned} P\big(F_\mathrm{O};\langle F_\mathrm{O}\rangle\big) &= \left(\frac{2}{\pi\Sigma}\right)^{1/2}\exp\left(-\frac{F_\mathrm{O}^2}{2\Sigma}\right)\cosh\left(-\frac{F_\mathrm{O}\langle F_\mathrm{O}\rangle}{\Sigma}\right) \\ &\equiv W\big(F_\mathrm{O},\langle F_\mathrm{O}\rangle,\Sigma\big). \end{aligned} \qquad (3)$$

This is called the Woolfson (1956) distribution. The special case where $\langle F_\mathrm{O}\rangle = 0$ is the centric Wilson distribution, denoted $W_0\big(F_\mathrm{O},\Sigma\big)$.

The Rice, Wilson, Woolfson and centric Wilson distributions are the basis for all the maximum likelihood functions used in *Phaser*. The analysis of each problem (*e.g.* rotation search, translation search or refinement) gives rise to different estimations of the mean of the structure-factor distribution ($\langle F_\mathrm{O}\rangle$) and different variances of the structure-factor distribution (Σ) in each case (to give *e.g.* the rotation function, translation function or refinement function, respectively).

When there is experimental error in F_O, the variance of the Gaussian is inflated by an amount ν to reflect the influence of that error. This approach to the incorporation of experimental error approximates the recorded scalar measurement error on the structure-factor intensity as a complex measurement error in the structure-factor amplitude. This approximation is a good one when the measurement error makes a much smaller contribution to the variance than other contributions (for, example the model error). The suggestion to assume that the measurement error is complex was first made by Green (1979) in the context of isomorphous replacement. It has been used subsequently by Murshudov *et al.* (1997) in *REFMAC* and by Bricogne & Irwin (1996) in Buster/TNT, and has been shown to work well in practice. The Rice probability function for acentric reflections including experimental error thus takes the form

$$P(F_O; \langle F_O \rangle) = \frac{2F_O}{\Sigma + \nu} \exp\left(-\frac{F_O^2 + \langle F_O \rangle^2}{\Sigma + \nu} \right) I_0\left(\frac{2F_O \langle F_O \rangle}{\Sigma + \nu} \right) \equiv \Re(F_O, \langle F_O \rangle, \Sigma + \nu). \quad (4)$$

The variances of the Wilson, Woolfson and centric Wilson probability distributions are similarly inflated, with Σ replaced by $\Sigma + \nu$.

2.1.1. Anisotropy correction. Maximum likelihood functions are less sensitive when there is systematic variation in intensities not expected by the likelihood functions, for example an anisotropic variation in reflection intensities with direction in reciprocal space. The sensitivity of the maximum likelihood functions can be restored in this case by effectively removing the anisotropy using the method of Popov & Bourenkov (2003), in which an anisotropic Σ_N scale factor of seven parameters is applied to both structure-factor amplitudes F and their errors (σ_F), to generate corrected E values and their errors (σ_E values). When expressed in terms of β values (Trueblood *et al.*, 1996)

$$\Sigma_N(\mathbf{h}) = K J(\mathbf{h}) \exp\left[-\left(\beta_1 h^2 + \beta_2 k^2 + \beta_3 l^2 + 2\beta_4 hk + 2\beta_5 hl + 2\beta_6 kl \right) \right], \quad \mathbf{h} = (h, k, l), \quad (5)$$

then $E_O = F_O/(\varepsilon\Sigma_N)^{1/2}$ and $\sigma_E = \sigma_F/(\varepsilon\Sigma_N)^{1/2}$, where ε is the expected intensity factor for reflection $\mathbf{h}$, which corrects for the fact that for certain reflections the contributions from symmetry-related models are identical. The function $J(\mathbf{h})$ is the intensity expected, on absolute scale, from a crystal with its atoms at rest; it depends on the content of the asymmetric unit and on the resolution of the reflection only, and it is computed using the average value of scattering determined from experimental protein crystal data (the 'BEST' curve; Popov & Bourenkov, 2003). The scale factor K and the six anisotropic parameters $(\beta_1, \ldots, \beta_6)$ are determined by refinement to maximize the Wilson log-likelihood function:

$$\text{WilsonLL} = \sum_{\mathbf{h},\text{acentric}} \ln\left[\Re_0\left(F_O, \varepsilon\Sigma_N + \sigma_F^2 \right) \right] + \sum_{\mathbf{h},\text{centric}} \ln\left[W_0\left(F_O, \varepsilon\Sigma_N + \sigma_F^2 \right) \right]. \quad (6)$$

The anisotropic β values can be interconverted to anisotropic B factors or U factors (Grosse-Kunstleve & Adams, 2002). The degree of anisotropy reported is the difference between the largest and smallest eigenvalues (B factors) of the anisotropic tensor.

2.1.2. Brute rotation function. There are two maximum likelihood rotation functions implemented in *Phaser*: the Wilson maximum likelihood rotation function (MLRF_0) and the Sim maximum likelihood rotation function (MLRF) (Read, 2001). To find the best orientation of a model, one or the other is calculated for the model on a grid of orientations covering the rotational asymmetric unit for the space group. At each search orientation the lengths of the structure factors for the model in that orientation and in its symmetry-related orientations in the unit cell are known, but the relative phases of the structure factors (which would be given by knowing the positions of the models as well as the orientations) are unknown. The probability distribution for the rotation function is thus given by a random walk of structure factors in reciprocal space; the lengths of the steps of the random walk are given by the lengths of the structure-factor contributions that make up the total structure factor for the unit cell, with an additional term being given by model incompleteness (Read, 2001; McCoy, 2004).

For the Wilson MLRF_0, the structure-factor probability for each reflection is given by a two-dimensional Gaussian centred on the origin. Integrating out the phase of F_O gives the probabilities of the structure-factor amplitudes, and the rotation function is expressed in terms of the logarithms of the probabilities:

$$\text{MLRF}_0 = \sum_{\mathbf{h},\text{acentric}} \ln\left[\Re_0\left(F_O, \varepsilon\Sigma_W + \sigma_F^2 \right) \right] + \sum_{\mathbf{h},\text{centric}} \ln\left[W_0\left(F_O, \varepsilon\Sigma_W + \sigma_F^2 \right) \right], \quad (7)$$

where $\Sigma_W = \{\sum_j D_j^2 F_j^2\} + [\Sigma_N - \sum_j D_j^2 \langle F_j^2 \rangle]$.

Each F_j represents a structure-factor contribution with unknown phase relative to the other contributions; it could be the contribution from a single symmetry copy of the rotating molecule, or the sum of symmetry-related contributions from a component with fixed orientation and position. $\Sigma_N = \langle F_O^2/\varepsilon \rangle$ is the expected value of the total structure factor. The term in curly brackets is the term given by the random walk of structure factors in the unit cell (each structure factor corrected by the correlated component of the atomic errors, D) and the term in square brackets is the additional variance due to any incompleteness of the model, *i.e.* Σ_N reduced by the expected value of the modelled contributions.

When compared with the Wilson MLRF_0, somewhat better discrimination of the best orientation is given by the Sim MLRF (Read, 2001), which is the default MLRF in *Phaser*. For the Sim MLRF, the structure-factor probability for each reflection is given by a two-dimensional Gaussian offset from the origin by the length of one of the structure-factor contributions. The probability distribution has smallest variance when the largest structure-factor contribution is chosen as the offset:

$$\text{MLRF} = \sum_{\mathbf{h},\text{acentric}} \ln\left[\Re\left(F_O, D_{\text{big}} F_{\text{big}}, \varepsilon\Sigma_S + \sigma_F^2 \right) \right] + \sum_{\mathbf{h},\text{centric}} \ln\left[W\left(F_O, D_{\text{big}} F_{\text{big}}, \varepsilon\Sigma_S + \sigma_F^2 \right) \right], \quad (8)$$

where $\Sigma_S = \{\sum_j D_j^2 F_j^2 - D_{\text{big}}^2 F_{\text{big}}^2\} + [\Sigma_N - \sum_j D_j^2 \langle F_j^2 \rangle]$ and $D_{\text{big}} F_{\text{big}} = \max[\{D_j F_j\}]$.

The maximum likelihood rotation functions are significantly different from previous Patterson-based rotation functions. The equations naturally account for knowledge of partial structure, since the structure-factor contributions F_j need not correspond only to the search model, but can correspond to any components modelled in the unit cell. The contribution from fixed and moving (*i.e.* rotating) contributions is perhaps clearer if the variances for the Sim MLRF are written in the following form

$$\Sigma_S = \Sigma_N + \Sigma_{fix} + \Sigma_{rot} - D_{big}^2 F_{big}^2,$$

$$\Sigma_{fix} = \sum_{j_{fix}} \left(D_{j_{fix}}^2 F_{j_{fix}}^2 - D_{j_{fix}}^2 \langle F_{j_{fix}}^2 \rangle \right)$$

$$\Sigma_{rot} = \sum_{j_{move}} \left(D_{j_{move}}^2 F_{j_{move}}^2 - D_{j_{move}}^2 \langle F_{j_{move}}^2 \rangle \right).$$

The subscripts j_{fix} refer to the contributions of any fixed (*i.e.* non-rotating) models that have unknown positions relative to each other (and hence structure factors with unknown relative phase); in most cases any fixed components will have known relative positions so that their contributions can be summed to a single term. The subscripts j_{move} refer to the symmetry-related contributions from the moving (*i.e.* rotating) model. Putting in the contributions of fixed components improves the sensitivity of the likelihood target in two ways. First, the perturbation term Σ_{fix} adjusts the variance according to the size of the fixed contribution, thus providing information on how much of the structure factor remains to be explained by the rotating model. Second, the fixed contribution is likely to be larger than that of any symmetry-related copy of the rotating molecule, thus reducing the overall variance through the F_{big} term. Inclusion of partial structure information in the rotation function has previously only been attempted using Patterson subtraction techniques, *i.e.* using coefficients $|F_O|^2 - |F_C|^2$ (Nordman, 1994; Zhang & Matthews, 1994) or coefficients $(|F_O| - |F_C|)^2$ (Dauter *et al.*, 1991), which suffer from the problem of achieving correct relative scaling between F_O and F_C.

Maximum likelihood rotation functions can also be used to calculate 'degenerate' translation functions, wherein the translation in two directions perpendicular to a rotation axis is determined (Read, 2001). Structure-factor contributions related by the rotation axis can be collected, whereas contributions related by other symmetry operators have unknown relative phase. Although implemented in *Phaser*, this application of MLRF has found little use in practice because current computational resources do not place limits on the calculation of a full three-dimensional fast translation function (see §§2.1.4 and 2.1.5), which has better discrimination of the correct translation. Note that the term 'degenerate' as used here does not refer to the degeneracy in the coordinates of the first MR model to be fixed in space groups with an undefined origin (*e.g.* the *y* coordinate in the standard setting of $P2_1$).

2.1.3. Fast rotation function. The Sim MLRF and Wilson $MLRF_0$ are very slow to compute. A significant speed improvement is achieved in *Phaser* by the calculation of approximations to the Wilson $MLRF_0$, the likelihood-enhanced fast rotation functions (LERFs; Storoni *et al.*, 2004). The Wilson $MLRF_0$ is used as the starting point for the approximation rather than the Sim MLRF because, although the Sim MLRF gives slightly better results than the Wilson $MLRF_0$, it requires that the biggest calculated structure factor be selected for each reflection and each orientation. The LERFs are derived from the Taylor series expansion of the Wilson $MLRF_0$ and calculated *via* fast Fourier transform. The highest peaks from the LERFs are then rescored with a maximum likelihood rotation function (Sim MLRF by default), which gives better discrimination of the correct orientation (Storoni *et al.*, 2004).

The first-order likelihood-enhanced fast rotation function (LERF1) is the first term in the Taylor series expansion of the Wilson $MLRF_0$. It can be thought of as a scaled and variance weighted version of the Patterson overlap function used in the traditional Crowther (1972) fast rotation function. The function can be expressed as

$$\mathrm{LERF1}(\mathbf{R}) = \sum_{\mathbf{h}} \sum_{\mathbf{k}} I_1^{\mathrm{t}}(\mathbf{h})\, I_1^{\mathrm{s}}(\mathbf{k})\, \chi_\Omega\left(\mathbf{h} - \mathbf{k}\mathbf{R}^{-1}\right), \qquad (9)$$

where

$$I_1^{\mathrm{t}}(\mathbf{h}) = \frac{1}{\Sigma_{N'}} \left[\frac{F_O^2(\mathbf{h})}{\varepsilon \Sigma_{N'}} - 1 \right],$$

$$I_1^{\mathrm{s}}(\mathbf{k}) = \sum_{j_{move}} D_{j_{move}}^2 F_{j_{move}}^2(\mathbf{k}) - \left\langle D_{j_{move}}^2 F_{j_{move}}^2 \right\rangle$$

and

$$\Sigma_{N'} = \Sigma_N + \Sigma_{fix}.$$

χ_Ω is the Fourier transform of the function that takes the value 1 within the spherical volume Ω and 0 outside. χ_Ω can be expressed in terms of spherical harmonics $Y_{l,m}$ and the irreducible matrices of the rotation group $D^l_{m,m'}$. When the rotation is parameterized in terms of Eulerian angles (φ, θ, ψ) the matrices take a form that enables computation of the rotation function for each θ as a two-dimensional fast Fourier transform.

The second-order likelihood-enhanced fast rotation function (LERF2) adds to LERF1 the second-order Taylor series terms only involving models related by the identity symmetry operator (*i.e.* LERF2 does not include any cross-terms between symmetry-related models with different symmetry operators). *Phaser* also has available the traditional Crowther fast rotation function (Crowther, 1972), which was implemented primarily to enable accurate comparisons with the new LERFs. Both LERF1 and LERF2 give better discrimination of the correct orientation from noise than the Crowther fast rotation function, although LERF2 does not improve the results significantly over those obtained by LERF1. Crucially, LERF2 does not significantly improve the *Z* score of a solution and therefore its presence in the peak list, and so the same orientations will be rescored with the Sim MLRF (or the Wilson $MLRF_0$) no matter which of the two functions are used. LERF1 is the fast rotation function called by default.

2.1.4. Brute translation function. At each search position in a translation function search the structure factors for the search model can be calculated. The maximum likelihood translation function (MLTF) is therefore the same function as the maximum likelihood refinement function (Read, 2001). To find the best position of a model, the MLTF is calculated for the model on a hexagonal grid of positions,

$$\mathrm{MLTF} = \sum_{\mathbf{h},\mathrm{acentric}} \ln\left[\Re\left(F_\mathrm{O}, DF_\mathrm{C}, \varepsilon\sigma_\Delta^2 + \sigma_F^2\right)\right] + \sum_{\mathbf{h},\mathrm{centric}} \ln\left[W\left(F_\mathrm{O}, DF_C, \varepsilon\sigma_\Delta^2 + \sigma_F^2\right)\right], \quad (10)$$

where $\sigma_\Delta^2 = \Sigma_\mathrm{N} - D^2\Sigma_\mathrm{P}$ and $\Sigma_\mathrm{P} = \langle F_\mathrm{C}^2/\varepsilon\rangle$.

MLTF makes good use of partial structure information to enhance the signal for the position of the model that is the subject of the search underway. The partial structure information comes from models already placed (fixed) in the asymmetric unit. This is made clearer by expressing the MLTF explicitly in terms of fixed and moving (*i.e.* translating) models:

$$\mathrm{MLTF} = \sum_{\mathbf{h},\mathrm{acentric}} \ln\left[\Re\left(F_\mathrm{O}, F_\Phi, \varepsilon\Sigma_\mathrm{T} + \sigma_F^2\right)\right] + \sum_{\mathbf{h},\mathrm{centric}} \ln\left[W\left(F_\mathrm{O}, F_\Phi, \varepsilon\Sigma_\mathrm{T} + \sigma_F^2\right)\right], \quad (11)$$

where

$$F_\Phi = \left|D_\mathrm{move}\mathbf{F}_\mathrm{move}(\mathbf{T}) + D_\mathrm{fix}\mathbf{F}_\mathrm{fix}\right|,$$

$$\Sigma_\mathrm{T} = \Sigma_\mathrm{N} - D_\mathrm{fix}^2\Sigma_\mathrm{P}^\mathrm{fix} - D_\mathrm{move}^2\Sigma_\mathrm{P}^\mathrm{move},$$

$$\Sigma_\mathrm{P}^\mathrm{move} = \left\langle F_\mathrm{move}^2/\varepsilon\right\rangle,$$

and

$$\Sigma_\mathrm{P}^\mathrm{fix} = \left\langle F_\mathrm{fix}^2/\varepsilon\right\rangle.$$

$\mathbf{F}_\mathrm{fix}$ refers to the summed contribution of fixed models with known position and phase. $\mathbf{F}_\mathrm{move}$ refers to the summed contribution of translating models with known position and phase at translation $\mathbf{T}$. Σ_T is the variance that takes into account the acquisition of extra information from the contributions of the fixed and moving models.

2.1.5. Fast translation function. As are the maximum likelihood rotation functions, the MLTF is slow to compute. A speed improvement is achieved in *Phaser* in the same way as for the Wilson MLRF_0. An approximation to MLTF, the likelihood-enhanced fast translation function (LETF), is calculated by fast Fourier transform and then the top peaks rescored with MLTF (McCoy *et al.*, 2005). The fast translation function LETF1 was derived from the first term in the Taylor series expansion of the brute translation function described above.

$$\mathrm{LETF1}(\mathbf{T}) = \sum_\mathbf{h} \frac{1}{w_\mathbf{h}\varepsilon\Sigma_\mathrm{T}}\left(\frac{\langle r\rangle_\mathbf{h} F_\mathrm{O}}{\left\langle F_\Phi^2\right\rangle^{1/2}} - 1\right)F_\Phi^2(\mathbf{T}), \quad (12)$$

where

$$\langle r\rangle_{\mathbf{h},\mathrm{acentric}} = \frac{I_1\left(2F_\mathrm{O}\left\langle F_\Phi^2\right\rangle^{1/2}/\varepsilon\Sigma_\mathrm{T}\right)}{I_0\left(2F_\mathrm{O}\left\langle F_\Phi^2\right\rangle^{1/2}/\varepsilon\Sigma_\mathrm{T}\right)}$$

and

$$\langle r\rangle_{\mathbf{h},\mathrm{centric}} = \frac{\sinh\left(F_\mathrm{O}\left\langle F_\Phi^2\right\rangle^{1/2}/\varepsilon\Sigma_\mathrm{T}\right)}{\cosh\left(F_\mathrm{O}\left\langle F_\Phi^2\right\rangle^{1/2}/\varepsilon\Sigma_\mathrm{T}\right)},$$

$$w_{\mathbf{h},\mathrm{acentric}} = 1 \quad \text{and} \quad w_{\mathbf{h},\mathrm{centric}} = 2.$$

LETF1 is calculated with a single fast Fourier transform following the method of Navaza & Vernoslova (1995). As for the brute translation function, the fast translation function is able to include known partial structure information.

Four other fast translation functions are implemented in *Phaser*. Three of these are approximations to MLTF, *i.e.* an alternative first-order approximation (LETFL) and two second-order approximations (LETF2 and LETFQ) (McCoy *et al.*, 2005). There is also a form of the correlation coefficient used by other MR translation function programs [*AMoRe* (Navaza, 1994) and *MOLREP* (Vagin & Teplyakov, 1997)]. In *Phaser*, the calculated structure factors are multiplied by the Luzatti D value that takes into account the expected coordinate error, *via* the ensembling procedure (see §2.2.2). The results are thus improved over the implementations mentioned above, which do not include this factor.

All four likelihood-enhanced (LETF) approximations to MLTF give better discrimination of the correct translation from noise than the correlation coefficient (McCoy *et al.*, 2005). The first-order approximations to MLTF also have the significant advantage that they only require one FFT sampled at $d_\mathrm{min}/4$, while the second-order approximations have the advantage of only requiring two FFTs: the correlation coefficient requires three FFTs. Although the second-order functions are better approximations than the first-order ones, the improvement in discrimination of the correct solution is minimal, and not warranted by the increase in computation time and memory required. As in the case of the rotation function, as long as the correct solution is in the list of peaks selected as a result of the LETF, the correct position will be easily identified by the superior discrimination given by MLTF after rescoring the peaks. LETF1 is chosen as the default in *Phaser*.

2.1.6. Refinement target function. Since the rotation and translation functions (both the brute and fast forms) are calculated on a grid of orientations and positions, it is unlikely that the highest scoring orientation or position in the search will correspond to the true maximum of the function. The optimal orientation and position for each component in the solution is found by refining them away from the search grid positions. In *Phaser*, appropriate choices of target function for the refinement allow it to accommodate any combination of components with defined rotation only, defined rotation and degenerate translation only, and/or defined rotation and translation. In this way, the refinement target function is different from that used in dedicated crystallographic refinement programs, which only refine structures where all components have known rotation and translation, *i.e.* all atoms have known coordinates. When there is a component of the solution that includes a rotation only or degenerate

translation component, the Sim MLRF is used; components in the solution that have known rotations and translations are incorporated as the fixed component to the Sim MLRF. When all components of the solution have rotation and translation components, the MLTF is used, as in other refinement programs. The gradients for the refinement are generated by finite difference methods (rather than analytically).

The traditional way of determining whether or not an MR solution is correct after rigid-body refinement has been to look at the *R* factor, with general opinion being that the final *R* factor should be less than 45–50% for the solution to be correct. However, the greater sensitivity of the MLRF and MLTF in discriminating the correct solution from noise with poorer models means that it is commonly the case that *Phaser* finds solutions with high signal to noise ratios, but with *R* factors considerably higher than this threshold (55% or more). The poor electron density maps for structures with *R* factors this high can make proceeding from MR to model building and restrained atomic refinement problematic, and can present a bottleneck in structure solutions by MR with *Phaser*. Model editing and electron density modification methods may nonetheless overcome this hurdle, depending on the resolution of the data, the solvent content and the presence or absence of noncrystallographic symmetry.

2.2. Multivariate statistics

The maximum likelihood functions described above are derived from univariate structure-factor distributions. Other applications, where correlations between structure factors are significant, require the joint distribution of collections of structure factors to be considered. For acentric structure factors these are defined through the multivariate complex normal distribution (Wooding, 1956),

$$P(\mathbf{F}) = |\pi\boldsymbol{\Sigma}|^{-1}\exp\left(-\mathbf{F}^{\mathrm{H}}\boldsymbol{\Sigma}^{-1}\mathbf{F}\right), \tag{13}$$

where $\mathbf{F}$ is a column vector, $\mathbf{F}^{\mathrm{H}}$ is a row vector of its complex conjugate (the Hermitian transpose) and $\boldsymbol{\Sigma}$ is the covariance matrix with elements σ_{ij} given by

$$\sigma_{ij} = \left\langle \mathbf{F}_i \mathbf{F}_j^{*} \right\rangle. \tag{14}$$

Note that the element $\sigma_{ji} = \sigma_{ij}^{*}$, *i.e.* that the matrix $\boldsymbol{\Sigma}$ is Hermitian.

If the vector $\mathbf{F}$ is partitioned into $\mathbf{G}$ and $\mathbf{H}$, multivariate statistics describes how to derive the conditional distribution of $\mathbf{G}$ given $\mathbf{H}$, $P(\mathbf{G};\mathbf{H})$, from the joint probability distribution $P(\mathbf{F})$ (Johnson & Wichern, 1998). In the applications below, $P(\mathbf{F})$ is the joint distribution of observed and calculated structure factors, and the partitioning is between the observed structure factors $\mathbf{G}$ and the calculated structure factors $\mathbf{H}$. Assuming that the expected values of $\mathbf{F}$ are all zero before introducing information from $\mathbf{H}$,

$$P(\mathbf{G};\mathbf{H}) = \left|\pi\boldsymbol{\Sigma}_{GG;HH}\right|^{-1} \times \exp\left[-\left(\mathbf{G}-\boldsymbol{\mu}_{G;H}\right)^{\mathrm{H}}\boldsymbol{\Sigma}_{GG;HH}^{-1}\left(\mathbf{G}-\boldsymbol{\mu}_{G;H}\right)\right], \tag{15}$$

where the mean $\boldsymbol{\mu}_{G;H} = \boldsymbol{\Sigma}_{GH}\boldsymbol{\Sigma}_{HH}^{-1}\mathbf{H}$ and the covariance matrix $\boldsymbol{\Sigma}_{GG;HH} = \boldsymbol{\Sigma}_{GG} - \boldsymbol{\Sigma}_{GH}\boldsymbol{\Sigma}_{HH}^{-1}\boldsymbol{\Sigma}_{GH}^{\mathrm{H}}$, and the initial covariance matrix is partitioned as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{GG} & \boldsymbol{\Sigma}_{GH} \\ \boldsymbol{\Sigma}_{GH}^{\mathrm{H}} & \boldsymbol{\Sigma}_{HH} \end{bmatrix}.$$

The standard manipulations give the form of the conditional probability of observed structure factors given the calculated structure factors, with the mean of the distribution and the terms in the covariance matrices calculated from first principles.

For centric reflections, the multivariate normal distribution is applied to real numbers, and the covariance matrix is symmetric.

2.2.1. SAD Function. The SAD likelihood function for an acentric reflection for which F_{O}^{+} and F_{O}^{-} are both measured is derived by introducing the phase of the observed structure factors and then integrating out these phases at the end of the analysis:

$$\begin{aligned} P\left(F_{\mathrm{O}}^{+}, F_{\mathrm{O}}^{-}; \mathbf{F}_H^{+}, \mathbf{F}_H^{-}\right) &= \int_0^{2\pi}\int_0^{2\pi} P\left(F_{\mathrm{O}}^{+}, \alpha^{+}, F_{\mathrm{O}}^{-}, \alpha^{-}; \mathbf{F}_H^{+}, \mathbf{F}_H^{-}\right) \mathrm{d}\alpha^{+}\mathrm{d}\alpha^{-} \\ &= \int_0^{2\pi} P\left(F_{\mathrm{O}}^{-}, \alpha^{-}; \mathbf{F}_H^{+}, \mathbf{F}_H^{-}\right)\left[\int_0^{2\pi} P\left(F_{\mathrm{O}}^{+}, \alpha^{+}; F_{\mathrm{O}}^{-}, \alpha^{-}, \mathbf{F}_H^{+}, \mathbf{F}_H^{-}\right) \mathrm{d}\alpha^{+}\right]\mathrm{d}\alpha^{-}. \end{aligned} \tag{16}$$

$\mathbf{F}_H^{+}$ and $\mathbf{F}_H^{-*}$ are the structure factors calculated from the anomalous substructure. The probabilities $P(F_{\mathrm{O}}^{-}, \alpha^{-}; \mathbf{F}_H^{+}, \mathbf{F}_H^{-})$ and $P(F_{\mathrm{O}}^{+}, \alpha^{+}; F_{\mathrm{O}}^{-}, \alpha^{-}, \mathbf{F}_H^{+}, \mathbf{F}_H^{-})$ are derived using standard manipulations from the joint probability distribution $P(\mathbf{F}_{\mathrm{O}}^{+}, \mathbf{F}_{\mathrm{O}}^{-*}, \mathbf{F}_H^{+}, \mathbf{F}_H^{-*})$ where $\mathbf{F}_{\mathrm{O}}^{+}$ and $\mathbf{F}_{\mathrm{O}}^{-*}$ are the phased observed structure-factor amplitudes. The term in square brackets can be integrated analytically to give a Rice distribution, which primarily accounts for the anomalous difference. The other term accounts for the anomalous scatterers being part of the model of the total scattering. In addition to this term for acentric reflections for which F_{O}^{+} and F_{O}^{-} are both measured, the SAD likelihood function includes a term for acentric reflections for which only F_{O}^{+} or F_{O}^{-} is recorded ('singleton' reflections) and a term for centric reflections. These terms describe the phase information obtained from the partial structure contributed by the anomalous scatterers. The information from the normal scattering components is useful even if the anomalous scatterer is relatively light and can be very significant if the anomalous scatterer is also a heavy atom.

$$\begin{aligned} \mathrm{SAD} = &\sum_{\mathbf{h},\mathrm{acentric}} \ln\left\{\frac{F_{\mathrm{O}}^{-}}{\pi\left(\varepsilon\sigma_{\Delta}^{2}+\sigma_{F-}^{2}\right)}\int_0^{2\pi}\left[\exp\left(-\frac{\left|\mathbf{F}_{\mathrm{O}}^{-}-\mathbf{F}_H^{-}\right|^{2}}{\varepsilon\sigma_{\Delta}^{2}+\sigma_{F-}^{2}}\right)\right.\right. \\ &\left.\left.\times\,\Re\left(F_{\mathrm{O}}^{+}, F_{\mathrm{C}}^{+}, \varepsilon\sigma_{+}+\sigma_{F+}^{2}+\sigma_{F-}^{2}\right)\mathrm{d}\alpha^{-}\right]\right\} \\ &+\sum_{\mathbf{h},\mathrm{centric}} \ln\left[W\left(F_{\mathrm{O}}, F_H, \varepsilon\sigma_{\Delta}^{2}+\sigma_{F}^{2}\right)\right] \\ &+\sum_{\mathbf{h},\mathrm{singleton}} \ln\left[\Re\left(F_{\mathrm{O}}^{+/-}, F_H^{+/-}, \varepsilon\sigma_{\Delta}^{2}+\sigma_{F+/-}^{2}\right)\right], \end{aligned} \tag{17}$$

where $F_{\mathrm{C}}^{+} = \left|\mathbf{F}_H^{+} + \mathbf{D}_{\Phi}\left(\mathbf{F}_{\mathrm{O}}^{-} - \mathbf{F}_H^{-}\right)\right|$.

The variance terms σ_Δ^2 and σ_+, and the real and imaginary components of $\mathbf{D}_\Phi$ are refined along with the atomic parameters to optimize the log-likelihood. The term σ_Δ^2 measures the error in predicting a single structure factor using only the information from the corresponding single calculated structure factor, and roughly corresponds to a measure of missing real scattering power. The term σ_+ measures the error in predicting F_O^+ using the information from F_O^- and the calculated structure factors for both hands, and roughly corresponds to a measure of the error in the calculated anomalous differences. Finally, the term $\mathbf{D}_\Phi$ accounts for the effect of correlated errors in $\mathbf{F}_H^+$ and $\mathbf{F}_H^-$.

The SAD likelihood function explicitly accounts for the correlations between F_O^+ and F_O^- (McCoy *et al.*, 2004; Pannu & Read, 2004). Only one numerical (phase) integration is required. The number of phase points used for the integration is dynamically allocated to each reflection based on the variances for that reflection. Large variances mean that the probability distribution is diffuse, and few points are needed to calculate the integral. Small variances mean that the probability distribution is sharp, and many points are needed in order to sample the peaks of the distribution.

Log-likelihood gradient maps, analogous to those used for other likelihood targets in *Sharp* (Vonrhein *et al.*, 2006), are calculated to determine the possible positions of new atomic sites. Log-likelihood gradient maps are specific to the values of $f + f'$ and used for the calculation of the map coefficients, corresponding to the anomalous scatterer whose position is sought. Log-likelihood gradient maps can also be calculated for purely real (by setting $f + f' = 1$, $f'' = 0$) or purely anomalous (by setting $f + f' = 0$, $f'' = 1$) scatterers.

2.2.2. Ensembling. A set of structurally aligned models from the PDB can be used to generate a single calculated structure factor set using an 'ensembling' procedure. The method uses the estimated r.m.s. deviation between the model and the target to weight the structure factors contributing to the set and to determine the fall-off in structure factors with resolution.

The joint probability distribution of the target and model structure factors has a covariance matrix that can be partitioned as

$$\mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_{tt} & \mathbf{\Sigma}_{tm} \\ \mathbf{\Sigma}_{tm}^T & \mathbf{\Sigma}_{mm} \end{pmatrix}, \tag{18}$$

where the subscripts t and m refer to the target and model structure factors, respectively.

$\mathbf{\Sigma}_{tt}$ is a 1×1 matrix (*i.e.* a scalar), and when the analysis is performed in terms of the normalized structure factors (*i.e.* structure factors normalized so that their mean-square values are one), then $\mathbf{\Sigma}_{tt} = 1$.

$\mathbf{\Sigma}_{tm}$ is a $1 \times n$ row vector of σ_A values between the target and n models, which are approximated for each model using a four-parameter curve (Murshudov *et al.*, 1997)

$$\sigma_{tm} = \left\{ f_P \left[1 - f_{sol} \exp\left(\frac{-B_{sol}}{4d^2} \right) \right] \right\}^{1/2} \exp\left(\frac{-2\pi^2 \mathrm{RMS}^2}{3d^2} \right), \tag{19}$$

where f_P (= 1 by default) is the fraction of ordered structure modelled, f_{sol} (= 0.95 by default) and B_{sol} (= 300 Å^2 by default) describe the low-resolution fall-off from not modelling the bulk solvent, RMS is the estimated r.m.s. deviation of the atoms in the model to the atoms in the target structure, and d is the resolution. The default values of f_{sol} and B_{sol} were chosen by examining σ_A curves for a variety of data sets. The r.m.s. deviation must be given as input, but can be entered indirectly *via* sequence identity using the formula of Chothia & Lesk (1986), which relates the r.m.s. deviation of main-chain atoms to the sequence identity ($f_{identity}$), but with the minimum increased from 0.4 to 0.8 Å.

$$\mathrm{RMS} = \max\left\{ 0.8\,\text{Å}, 0.4\,\text{Å} \times \exp\left[1.87 \times \left(1.0 - f_{identity} \right) \right] \right\}. \tag{20}$$

The r.m.s. deviation given by this formula can be a severe underestimate if there is conformational difference between the model(s) and the target structure. If such a conformational difference is expected or suspected, then the r.m.s. deviation should be inflated from the value determined from the formula and entered directly (for example, see McCoy, 2007). As there is no equivalent formula for RNA or DNA, the r.m.s. deviation of nucleic acid must be entered directly.

$\mathbf{\Sigma}_{mm}$ is the $n \times n$ covariance matrix involving only the models. When normalized structure factors are used, it becomes a correlation matrix with diagonal elements equal to 1 and the off-diagonal elements given by

$$\rho_{ij} = \left\langle \mathbf{E}_i \mathbf{E}_j^* \right\rangle. \tag{21}$$

The off-diagonal terms will not have a significant imaginary term unless the models are translationally misaligned, leading to a systematic phase shift. This will never be the case for correctly aligned structures, and so the off-diagonal terms are therefore assumed to be real,

$$\rho_{ij} = \left\langle \Re\left(\mathbf{E}_i \mathbf{E}_j^* \right) \right\rangle = \left\langle E_i E_j \cos\left(\alpha_i - \alpha_j \right) \right\rangle. \tag{22}$$

The ensemble structure factor is then taken as the mean of the distribution and is given by

$$E_{ens} = \mathbf{\Sigma}_{tm} \mathbf{\Sigma}_{mm}^{-1} \mathbf{E} = \sum_j w_j E_j, \tag{23}$$

where w_j are the weights applied to the model normalized structure factors.

$$\sigma E_{ens} = 1 - \mathbf{\Sigma}_{tm} \mathbf{\Sigma}_{mm}^{-1} \mathbf{\Sigma}_{tm}^T. \tag{24}$$

The ensemble structure factors could be calculated for the models in each orientation and position in the rotation and translation searches, but this would be prohibitively time consuming. Instead, structure factors are calculated for a model in a large $P1$ unit cell and structure factors for the orientation and position in the correct unit cell generated by structure-factor interpolation (Lattman & Love, 1970).

2.3. Normal-mode analysis

Suhre & Sanejouand (2004) have shown that perturbation of a model along the lowest frequency normal modes can

generate a model that is closer to the target structure when there has been a conformational change between the model and the target structure. This method has now been implemented in *Phaser*. The normal modes of the elastic network model are obtained by eigenvalue decomposition of the Hessian matrix H:

$$H = \begin{bmatrix} H_{a=1,b=1} & \cdots & H_{a=1,b=N} \\ \vdots & \ddots & \vdots \\ H_{a=N,b=1} & \cdots & H_{a=N,b=N} \end{bmatrix}, \quad (25)$$

where a and b refer to the atom numbers and N is the number of atoms. $H_{a,b}$ are the 3×3 matrices containing the second derivatives of the energy with respect to the three spatial coordinates:

$$H_{a,b} = \frac{1}{2|\mathbf{r}_{ab}|^2} \times \begin{bmatrix} (\mathbf{r}_{a,b}\cdot\mathbf{x})(\mathbf{r}_{a,b}\cdot\mathbf{x}) & (\mathbf{r}_{a,b}\cdot\mathbf{x})(\mathbf{r}_{a,b}\cdot\mathbf{y}) & (\mathbf{r}_{a,b}\cdot\mathbf{x})(\mathbf{r}_{a,b}\cdot\mathbf{z}) \\ -(\mathbf{r}_{a,b}\cdot\mathbf{y})(\mathbf{r}_{a,b}\cdot\mathbf{x}) & (\mathbf{r}_{a,b}\cdot\mathbf{y})(\mathbf{r}_{a,b}\cdot\mathbf{y}) & (\mathbf{r}_{a,b}\cdot\mathbf{y})(\mathbf{r}_{a,b}\cdot\mathbf{z}) \\ -(\mathbf{r}_{a,b}\cdot\mathbf{z})(\mathbf{r}_{a,b}\cdot\mathbf{x}) & -(\mathbf{r}_{a,b}\cdot\mathbf{z})(\mathbf{r}_{a,b}\cdot\mathbf{y}) & (\mathbf{r}_{a,b}\cdot\mathbf{z})(\mathbf{r}_{a,b}\cdot\mathbf{z}) \end{bmatrix}, \quad (26)$$

when $|\mathbf{r}_{a,b}| \leq R$ and where $\mathbf{r}_{a,b} = \mathbf{r}_a - \mathbf{r}_b$, $\mathbf{r}_a$ and $\mathbf{r}_b$ are the coordinates of the atoms a and b, R is the cut-off radius for considering the interaction (= 5 Å by default), and C is the force constant (= 1 by default). When $|\mathbf{r}_{a,b}| > R$, $H = 0$. The atoms are taken to be of equal mass. The eigenvalues λ and eigenvectors U of H can then be calculated.

$$\lambda U = HU. \quad (27)$$

The eigenvalues are directly proportional to the squares of the vibrational frequencies of the normal modes, the lowest eigenvalues thus giving the lowest normal modes. Six of the eigenvalues will be zero, corresponding to the six degrees of freedom for a rotation and translation of the entire structure.

For all but the smallest proteins, eigenvalue decomposition of the all-atom Hessian is not computationally feasible with current computer technology. Various methods have been developed to reduce the size of the eigenvalue problem. Bahar *et al.* (1997) and Hinsen (1998) have shown that it is possible to find the lowest frequency normal modes of proteins in the elastic network model by considering amino acid Cα atoms only. However, this merely postpones the computational problem until the proteins are an order of magnitude larger. The problem is solved for any size protein with the rotation–translation block (RTB) approach (Durand *et al.*, 1994; Tama *et al.*, 2000), where the protein is divided into blocks of atoms and the rotation and translation modes for each block used project the full Hessian into a lower dimension. The projection matrix is a block-diagonal matrix of dimensions $3N \times 3N$.

$$P = \begin{bmatrix} P_{nb=1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & P_{nb=NB} \end{bmatrix}. \quad (28)$$

Each of the NB block matrices P_{nb} has dimensions $3N_{nb} \times 6$, where N_{nb} is the number of atoms in the block nb,

$$P_{nb} = \begin{bmatrix} P_{nb,j=1} \\ \vdots \\ P_{nb,j=N_{nb}} \end{bmatrix}. \quad (29)$$

For atom j in block nb displaced $\mathbf{r} = \mathbf{r}_j - \bar{\mathbf{r}}_{nb}$ from the centre of mass, $\bar{\mathbf{r}}_{nb}$ of the block, the 3×6 matrix $P_{nb,j}$ is

$$P_{nb,j} = \begin{bmatrix} 1 & 0 & 0 & 0 & \mathbf{r}\cdot\mathbf{z} & -\mathbf{r}\cdot\mathbf{y} \\ 0 & 1 & 0 & -\mathbf{r}\cdot\mathbf{z} & 0 & \mathbf{r}\cdot\mathbf{x} \\ 0 & 0 & 1 & \mathbf{r}\cdot\mathbf{y} & -\mathbf{r}\cdot\mathbf{x} & 0 \end{bmatrix}. \quad (30)$$

The first three columns of the matrix contain the infinitesimal translation eigenvectors of the block and last three columns contain the infinitesimal rotation eigenvectors of the block. The orthogonal basis Q of P_{nb} is then found by QR decomposition:

$$P_{nb} = Q_{nb}R_{nb}, \quad (31)$$

where Q_{nb} is a $3N_{nb} \times 6$ orthogonal matrix and R_{nb} is a 6×6 upper triangle matrix. H can then be projected into the subspace spanned by the translation/rotation basis vectors of the blocks:

$$H_{\rm P} = Q^{-1}HQ, \quad (32)$$

where

$$Q = \begin{bmatrix} Q_{nb=1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & Q_{nb=NB} \end{bmatrix}.$$

The eigenvalues $\lambda_{\rm P}$ and eigenvectors $U_{\rm P}$ of the projected Hessian are then found.

$$\lambda_{\rm P}U_{\rm P} = H_{\rm P}U_{\rm P}. \quad (33)$$

The RTB method is able to restrict the size of the eigenvalue problem for any size of protein with the inclusion of an appropriately large N_{nb} for each block. In the implementation of the RTB method in *Phaser*, N_{nb} for each block is set for each protein such that the total size of the eigenvalue problem is restricted to a matrix $H_{\rm P}$ of maximum dimensions 750×750. This enables the eigenvalue problem to be solved in a matter of minutes with current computing technology. The eigenvectors of the translation/rotation subspace can then be expanded back to the atomic space (dimensions of U are $N \times N$):

$$U = Q^{-1}U_{\rm P}. \quad (34)$$

As for the decomposition of the full Hessian H, the eigenvalues are directly proportional to the squares of the vibrational frequencies of the normal modes, the lowest eigenvalues thus giving the lowest normal modes. Although the eigenvalues and eigenvectors generated from decomposition of the full Hessian and using the RTB approach will diverge with increasing frequency, the RTB approach is able to model with good accuracy the lowest frequency normal modes, which are the modes of interest for looking at conformational difference in proteins.

The all-atom, Cα only and RTB normal-mode analysis methods are implemented in *Phaser*. After normal-mode analysis, n normal modes can be used to generate $2^n - 1$ (nonzero) combinations of normal modes. *Phaser* allows the user to specify the r.m.s. deviation between model and target desired by the perturbation, and the fraction dq of the displacement vector for each mode combination corresponding to each model combination is then used to generate the models. Large r.m.s. deviations will cause the geometry of the model to become distorted. *Phaser* reports when the model becomes so distorted that there are Cα clashes in the structure.

2.4. Packing function

The packing of potential solutions in the asymmetric unit is not inherently part of the translation function. It is therefore possible that an arrangement of models has a high log-likelihood gain, although the models may overlap and therefore be physically unreasonable. The packing of the solutions is checked using a clash test using a subset of the atoms in the structure: the 'trace' atoms. For proteins, the trace atoms are the Cα positions, spaced at 3.8 Å. For nucleic acid, the phosphate and C atoms in the ribose-phosphate backbone and the N atoms of the bases are selected as trace atoms. These atoms are also spaced at about 3.8 Å, so that the density of trace atoms in nucleic acid is similar to that of proteins, which makes the number of protein–protein, protein–nucleic acid and nucleic acid–nucleic acid clashes comparable where there is a mixed protein–nucleic acid structure.

For the clash test, the number of trace atoms from another model within a given distance (default 3 Å) is counted. The clash test includes symmetry-related copies of the model under consideration, other components in the asymmetric unit and their symmetry-related copies. If the search model has a low sequence identity with the target, or has large flexible loops that could adopt an alternative conformation, the number of clashes may be expected to be nonzero. By default the best packing solutions are carried forward, although a specific number of allowed clashes may also be given as the cut-off for acceptance. However, it is better to edit models before use so that structurally nonconserved surface loops are excluded, as they will only contribute noise to the rotation and translation functions.

Where an ensemble of structures is used as the model, the highest homology model is taken as the template for the packing search. Before this model is used, the trace atom positions are edited to take account of large conformational differences between the models in the ensemble. Equivalent trace atom positions are compared and if the coordinates deviate by more than 3 Å then the template trace atom is deleted. Thus, use of an ensemble not only improves signal to noise in the maximum likelihood search functions, it also improves the discrimination of possible solutions by the packing function.

2.5. Minimizer

Minimization is used in *Phaser* to optimize the parameters against the appropriate log-likelihood function in the anisotropy correction, in MR (refines the position and orientation of a rigid-body model) and in SAD phasing. The same minimizer code is used for all three applications and has been designed to be easily extensible to other applications. The minimizer for the anisotropy correction uses Newton's method, while MR and SAD use the standard Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. Both minimization methods in *Phaser* include a line search. The line search algorithm is a basic iterative method for finding the local minimum of a target function f. Starting at parameters $\boldsymbol{x}$, the algorithm finds the minimum (within a convergence tolerance) of

$$\varphi(\gamma) = f(\boldsymbol{x} + \gamma \boldsymbol{d}) \tag{35}$$

by varying γ, where γ is the step distance along a descent direction $\boldsymbol{d}$. Newton's method and the BFGS algorithm differ in the determination of the descent direction $\boldsymbol{d}$ that is passed to the line search, and thus the speed of convergence. Within one cycle of the line search (where there is no change in $\boldsymbol{d}$) the trial step distances γ are chosen using the golden section method. The golden ratio ($5^{1/2}/2 + 1/2$) divides a line so that the ratio of the larger part to the total is the same as the ratio of the smaller to larger. The method makes no assumptions about the function's behaviour; in particular, it does not assume that the function is quadratic within the bracketed section. If this assumption were made, the line search could proceed *via* parabolic interpolation.

Newton's method uses the Hessian matrix H of second derivatives and the gradient g at the initial set of parameters $\boldsymbol{x}_0$ to find the values of the parameters at the minimum $\boldsymbol{x}_{\rm min}$.

$$\boldsymbol{x}_{\rm min} = \boldsymbol{x}_0 - H(\boldsymbol{x}_0)^{-1} g(\boldsymbol{x}_0). \tag{36}$$

If the function is quadratic in $\boldsymbol{x}$ then Newton's method will find the minimum in one step, but if not, iteration is required. The method requires the inversion of the Hessian matrix, which, for large matrices, consumes a large amount of computational time and memory resources. The eigenvalues of the Hessian need to be positive for the function to be at a minimum, rather than a maximum or saddle point, since the method converges to any point where the gradient vector is zero. When used with the anisotropy correction, the full Hessian matrix is calculated analytically.

The BFGS algorithm is one of the most powerful minimization methods when calculation of the full Hessian using analytic or finite difference methods is very computationally intensive. At every step, the gradient search vector is analysed to build up an approximate Hessian matrix H, in order to make the resulting search vector direction $\boldsymbol{d}$ better than the original gradient vector direction. In the 'pure' form of the BFGS algorithm, the method is started with matrix H equal to the identity matrix. The off-diagonal elements of the Hessian, the mixed second derivatives (*i.e.* $\partial^2 {\rm LL}/\partial p_i \partial p_j$) are thus initially zero. As the BFGS cycle proceeds, the off-diagonal

elements become nonzero using information derived from the gradient. However, in *Phaser*, the matrix H is not the identity but rather is seeded with diagonal elements equal to the second derivatives of the parameters (p_i) with respect to the log-likelihood target function (LL) (*i.e.* $\partial^2 \mathrm{LL}/\partial p_i^2$, or curvatures), the values found in the 'true' Hessian. For the SAD refinement the diagonal elements are calculated analytically, but for the MR refinement the diagonal elements are calculated by finite difference methods. Seeding the Hessian with the diagonal elements dramatically accelerates convergence when the parameters are on different scales; when an identity matrix is used, the parameters on a larger scale can fail to shift significantly because their gradients tend to be smaller, even though the necessary shifts tend to be larger. In the inverse Hessian, small curvatures for parameters on a large scale translate into large scale factors applied to the corresponding gradient terms. If any of these curvature terms are negative (as may happen when the parameters are far from their optimal values), the matrix is not positive definite. Such a situation is corrected by using problem-specific information on the expected relative scale of the parameters from the 'large-shift' variable, as discussed below in §2.5.1.

In addition to the basic minimization algorithms, the minimizer incorporates the ability to bound, constrain, restrain and reparameterize variables, as discussed in detail below. Bounds must be applied to prevent parameters becoming nonphysical, constraints effectively reduce the number of parameters, restraints are applied to include prior probability information, and reparameterization of variables makes the parameter space more quadratic and improves the performance of the minimizer.

2.5.1. Problem-specific parameter scaling information. When a function is defined for minimization in *Phaser*, information must be provided on the relative scales of the parameters of that function, through a 'large-shifts' variable. As its name implies, the variable defines the size of a parameter shift that would be considered 'large' for each parameter. The ratios of these large-shift values thus specify prior knowledge about the relative scales of the different parameters for each problem. Suitable large-shift values are found by a combination of physical insight (*e.g.* the size of a coordinate shift considered to be large will be proportional to $d_{\min}$ for the data set) and numerical simulations, studying the behaviour of the likelihood function as parameters are varied systematically in a variety of test cases.

The large-shifts information is used in two ways. Firstly, it is used to prevent the line search from taking an excessively large step, which can happen if the estimated curvature for a parameter happens to be too small and can lead to the refinement becoming numerically unstable. If the initial step for a line search would change any parameter by more than its large-shift value, the initial step is scaled down. Secondly, it is used to provide relative scale information to correct negative curvature values. Parameters with positive curvatures are used to define the average relationship between the large-shift values and the curvatures, which can then be used to compute appropriate curvature values for the parameters with negative curvatures. This stabilizes the refinement until it is sufficiently close to the minimum that all curvatures become positive.

2.5.2. Reparameterization. Second-order minimization algorithms in effect assume that, at least in the region around the minimum, the function can be approximated as a quadratic. Where this assumption holds, the minimizer will converge faster. It is therefore advantageous to use functions of the parameters being minimized so that the target function is more quadratic in the new parameter space than in the original parameter space (Edwards, 1992). For example, atomic B factors tend to converge slowly to their refined values because the B factor appears in the exponential term in the structure-factor equation. Although any function of the parameters can be used for this purpose, we have found that taking the logarithm of a parameter is often the most effective reparameterization operation (not only for the B factors).

$$x' = \ln(x + x_{\mathrm{offset}}). \tag{37}$$

The offset x_{offset} is chosen so that the value of x' does not become undefined for allowed values of x, and to optimize the quadratic nature of the function in x'. For instance, atomic B factors are reparameterized using an offset of 5 Å^2, which allows the B factors to approach zero and also has the physical interpretation of accounting roughly for the width of the distribution of electrons for a stationary atom.

2.5.3. Bounds. Bounds on the minimization are applied by setting upper and/or lower limits for each variable where required (*e.g.* occupancy minimum set to zero). If a parameter reaches a limit during a line search, that line search is terminated. In subsequent line searches, the gradient of that parameter is set to zero whenever the search direction would otherwise move the parameter outside of its bounds. Multiplying the gradient by the step size thus does not alter the value of the parameter at its limit. The parameter will remain at its limit unless calculation of the gradient in subsequent cycles of minimization indicates that the parameter should move away from the boundary and into the allowed range of values.

2.5.4. Constraints. Space-group-dependent constraints apply to the anisotropic tensor applied to Σ_{N} in the anisotropic diffraction correction. Atoms on special positions also have constraints on the values of their anisotropic tensor. The anisotropic displacement ellipsoid must remain invariant under the application of each symmetry operator of the space group or site-symmetry group, respectively (Giacovazzo, 1992; Grosse-Kunstleve & Adams, 2002). These constraints reduce the number of parameters by either fixing some values of the anisotropic B factors to zero or setting some sets of B factors to be equal. The derivatives in the gradient and Hessian must also be constrained to reflect the constraints in the parameters.

2.5.5. Restraints. Bayes' theorem describes how the probability of the model given the data is related to the likelihood and gives a justification for the use of restraints on the parameters of the model.

$$P(\mathrm{model};\mathrm{data}) = \frac{P(\mathrm{data};\mathrm{model})P(\mathrm{model})}{P(\mathrm{data})}. \tag{38}$$

If the probability of the data is taken as a constant, then

$$P(\text{model};\text{data}) \propto L(\text{model};\text{data})P(\text{model}). \quad (39)$$

$P(\text{model})$ is called the prior probability. When the logarithm of the above equation is taken,

$$\ln[P(\text{model};\text{data})] = k + \text{LL}(\text{model};\text{data}) + \ln[P(\text{model})]. \quad (40)$$

Prior probability is therefore introduced into the log-likelihood target function by the addition of terms. If parameters of the model are assumed to have independent Gaussian probability distributions, then the Bayesian view of likelihood will lead to the addition of least-squares terms and hence least-squares restraints on those parameters, such as the least-squares restraints applied to bond lengths and bond angles in typical macromolecular structure refinement programs. In *Phaser*, least-squares terms are added to restrain the *B* factors of atoms to the Wilson *B* factor in SAD refinement, and to restrain the anisotropic *B* factors to being more isotropic (the 'sphericity' restraint). A similar sphericity restraint is used in *SHELXL* (Sheldrick, 1995) and in *REFMAC5* (Murshudov *et al.*, 1999).

3. Automation

Phaser is designed as a large set of library routines grouped together and made available to users as a series of applications, called modes. The routine-groupings in the modes have been selected mainly on historical grounds; they represent traditional steps in the structure solution pipeline. There are 13 such modes in total: 'anisotropy correction', 'cell content analysis', 'normal-mode analysis', 'ensembling', 'fast rotation function', 'brute rotation function', 'fast translation function', 'brute translation function', 'log-likelihood gain', 'rigid-body refinement', 'single-wavelength anomalous dispersion', 'automated molecular replacement' and 'automated experimental phasing'. The 'automated molecular replacement' and 'automated experimental phasing' modes are particularly powerful and aim to automate fully structure solution by MR and SAD, respectively.

Aspects of the decision making within the modes are under user input control. For example, the 'fast rotation function' mode performs the ensembling calculation, then a fast rotation function calculation and then rescores the top solutions from the fast search with a brute rotation function. There are three possible fast rotation function algorithms and two possible brute rotation functions to choose from. There are four possible criteria for selecting the peaks in the fast rotation function for rescoring with the brute rotation function, and for selecting the results from the rescoring for output. Alternatively, the rescoring of the fast rotation function with the brute rotation function can be turned off to produce results from the fast rotation function only. Other modes generally have fewer routines but are designed along the same principles (details are given in the documentation).

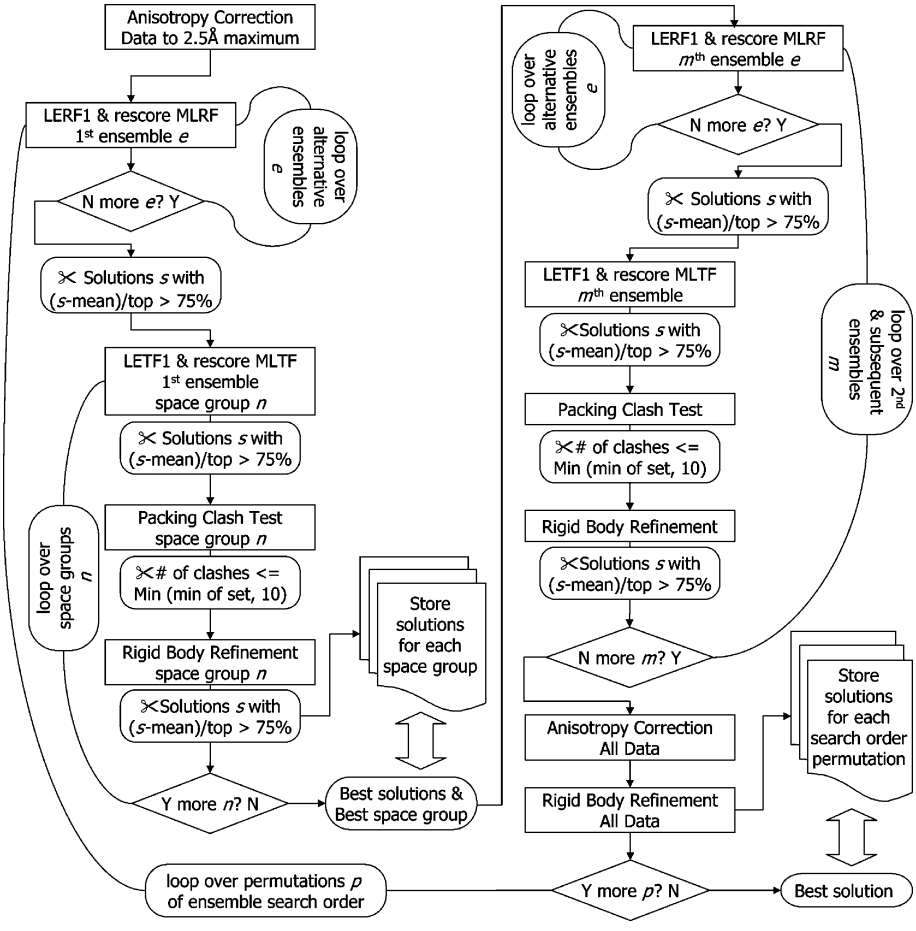

Figure 1
Flow diagram for automated molecular replacement in *Phaser*.

3.1. Automated molecular replacement

Most structures that can be solved by MR with *Phaser* can be solved using the 'automated molecular replacement' mode. The flow diagram for this mode is shown in Fig. 1. The search strategy automates four search processes: those for multiple components in the asymmetric unit, for ambiguity in the hand of the space group and/or other space groups in the same point group, for permutations in the search order for components (when there are multiple components), and for finding the best model when there is more than one possible model for a component.

3.1.1. Multiple components of asymmetric unit. Where there are many models to be placed in the asymmetric unit, the signal from the placement of the first model may be buried in noise and the correct placement of this first model only found in the context of all models being placed in the asymmetric unit. One way of tackling this problem has been to use stochastic methods to search the multi-dimensional space (Chang & Lewis, 1997; Kissinger *et al.*, 1999; Glykos & Kokkinidis, 2000). However, we have chosen to use a tree-search-with-pruning approach, where a list of possible placements of the first (and subsequent) models is kept until the placement of the final model. This tree-search-with-pruning search strategy can generate very branched searches that would be challenging for users to negotiate by running separate jobs, but becomes trivial with suitable automation. The search strategy exploits the strength of the maximum likelihood target functions in using prior information in the search for subsequent components in the asymmetric unit.

The tree-search-with-pruning strategy is heavily dependent on the criteria used for selecting the peaks that survive to the next round. Four selection criteria are available in *Phaser*: selection by percentage difference between the top and mean log-likelihood of the search, selection by Z score, selection by number of peaks, and selection of all peaks. The default is selection by percentage, with the default percentage set at 75%. This selection method has the advantage that, if there is one clear peak standing well above the noise, it alone will be passed to the next round, while if there is no clear signal, all peaks high in the list will be passed as potential solutions to the next round. If structure solution fails, it may be possible to rescue the solution by reducing the percentage cut-off used for selection from 75% to, for example, 65%, so that if the correct peak was just missing the default cut-off, it is now included in the list passed to the next round.

The tree-search-with-pruning search strategy is sub-optimal where there are multiple copies of the same search model in the asymmetric unit. In this case the search generates many branches, each of which has a subset of the complete solution, and so there is a combinatorial explosion in the search. The tree search would only converge onto one branch (solution) with the placement of the last component on each of the branches, but in practice the run time often becomes excessive and the job is terminated before this point can be reached. When searching for multiple copies of the same component in the asymmetric unit, several copies should be added at each search step (rather than branching at each search step), but this search strategy must currently be performed semi-manually as described elsewhere (McCoy, 2007).

3.1.2. Alternative space groups. The space group of a structure can often be ambiguous after data collection. Ambiguities of space group within the one point group may arise from theoretical considerations (if the space group has an enantiomorph) or on experimental grounds (the data along one or more axes were not collected and the systematic absences along these axes cannot be determined). Changing the space group of a structure to another in the same point group can be performed without re-indexing, merging or scaling the data. Determination of the space group within a point group is therefore an integral part of structure solution by MR. The translation function will yield the highest log-likelihood gain for a correctly packed solution in the correct space group. *Phaser* allows the user to make a selection of space groups within the same point group for the first translation function calculation in a search for multiple components in the asymmetric unit. If the signal from the placement of the first component is not significantly above noise, the correct space group may not be chosen by this protocol, and the search for all components in the asymmetric unit should be completed separately in all alternative space groups.

3.1.3. Alternative models. As the database of known structures expands, the number of potential MR models is also rapidly increasing. Each available model can be used as a separate search model, or combined with other aligned structures in an 'ensemble' model. There are also various ways of editing structures before use as MR models (Schwarzenbacher *et al.*, 2004). The number of MR trials that can be performed thus increases combinatorially with the number of potential models, which makes job tracking difficult for the user. In addition, most users stop performing MR trials as soon as any solution is found, rather than continuing the search until the MR solution with the greatest log-likelihood gain is found, and so they fail to optimize the starting point for subsequent steps in the structure solution pipeline.

The use of alternative models to represent a structure component is also useful where there are multiple copies of one type of component in the asymmetric unit and the different copies have different conformations due to packing differences. The best solution will then have the different copies modelled by different search models; if the conformation change is severe enough, it may not be possible to solve the structure without modelling the differences. A set of alternative search models may be generated using previously observed conformational differences among similar structures, or, for example, by normal-mode analysis (see §2.3).

Phaser automates searches over multiple models for a component, where each potential model is tested in turn before the one with the greatest log-likelihood gain is found. The loop over alternative models for a component is only implemented in the rotation functions, as the solutions passed from the rotation function to the translation function step explicitly specify which model to use as well as the orientation for the translation function in question.

3.1.4. Search order permutation. When searching for multiple components in the asymmetric unit, the order of the search can be a factor in success. The models with the biggest component of the total structure factor will be the easiest to find: when weaker scattering components are the subject of the initial search, the solution may be buried in noise and not significant enough to survive the selection criteria in the tree-search-with-pruning search strategy. Once the strongest scattering components are located, then the search for weaker scattering components (in the background of the strong scattering components) is more likely to be a success. Having a high component of the total structure factor correlates with

the model representing a high fraction of the total contents of the asymmetric unit, low r.m.s. deviation between model and target atoms, and low B factors for the target to which the model corresponds. Although the first of these (high completeness) can be determined in advance from the fraction of the total molecular weight represented by the model, the second can only be estimated from the Chothia & Lesk (1986) formula and the third is unknown in advance. If structure solution fails with the search performed in the order of the molecular weights, then other permutations of search order should be tried. In *Phaser*, this possibility is automated on request: the entire search strategy (except for the initial anisotropic data correction) is performed for all unique permutations of search orders.

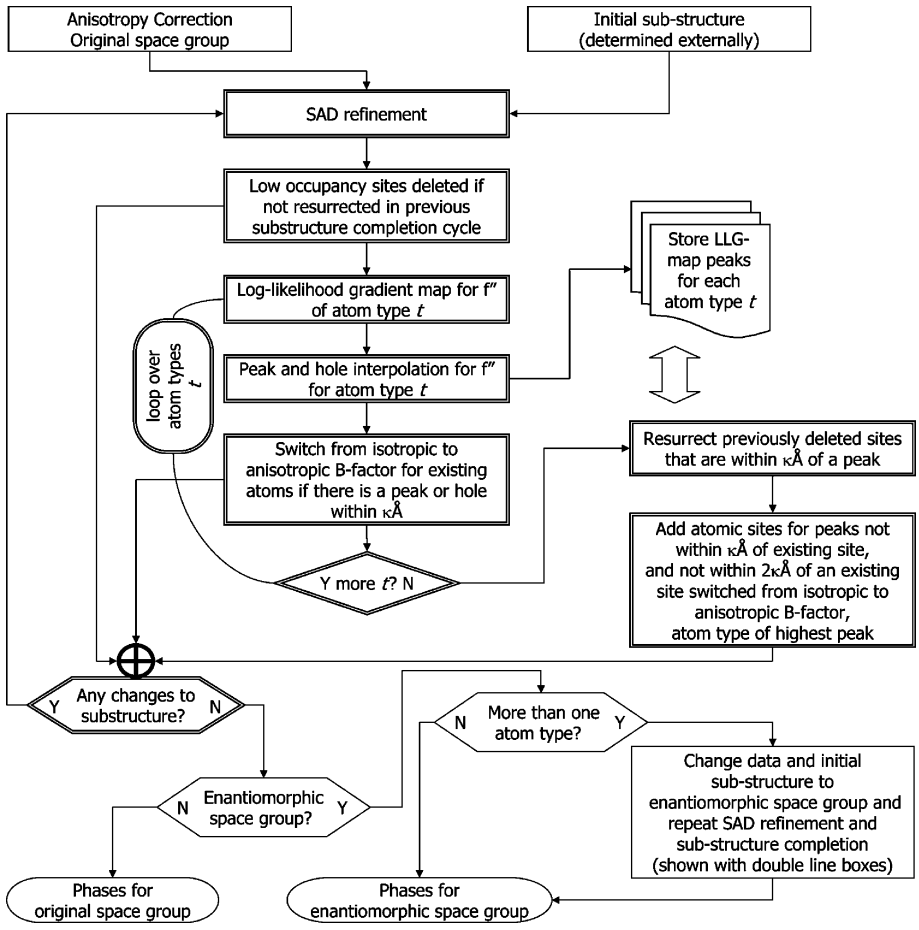

Figure 2
Flow diagram for automated experimental phasing in *Phaser*.

3.2. Automated experimental phasing

SAD is the simplest type of experimental phasing method to automate, as it involves only one crystal and one data set. SAD is now becoming the experimental phasing method of choice, overtaking multiple-wavelength anomalous dispersion because only a single data set needs to be collected. This can help minimize radiation damage to the crystal, which has a major adverse effect on the success of multi-wavelength experiments. The 'automated experimental phasing' mode in *Phaser* takes an atomic substructure determined by Patterson, direct or dual-space methods (Karle & Hauptman, 1956; Rossmann, 1961; Mukherjee *et al.*, 1989; Miller *et al.*, 1994; Sheldrick & Gould, 1995; Sheldrick *et al.*, 2001; Grosse-Kunstleve & Adams, 2003) and refines the positions, occupancies, B factors and values of the atoms to optimize the SAD function, then uses log-likelihood gradient maps to complete the atomic substructure. The flow diagram for this mode is shown in Fig. 2. The search strategy automates two search processes: those for ambiguity in the hand of the space group and for completing atomic substructure from log-likelihood gradient maps. A feature of using the SAD function for phasing is that the substructure need not only consist of anomalous scatterers; indeed it can consist of only real scatterers, since the real scattering of the partial structure is used as part of the phasing function. This allows structures to be completed from initial real scattering models.

3.2.1. Enantiomorphic space groups. Since the SAD phasing mode of *Phaser* takes as input an atomic substructure model, the space group of the solution has already been determined to within the enantiomorph of the correct space group. Changing the enantiomorph of a SAD refinement involves changing the enantiomorph of the heavy atoms, or in some cases the space group (*e.g.* the enantiomorphic space group of $P4_1$ is $P4_3$). In some rare cases (*Fdd*2, $I4_1$, $I4_122$, $I4_1md$, $I4_1cd$, $I\bar{4}2d$, $F4_132$; Koch & Fischer, 1989) the origin of the heavy-atom sites is changed [*e.g.* the enantiomorphic space group of $I4_1$ is $I4_1$ with the origin shifted to $(\frac{1}{2}, 0, 0)$]. If there is only one type of anomalous scatterer, the refinement need not be repeated in both hands: only the phasing needs to be carried out in the second hand to be considered. However, if there is more than one type of anomalous scatterer, then the refinement and substructure completion needs to be repeated, as it will not be enantiomorphically symmetric in the other hand. To facilitate this, *Phaser* runs the refinement and substructure completion in both hands [as does other experimental phasing software, *e.g. Solve* (Terwilliger & Berendzen, 1999) and *autosharp* (Vonrhein *et al.*, 2006)]. The correct space group can then be found by inspection of the electron density maps; the density will only be interpretable in the correct space group. In cases with significant contributions from at least two types of anomalous scatterer in the substructure, the correct space group can also be identified by the log-likelihood gain.

3.2.2. Completing the substructure. Peaks in log-likelihood gradient maps indicate the coordinates at which new atoms should be added to improve the log-likelihood gain. In the initial maps, the peaks are likely to indicate the positions of the strongest anomalous scatterers that are missing from the model. As the phasing improves, weaker anomalous scatterers, such as intrinsic sulfurs, will appear in the log-likelihood gradient maps, and finally, if the phasing is exceptional and the resolution high, non-anomalous scatterers will appear, since the SAD function includes a contribution from the real scattering.

After refinement, atoms are excluded from the substructure if their occupancy drops below a tenth of the highest occupancy amongst those atoms of the same atom type (and therefore f''). Excluded sites are flagged rather than permanently deleted, so that if a peak later appears in the log-likelihood gradient map at this position, the atom can be reinstated and prevented from being deleted again, in order to prevent oscillations in the addition of new sites between cycles and therefore lack of convergence of the substructure completion algorithm.

New atoms are added automatically after a peak and hole search of the log-likelihood gradient maps. The cut-off for the consideration of a peak as a potential new atom is that its Z score be higher than 6 (by default) and also higher than the depth of the largest hole in the map, *i.e.* the largest hole is taken as an additional indication of the noise level of the map. The proximity of each potential new site to previous atoms is then calculated. If a peak is more than a cut-off distance (κ Å) of a previous site, the peak is added as a new atom with the average occupancy and B factor from the current set of sites. If the peak is within κ Å of an isotropic atom already present, the old atom is made anisotropic. Holes in the log-likelihood gradient map within κ Å of an isotropic atom also cause the atom's B factor to be switched to anisotropic. However, if the peak or hole is within κ Å of an anisotropic atom already present, the peak or hole is ignored. If a peak is within κ Å of a previously excluded site, the excluded site is reinstated and flagged as not for deletion in order to prevent oscillations, as described above. At the end of the cycle of atom addition and isotropic to anisotropic atomic B-factor switching, new sites within 2κ Å of an old atom that is now anisotropic are then removed, since the peak may be absorbed by refining the anisotropic B factor; if not, it will be accepted as a new site in the next cycle of log-likelihood gradient completion. The distance κ may be input directly by the user, but by default it is the 'optical resolution' of the structure ($\kappa = 0.715 d_{\rm min}$), but not less than 1 Å and no more than 10 Å.

If the structure contains more than one significant anomalous scatterer, then log-likelihood gradient maps are calculated from each atom type, the maps compared and the atom type associated with each significant peak assigned from the map with the most significant peak at that location.

3.2.3. Initial real scattering model. One of the reasons for including MR and SAD phasing within one software package is the ability to use MR solutions with the SAD phasing target to improve the phases. Since the SAD phasing target contains a contribution from the real scatterers, it is possible to use a partial MR model with no anomalous scattering as the initial atomic substructure used for SAD phasing. This approach is useful where there is a poor MR solution combined with a poor anomalous signal in the data. If the poor MR solution means that the structure cannot be phased from this model alone, and the poor anomalous signal means that the anomalous scatterers cannot be located in the data alone, then using the MR solution as the starting model for SAD phasing may provide enough phase information to locate the anomalous scatterers. The combined phase information will be stronger than from either source alone. To facilitate this method of structure solution, *Phaser* allows the user to input a partial structure model that will be interpreted in terms of its real scattering only and, following phasing with this substructure, to complete the anomalous scattering model from log-likelihood gradient maps as described above.

3.3. Input and output

The fastest and most efficient way, in terms of development time, to link software together is using a scripting language, while using a compiled language is most efficient for intensive computation. Following the lead of the *PHENIX* project (Adams *et al.*, 2002, 2004), *Phaser* uses Python (http://python.org) as the scripting language, C++ as the compiled language, and the Boost.Python library (http://boost.org/libs/python/) for linking C++ and Python. Other packages, notably *X-PLOR* (Brünger, 1993) and *CNS* (Brünger *et al.*, 1998), have defined their own scripting languages, but the choice of Python ensures that the scripting language is maintained by an active community. *Phaser* functionality has mostly been made available to Python at the 'mode' level. However, some low-level SAD refinement routines in *Phaser* have been made available to Python directly, so that they can be easily incorporated into phenix.refine.

A long tradition of *CCP4* keyword-style input in established macromolecular crystallography software (almost exclusively written in Fortran) means that, for many users, this has been the familiar method of calling crystallographic software and is preferred to a Python interface. The challenge for the development of *Phaser* was to find a way of satisfying both keyword-style input and Python scripting with minimal increase in development time. Taking advantage of the C++ class structure allowed both to be implemented with very little additional code. Each keyword is managed by its own class. The input to each mode of *Phaser* is controlled by Input objects, which are derived from the set of keyword classes appropriate to the mode. The keyword classes are in turn derived from a CCP4base class containing the functionality for the keyword-style input. Each keyword class has a parse routine that calls the CCP4base class functions to parse the keyword input, stores the input parameters as local variables and then passes these parameters to a keyword class set function. The keyword class set functions check the validity and consistency of the input, throw errors where appropriate and finally set the keyword class's member parameters.

Alternatively, the keyword class set functions can be called directly from Python. These keyword classes are a standalone part of the *Phaser* code and have already been used in other software developments (*Pointless*; Evans, 2006).

An Output object controls all text output from *Phaser* sent to standard output and to text files. Switches on the Output object give different output styles: *CCP4*-style for compatibility with *CCP4* distribution, *PHENIX*-style for compatibility with the *PHENIX* interface, CIMR-style for development, XML-style output for developers of automation scripts and a 'silent running' option to be used when running *Phaser* from Python. In addition to the text output, where possible *Phaser* writes results to files in standard format; coordinates to 'pdb' files and reflection data (*e.g.* map coefficients) to 'mtz' files. Switches on the Output object control the writing of these files.

3.3.1. *CCP4*-style output. CCP4-style output is a text log file sent to standard output. While this form of output is easily comprehensible to users, it is far from ideal as an output style for automation scripts. However, it is the only output style available from much of the established software that developers wish to use in their automation scripts, and it is common to use Unix tools such as 'grep' to extract key information. For this reason, the log files of *Phaser* have been designed to help developers who prefer to use this style of output. *Phaser* prints four levels of log file, summary, log, verbose and debug, as specified by user input. The important output information is in all four levels of file, but it is most efficient to work with the summary output. *Phaser* prints 'SUCCESS' and 'FAILURE' at the end of the log file to demarcate the exit state of the program, and also prints the names of any of the other output files produced by the program to the summary output, amongst other features.

3.3.2. XML output. XML is becoming commonly used as a way of communicating between steps in an automation pipeline, because XML output can be added very simply by the program author and relatively simply by others with access to the source code. For this reason, *Phaser* also outputs an XML file when requested. The XML file encapsulates the mark-up within ⟨phaser⟩ tags. As there is no standard set of XML tags for crystallographic results, *Phaser*'s XML tags are mostly specific to *Phaser* but were arrived at after consultation with other developers of XML output for crystallographic software.

3.3.3. Python interface. The most elegant and efficient way to run *Phaser* as part of an automation script is to call the functionality directly from Python. Using *Phaser* through the Python interface is similar to using *Phaser* through the keyword interface. Each mode of operation of *Phaser* described above is controlled by an Input object and its parameter set functions, which have been made available to Python with the Boost.Python library. *Phaser* is then run with a call to the 'run-job' function, which takes the Input object as a parameter. The 'run-job' function returns a Result object on completion, which can then be queried using its get functions. The Python Result object can be stored as a 'pickled' class structure directly to disk. Text is not sent to standard out in the *CCP4* logfile way but may be redirected to another output stream. All Input and Result objects are fully documented.

4. Future developments

Phaser will continue to be developed as a platform for implementing novel phasing algorithms and bringing the most effective approaches to the crystallographic community. Much work remains to be done formulating maximum likelihood functions with respect to noncrystallographic symmetry, to account for correlations in the data and to consider non-isomorphism, all with the aim of achieving the best possible initial electron density map.

After a generation in which Fortran dominated crystallographic software code, C++ and Python have become the new standard. Several developments, including *Phaser*, *PHENIX* (Adams *et al.*, 2002, 2004), *Clipper* (Cowtan, 2002) and *mmdb* (Krissinel *et al.*, 2004), simultaneously chose C++ as the compiled language at their inception at the turn of the millennium. At about the same time, Python was chosen as a scripting language by *PHENIX*, *ccp4mg* (Potterton *et al.*, 2002, 2004) and *PyMol* (DeLano, 2002), amongst others. Since then, other major software developments have also started or converted to C++ and Python, for example *PyWarp* (Cohen *et al.*, 2004), *MrBump* (Keegan & Winn, 2007) and *Pointless* (Evans, 2006). The choice of C++ for software development was driven by the availability of free compilers, an ISO standard (International Standardization Organization *et al.*, 1998), sophisticated dynamic memory management and the inherent strengths of using an object-oriented language. Python was equally attractive because of the strong community support, its object-oriented design, and the ability to link C++ and Python through the Boost.Python library or the SWIG library (http://www.swig.org/). Now that a 'critical mass' of developers has taken to using the new languages, C++ and Python are likely to remain the standard for crystallographic software for the current generation of crystallographic software developers.

Phaser source code has been distributed directly by the authors (see http://www-structmed.cimr.cam.ac.uk/phaser for details) and through the *PHENIX* and *CCP4* (Collaborative Computing Project, Number 4, 1994) software suites. The source code is released for several reasons, including that we believe source code is the most complete form of publication for the algorithms in *Phaser*. It is hoped that generous licensing conditions and source distribution will encourage the use of *Phaser* by other developers of crystallographic software and those writing crystallographic automation scripts. There are no licensing restrictions on the use of *Phaser* in macromolecular crystallography pipelines by other developers, and the license conditions even allow developers to alter the source code (although not to redistribute it). We welcome suggestions for improvements to be incorporated into new versions.

Compilation of *Phaser* requires the computational crystallography toolbox (*cctbx*; Grosse-Kunstleve & Adams, 2003), which includes a distribution of the cmtz library (Winn *et al.*, 2002). The Boost libraries (http://boost.org/) are required for

access to the functionality from Python. *Phaser* runs under a wide range of operating systems including Linux, Irix, OSF1/Tru64, MacOS-X and Windows, and precompiled executables are available for these platforms when only keyword-style access (and not Python access) is required. Graphical user interfaces to *Phaser* are available for both the *PHENIX* and the *CCP4* suites. User support is available through *PHENIX*, *CCP4* and from the authors (email cimr-phaser@lists.cam.ac.uk).

We thank Anne Baker for the bulk of the development of the CCP4 MR GUI for *Phaser*, Tom Terwilliger for much of the development of the Phenix AutoMR wizard, Richard Francis for writing the distribution cgi scripts, and the many users who have provided invaluable feedback. This work was funded by a Principal Research Fellowship from the Wellcome Trust (RJR) and by NIH/NIGMS under grant No. 1P01GM063210.

References

Adams, P. D., Gopal, K., Grosse-Kunstleve, R. W., Hung, L.-W., Ioerger, T. R., McCoy, A. J., Moriarty, N. W., Pai, R. K., Read, R. J., Romo, T. D., Sacchettini, J. C., Sauter, N. K., Storoni, L. C. & Terwilliger, T. C. (2004). *J. Synchrotron Rad.* **11**, 53–55.

Adams, P. D., Grosse-Kunstleve, R. W., Hung, L.-W., Ioerger, T. R., McCoy, A. J., Moriarty, N. W., Read, R. J., Sacchettini, J. C., Sauter, N. K. & Terwilliger, T. C. (2002). *Acta Cryst.* D**58**, 1948–1954.

Bahar, I., Atilgan, A. R. & Erman, B. (1997). *Folding Des.* **2**, 173–181.

Bricogne, G. & Irwin, J. (1996). *Macromolecular Refinement: Proceedings of the CCP4 Study Weekend*, edited by E. Dodson, M. Moore, A. Ralph & S. Bailey, pp. 85–92. Warrington: Daresbury Laboratory.

Brünger, A. T. (1993). *X-Plor*. Version 3.1. *System for X-ray Crystallography and NMR*. Yale University Press.

Brünger, A. T., Adams, P. D., Clore, G. M., DeLano, W. L., Gros, P., Grosse-Kunstleve, R. W., Jiang, J.-S., Kuszewski, J., Nilges, M., Pannu, N. S., Read, R. J., Rice, L. M., Simonson, T. & Warren, G. L. (1998). *Acta Cryst.* D**54**, 905–921.

Burley, S. K., Almo, S. C., Bonanno, J. B., Capel, M., Chance, M. R., Gaasterland, T., Lin, D., Sali, A., Studier, F. W. & Swaminathan, S. (1999). *Nature Genet.* **23**, 151–157.

Chang, G. & Lewis, M. (1997). *Acta Cryst.* D**53**, 279–289.

Chothia, C. & Lesk, A. M. (1986). *EMBO J.* **5**, 823–826.

Cohen, S. X., Morris, R. J., Fernandez, F. J., Ben Jelloul, M., Kakaris, M., Parthasarathy, V., Lamzin, V. S., Kleywegt, G. J. & Perrakis, A. (2004). *Acta Cryst.* D**60**, 2222–2229.

Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* D**50**, 760–763.

Cowtan K. (2002). *Handling Reflection Data using the Clipper Libraries*, CCP4 Newsletter 41.

Crowther, R. A. (1972). *The Molecular Replacement Method*, edited by M. G. Rossmann, pp. 173–178. New York: Gordon and Breach.

Dauter, Z., Betzel, C., Genov, N., Pipon, N. & Wilson, K. S. (1991). *Acta Cryst.* B**47**, 707–730.

DeLano, W. L. (2002). *The PyMOL Molecular Graphics System*. DeLano Scientific, San Carlos, CA, USA.

Durand, P., Trinquier, G. & Sanejouand, Y. H. (1994). *Biopolymers*, **34**, 759–771.

Edwards, A. W. F. (1992). *Likelihood*. Baltimore: Johns Hopkins University Press.

Evans, P. (2006). *Acta Cryst.* D**62**, 72–82.

Fisher, R. A. (1922). *Philos. Trans. R. Soc. A*, **222**, 309–368.

Giacovazzo, C. (1992). Editor. *Fundamentals of Crystallography*. IUCr/Oxford University Press.

Giacovazzo, C. (1998). *Direct Phasing in Crystallography: Fundamentals and Applications*. IUCr/Oxford University Press.

Glykos, N. M. & Kokkinidis, M. (2000). *Acta Cryst.* D**56**, 169–174.

Green, E. A. (1979). *Acta Cryst.* A**35**, 351–359.

Grosse-Kunstleve, R. W. & Adams, P. D. (2002). *J. Appl. Cryst.* **35**, 477–480.

Grosse-Kunstleve, R. W. & Adams, P. D. (2003). *Acta Cryst.* D**59**, 1966–1973.

Hinsen, K. (1998). *Proteins*, **33**, 417–429.

International Standardization Organization (ISO), International Electrotechnical Commission (IEC), American National Standards Institute (ANSI) and Information Technology Industry Council (ITI) (1998). International Standard ISO/IEC 14882, 1st ed., Information Technology Industry Council, 1250 Eye Street NW, Washington, DC 20005, USA (also available at http://webstore.ansi.org/).

Johnson, R. A. & Wichern, D. W. (1998). *Applied Multivariate Statistical Analysis*, 4th ed. New Jersey: Prentice-Hall.

Karle, J. & Hauptman, H. (1956). *Acta Cryst.* **9**, 635–651.

Keegan, R. M. & Winn, M. D. (2007). *Acta Cryst.* D**63**, 447–457.

Kissinger, C. R., Gehlhaar, D. K. & Fogel, D. B. (1999). *Acta Cryst.* D**55**, 484–491.

Kleywegt, G. J. & Read, R. J. (1997). *Structure*, **5**, 1557–1569.

Koch, E. & Fischer, W. (1989). *International Tables for Crystallography*, Vol. A, edited by T. Hahn, pp. 855–869. Dordrecht: IUCr/Kluwer Academic Publishers.

Krissinel, E. B., Winn, M. D., Ballard, C. C., Ashton, A. W., Patel, P., Potterton, E. A., McNicholas, S. J., Cowtan, K. D. & Emsley, P. (2004). *Acta Cryst.* D**60**, 2250–2255.

La Fortelle, E. de & Bricogne, G. (1997). *Methods Enzymol.* **276**, 472–494.

Lattman, E. E. & Love, W. E. (1970). *Acta Cryst.* B**26**, 1854–1857.

Luzzati, V. (1952). *Acta Cryst.* **5**, 802–810.

McCoy, A. J. (2004). *Acta Cryst.* D**60**, 2169–2183.

McCoy, A. J. (2007). *Acta Cryst.* D**63**, 32–41.

McCoy, A. J., Grosse-Kunstleve, R. W., Storoni, L. C. & Read, R. J. (2005). *Acta Cryst.* D**61**, 458–464.

McCoy, A. J., Storoni, L. C. & Read, R. J. (2004). *Acta Cryst.* D**60**, 1220–1228.

Miller, R., Gallo, S. M., Khalak, H. G. & Weels, C. M. (1994). *J. Appl. Cryst.* **27**, 613–621.

Mukherjee, A. K., Helliwell, J. R. & Main, P. (1989). *Acta Cryst.* A**45**, 715–718.

Murshudov, G. N., Vagin, A. A. & Dodson, E. J. (1997). *Acta Cryst.* D**53**, 240–255.

Murshudov, G. N., Vagin, A. A., Lebedev, A., Wilson, K. S. & Dodson, E. J. (1999). *Acta Cryst.* D**55**, 247–255.

Navaza, J. (1994). *Acta Cryst.* A**50**, 157–163.

Navaza, J. & Vernoslova, E. (1995). *Acta Cryst.* A**51**, 445–449.

Nordman, C. E. (1994). *Acta Cryst.* A**50**, 68–72.

Pannu, N. S. & Read, R. J. (2004). *Acta Cryst.* D**60**, 22–27.

Popov, A. N. & Bourenkov, G. P. (2003). *Acta Cryst.* D**59**, 1145–1153.

Potterton, E., McNicholas, S., Krissinel, E., Cowtan, K. & Noble, M. (2002). *Acta Cryst.* D**58**, 1955–1957.

Potterton, L., McNicholas, S., Krissinel, E., Gruber, J., Cowtan, K., Emsley, P., Murshudov, G. N., Cohen, S., Perrakis, A. & Noble, M. (2004). *Acta Cryst.* D**60**, 2288–2294.

Read, R. J. (2001). *Acta Cryst.* D**57**, 1373–1382.

Rossmann, M. G. (1961). *Acta Cryst.* **14**, 383–388.

Schwarzenbacher, R., Godzik, A., Grzechnik, S. K. & Jaroszewski, L. (2004). *Acta Cryst.* D**60**, 1229–1236.

Sheldrick, G. M. (1995). *SHELXL93*. Institut für Anorganische Chemie, Göttingen, Germany.

Sheldrick, G. M. & Gould, R. O. (1995). *Acta Cryst.* B**51**, 423–431.

Sheldrick, G. M., Hauptman, H. A., Weeks, C. M., Miller, R. & Usoń, I. (2001). *International Tables for Crystallography*, Vol. F, edited by

M. G. Rossmann & E. Arnold, pp. 233–245. Dordrecht: Kluwer Academic Publishers.
Sim, G. A. (1959). *Acta Cryst.* **12**, 813–815.
Storoni, L. C., McCoy, A. J. & Read, R. J. (2004). *Acta Cryst.* D**60**, 432–438.
Suhre, K. & Sanejouand, Y.-H. (2004). *Acta Cryst.* D**60**, 796–799.
Tama, F., Gadea, F. X., Marques, O. & Sanejouand, Y. H. (2000). *Proteins Struct. Funct. Genet.* **41**, 1–7.
Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**, 849–861.
Trueblood, K. N., Bürgi, H.-B., Burzlaff, H., Dunitz, J. D., Gramaccioli, C. M., Schulz, H. H., Shmueli, U. & Abrahams, S. C. (1996). *Acta Cryst.* A**52**, 770–781.
Vagin, A. & Teplyakov, A. (1997). *J. Appl. Cryst.* **30**, 1022–1025.
Vonrhein, C., Blanc, E., Roversi, P. & Bricogne, G. (2006). *Macromolecular Crystallography Protocols*, Vol. 2, edited by S. Doublié. Totowa, NJ, USA: Humana Press.
Wilson, A. J. C. (1949). *Acta Cryst.* **2**, 318–321.
Winn, M. D., Ashton, A. W., Briggs, P. J., Ballard, C. C. & Patel, P. (2002). *Acta Cryst.* D**58**, 1929–1936.
Wooding, R. A. (1956). *Biometrika*, **43**, 212–215.
Woolfson, M. M. (1956). *Acta Cryst.* **9**, 804–810.
Zhang, X.-J. & Matthews, B. W. (1994). *Acta Cryst.* D**50**, 675–686.