Acta Cryst A FOUNDATIONS ADVANCES

ISSN 2053-2733

Efficient structure-factor modeling for crystals with multiple components

Pavel V. Afonine,[a]* Paul D. Adams[a,b] and Alexandre G. Urzhumtsev[c,d]

[a]Molecular Biophysics and Integrated Bioimaging Department, Lawrence Berkeley National Laboratory, One Cyclotron Road, Berkeley, California 94720, USA, [b]Department of Bioengineering, University of California Berkeley, Berkeley, California, USA, [c]Centre for Integrative Biology, Institut de Génétique et de Biologie Moléculaire et Cellulaire, CNRS-INSERM-UdS, 1 rue Laurent Fries, BP 10142, Illkirch, 67404, France, and [d]Faculté des Sciences et Technologies, Université de Lorraine, BP 239, Vandoeuvre-les-Nancy, 54506, France. *Correspondence e-mail: pafonine@lbl.gov

Received 13 December 2022
Accepted 18 April 2023

Edited by P. M. Dominiak, University of Warsaw, Poland

Keywords: structure factors; multiple components; scattering functions; bulk solvent; refinement; density maps.

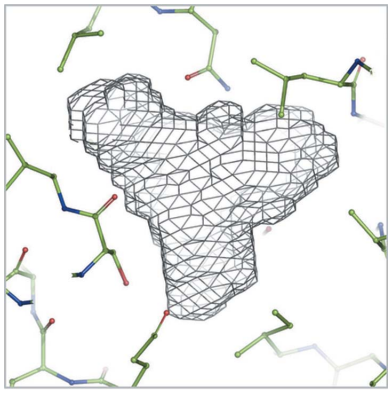
Diffraction intensities from a crystallographic experiment include contributions from the entire unit cell of the crystal: the macromolecule, the solvent around it and eventually other compounds. These contributions cannot typically be well described by an atomic model alone, *i.e.* using point scatterers. Indeed, entities such as disordered (bulk) solvent, semi-ordered solvent (*e.g.* lipid belts in membrane proteins, ligands, ion channels) and disordered polymer loops require other types of modeling than a collection of individual atoms. This results in the model structure factors containing multiple contributions. Most macromolecular applications assume two-component structure factors: one component arising from the atomic model and the second one describing the bulk solvent. A more accurate and detailed modeling of the disordered regions of the crystal will naturally require more than two components in the structure factors, which presents algorithmic and computational challenges. Here an efficient solution of this problem is proposed. All algorithms described in this work have been implemented in the computational crystallography toolbox (*CCTBX*) and are also available within *Phenix* software. These algorithms are rather general and do not use any assumptions about molecule type or size nor about those of its components.

1. Introduction

Experimentally measured intensities of the crystallographic structure factors reflect the content of the whole crystal. Therefore, accurate modeling of the crystal content requires corresponding structure factors to account for all scattering matter present in the unit cell. This includes bulk solvent and other semi-ordered or disordered entities, such as disordered loops or ligands. Currently, crystallographic packages such as *SHELXL* (Sheldrick, 2008), *CNS* (Brünger *et al.*, 1998), *REFMAC* (Murshudov *et al.*, 2011), *Phenix* (Liebschner *et al.*, 2019) employ the two-component model for the total structure factor:

$$\mathbf{F}_{\rm model}(\mathbf{s}) = k_{\rm total}(\mathbf{s})\big[\mathbf{F}_{\rm calc}(\mathbf{s}) + \mathbf{F}_{\rm bulk}(\mathbf{s})\big]. \quad (1)$$

Here $\mathbf{F}_{\rm calc}(\mathbf{s})$ is the contribution from all ordered atoms (macromolecule, solvent, ligands) and $\mathbf{s}$ represents a reciprocal-space vector. $\mathbf{F}_{\rm bulk}(\mathbf{s})$ accounts for the bulk solvent contribution using one of the available models: exponential (Moews & Kretsinger, 1975; Tronrud, 1997), radial-shell (Jiang & Brünger, 1994), flat with exponential (Jiang & Brünger, 1994) or per-resolution scalar scale (Afonine *et al.*, 2013). $k_{\rm total}(\mathbf{s})$ is the overall anisotropic resolution-dependent scale factor. A similar approach, referred to as *PLATON SQUEEZE* (Spek, 2015), is used in small-molecule crystallography, where the contribution of the disordered content of



the unit cell is explicitly calculated and added to the total model structure factors. *BUSTER* (Roversi *et al.*, 2000; Blanc *et al.*, 2004) uses the three-component model

$$\mathbf{F}_{\rm model}(\mathbf{s}) = k_{\rm total}(\mathbf{s})\big[\mathbf{F}_{\rm calc}(\mathbf{s}) + \mathbf{F}_{\rm bulk}(\mathbf{s}) + \mathbf{F}_{\rm miss}(\mathbf{s})\big], \qquad (2)$$

where $\mathbf{F}_{\rm miss}(\mathbf{s})$ describes components other than bulk solvent that cannot be modeled with individual atoms (such as the disordered part of a macromolecule or ligands).

Below we propose a more general definition of the total model structure factor:

$$\mathbf{F}_{\rm model}(\mathbf{s}) = k_{\rm total}(\mathbf{s})\left[\mathbf{F}_{\rm calc}(\mathbf{s}) + \sum_{n=1}^{N} k_n(\mathbf{s})\mathbf{F}_n(\mathbf{s})\right]. \qquad (3)$$

Here $\mathbf{F}_{\rm calc}(\mathbf{s})$ are calculated on the absolute scale from the principal part of the model, *e.g.* atomic model. Terms $\mathbf{F}_n(\mathbf{s})$ stand for structure factors arising from other (for example, non-atomic) components added to the sum with some scale factors $k_n(\mathbf{s})$. In the simplest case where there is no prior knowledge available about these non-atomic components, $\mathbf{F}_n(\mathbf{s})$ can be the structure factors calculated from a binary 0–1 mask of the component n, with 1 inside the region and 0 outside, similar to the flat bulk solvent model (Jiang & Brünger, 1994). However, any other model considering the contribution from different parts of the crystal as independent is applicable. When some prior information is available, then more sophisticated $\mathbf{F}_n(\mathbf{s})$ models can be used (Blanc *et al.*, 2004). The number N is not specific for the algorithms and is defined by a particular problem. Practically, we expect it to vary from a few up to several hundreds.

The values of the resolution-dependent scale factors $k_{\rm total}(\mathbf{s})$ and $k_n(\mathbf{s})$ can be obtained by fitting $\mathbf{F}_{\rm model}(\mathbf{s})$ to the observed structure-factor amplitudes $F_{\rm obs}(\mathbf{s})$. At this stage, we consider all structure factors as constants and search only for the scale factors.

When $N = 1$, *i.e.* when a single bulk solvent contribution is considered, a possible solution has been reported in detail and implemented in *CCTBX* and *Phenix* (Afonine *et al.*, 2013). When $N > 1$, a fast, robust and memory-efficient algorithm is needed. Here we propose four possible algorithms, discuss the strengths and weaknesses of each of them, and argue for one to be used as a default choice.

2. Methods

2.1. Common considerations

Assuming $k_0(\mathbf{s}) = 1$ and denoting $\mathbf{F}_0(\mathbf{s}) = \mathbf{F}_{\rm calc}(\mathbf{s})$, expression (3) can be rewritten as

$$\mathbf{F}_{\rm model}(\mathbf{s}) = k_{\rm total}(\mathbf{s}) \sum_{n=0}^{N} k_n(\mathbf{s})\mathbf{F}_n(\mathbf{s}). \qquad (4)$$

Here $k_{\rm total}(\mathbf{s})$ is the unknown overall anisotropic scale factor (Sheriff & Hendrickson, 1987; Afonine *et al.*, 2013), $k_0(\mathbf{s}) = 1$ and $k_n(\mathbf{s})$ for $n > 0$ are unknown scale functions. We suppose that $k_n(\mathbf{s})$ are smooth isotropic functions of the resolution, *i.e.* $k_n(s)$ where $s = |\mathbf{s}|$. No particular analytical shape is assumed for $k_n(s)$, as argued by Urzhumtsev & Podjarny (1995) and Afonine *et al.* (2013).

The functions $k_n(s)$ vary slowly within sufficiently thin resolution shells. The resolution shells are defined uniformly in the logarithmic resolution scale (Urzhumtsev *et al.*, 2009; Table 1 in Afonine *et al.*, 2013) with two additional and somewhat contradictory requirements: the shells should be thin enough to consider scale factors $k_n(s)$ as constant inside each shell and they should contain a sufficient number of reflections to make determination of $k_n(s)$ values statistically valid. The latter condition concerns mostly the lowest-resolution shells.

If all the N components have the same scattering function (form factor), then (4) can be simplified,

$$\mathbf{F}_{\rm model}(\mathbf{s}) = k_{\rm total}(\mathbf{s})\left[\mathbf{F}_0(\mathbf{s}) + k(s) \sum_{n=1}^{N} k_n\mathbf{F}_n(\mathbf{s})\right], \qquad (5)$$

where scale factors k_n are independent of resolution and can be thought of as occupancy factors of respective components, and $k(\mathbf{s})$ is an overall resolution-dependent scale factor for all the components. An advantage of (5) with respect to (4) is that it uses a single parameter k_n for all structure factors $\mathbf{F}_n(\mathbf{s})$, and the total number of independent parameters reduces from $(N + 1)M_{\rm shells}$ to $N + 2M_{\rm shells}$, where $M_{\rm shells}$ is the number of resolution shells.

2.2. Initialization

The scaling procedure is iterative and initiated with the observed structure-factor amplitudes or intensities, $F_{\rm obs}(\mathbf{s})$ or $I_{\rm obs}(\mathbf{s})$, and a set of $\mathbf{F}_n(\mathbf{s})$. The initial values of $k_{\rm total}(\mathbf{s})$ and of $k_1(s) = k_2(s) = \ldots = k_N(s)$ are obtained as described by Afonine *et al.* (2013) considering contributions from all non-atomic components as a single one. Once all components $\mathbf{F}_n(\mathbf{s})$ are accounted for, the overall scale factor $k_{\rm total}(\mathbf{s})$ can be updated.

Observed amplitudes $F_{\rm obs}(\mathbf{s})$ or intensities $I_{\rm obs}(\mathbf{s})$ and scaled $\tilde{\mathbf{F}}_n(\mathbf{s}) = \mathbf{F}_n(\mathbf{s}) \times k_{\rm total}(\mathbf{s})$ are the inputs to each of four algorithms, referred to below as algorithms 1–4. Then, calculations of improved $k_n(s)$ values are performed independently in resolution shells. The procedure is repeated iteratively, until convergence, with $k_{\rm total}(\mathbf{s})$ and $k_n(s)$ being updated at each iteration.

In what follows, to simplify expressions, we omit the index of the resolution shell when this does not lead to confusion.

2.3. Algorithms to search for the scale coefficients

2.3.1. Algorithm 1: sequential search. In algorithm 1 each component $\mathbf{F}_n(\mathbf{s})$ is added to $\mathbf{F}_{\rm model}(\mathbf{s})$ sequentially one at a time followed by the update of $k_{\rm total}(\mathbf{s})$. For each new $\mathbf{F}_n(\mathbf{s})$ that is being added the scale factors $k_n(s)$ are computed as described by Afonine *et al.* (2013). This means that at each iteration the procedure of Afonine *et al.* (2013) is applied N times, equal to the number of components, which makes the procedure very expensive computationally. Also, errors in initially roughly estimated parameters such as $k_{\rm total}(\mathbf{s})$ can

propagate into $k_n(s)$ of components being added and that can result in the failure of the whole procedure.

2.3.2. Algorithm 2: iterative one-step search. Considering all coefficients k_n in each resolution shell as constants, this algorithm searches simultaneously for their values, minimizing the residual

$$\begin{aligned} LS_I &= \frac{1}{4}\sum_{\mathbf{s}}\left[I_{\rm model}(\mathbf{s}) - I_{\rm obs}(\mathbf{s})\right]^2 \\ &= \frac{1}{4}\sum_{\mathbf{s}}\left\{\left[\sum_{n=0}^{N} k_n\tilde{\mathbf{F}}_n(\mathbf{s})\right]\left[\sum_{m=0}^{N} k_m\tilde{\mathbf{F}}^*_m(\mathbf{s})\right] - I_{\rm obs}(\mathbf{s})\right\}^2 \end{aligned} \quad (6)$$

with respect to k_n. Here the outer sums are calculated over reflections of the given shell. Developing the expression in curly brackets and swapping the sums over components and over reflections, this expression can be rewritten as

$$\begin{aligned} LS_I &= \frac{1}{4}\sum_{\mathbf{s}}\left\{\left[\sum_{n=0}^{N}\sum_{m=0}^{N} k_n\tilde{\mathbf{F}}_n(\mathbf{s})k_m\tilde{\mathbf{F}}^*_m(\mathbf{s})\right] - I_{\rm obs}(\mathbf{s})\right\}^2 \\ &= \frac{1}{4}\sum_{\mathbf{s}}\left\{\left[\sum_{n=0}^{N}\sum_{m=0}^{N} k_nk_mG_{nm}(\mathbf{s})\right] - I_{\rm obs}(\mathbf{s})\right\}^2 \\ &= \frac{1}{4}\sum_{\mathbf{s}}\left\{\left[\sum_{n=0}^{N}\sum_{m=0}^{N} k_nk_mG_{nm}(\mathbf{s})\right]^2\right. \\ &\quad \left. - 2I_{\rm obs}(\mathbf{s})\left[\sum_{n=0}^{N}\sum_{m=0}^{N} k_nk_mG_{nm}(\mathbf{s})\right] + I^2_{\rm obs}(\mathbf{s})\right\} \\ &= \frac{1}{4}\sum_{n=0}^{N}\sum_{m=0}^{N}\sum_{j=0}^{N}\sum_{l=0}^{N} k_nk_mk_jk_l\left[\sum_{\mathbf{s}} G_{jl}(\mathbf{s})G_{nm}(\mathbf{s})\right] \\ &\quad - \frac{1}{2}\sum_{n=0}^{N}\sum_{m=0}^{N} k_nk_m\left[\sum_{\mathbf{s}} G_{nm}(\mathbf{s})I_{\rm obs}(\mathbf{s})\right] + \frac{1}{4}\sum_{\mathbf{s}} I^2_{\rm obs}, \end{aligned} \quad (7)$$

where

$$\begin{aligned} G_{nm}(\mathbf{s}) = G_{mn}(\mathbf{s}) &= \frac{1}{2}\left[\tilde{\mathbf{F}}_n(\mathbf{s})\tilde{\mathbf{F}}^*_m(\mathbf{s}) + \tilde{\mathbf{F}}_m(\mathbf{s})\tilde{\mathbf{F}}^*_n(\mathbf{s})\right] \\ &= \tilde{F}_n(\mathbf{s})\tilde{F}_m(\mathbf{s})\cos\left[\varphi_n(\mathbf{s}) - \varphi_m(\mathbf{s})\right]. \end{aligned} \quad (8)$$

The model values to be compared with the observed intensities (6) include not only the intensities from the individual components, $n = m$, but also the cross-terms mixing unscaled structure factors from two components, $n \neq m$. Being a half-sum of two complex conjugates (8), coefficients $G_{nm}(\mathbf{s})$ describing these cross-terms are real numbers.

The polynomial of the fourth degree (7) with respect to individual scale factors k_n can be minimized using a standard approach, *e.g.* L-BFGS (Liu & Nocedal, 1989). Similar to other gradient-based algorithms for a local minimization, it is an iterative procedure which requires the initial values for refinable variables to be reasonably close to the expected solution, as well as all partial derivatives with respect to these variables. Depending on the number of refinable variables and the proximity of their initial values to the expected solution, several (typically between ten and 100) iterations of minimization are typically required. The derivatives of LS_I with respect to k_j, $j = 0, \ldots N$, required by the minimizer, are

$$\begin{aligned} \frac{\partial LS_I}{\partial k_j} &= \sum_{m=0}^{N}\sum_{n=0}^{N}\sum_{l=0}^{N} k_lk_nk_m\left[\sum_{\mathbf{s}} G_{jl}(\mathbf{s})G_{nm}(\mathbf{s})\right] \\ &\quad - \sum_{n=0}^{N} k_n\left[\sum_{\mathbf{s}} G_{jn}(\mathbf{s})I_{\rm obs}(\mathbf{s})\right]. \end{aligned} \quad (9)$$

2.3.3. Algorithm 3: non-iterative two-step search. In this algorithm, instead of using iterative minimization methods, we search for the minimum of (6) analytically, which does not require an estimate of initial values for k_n. First, we introduce $(N+1)^2$ intermediate parameters,

$$\xi_{mn} = k_mk_n = \xi_{nm} \text{ where } m, n = 0, \ldots, N. \quad (10)$$

We start from the search for their values that we decompose later into individual coefficients k_n.

Rewriting the function (6) using new variables (10) makes it a quadratic function of these new variables,

$$LS_I = \frac{1}{4}\sum_{\mathbf{s}}\left\{\left[\sum_{m,n=0}^{N} \xi_{nm}G_{nm}(\mathbf{s})\right] - I_{\rm obs}(\mathbf{s})\right\}^2, \quad (11)$$

which we minimize with respect to ξ_{nm}. The minimum of LS_I can be found as a solution of a system of linear equations with respect to these unknowns. After excluding the redundant variables due to the commutativity property, $\xi_{mn} = \xi_{nm}$, we stay with $\frac{1}{2}(N+1)(N+2)$ equations $[\partial/(\partial\xi_{jl})]LS_I = 0$ for the independent variables ξ_{jl}, $0 \leq j \leq l \leq N$:

$$\sum_{0\leq m\leq n}^{N} \xi_{nm}\varepsilon_{nm}\sum_{\mathbf{s}} G_{jl}(\mathbf{s})G_{nm}(\mathbf{s}) = \sum_{\mathbf{s}} G_{jl}(\mathbf{s})I_{\rm obs}(\mathbf{s}). \quad (12)$$

Here $\varepsilon_{nm} = 1$ if $m = n$ and $\varepsilon_{nm} = 2$ otherwise, as this comes after swapping the order of summation in derivatives of (11) and putting together the terms with the indices mn and nm. This is a system of linear equations that can be solved using a standard approach (for example, Meckes & Meckes, 2018).

Solution of (12) yields ξ_{nm} values (10), $0 \leq m \leq n \leq N$, which now allows one to search for $N+1$ scale coefficients k_n by minimizing the following residual:

$$\begin{aligned} LS_\xi &= \frac{1}{2}\sum_{m,n=0}^{N}\left[\ln(k_nk_m) - \ln\xi_{nm}\right]^2 \\ &= \frac{1}{2}\sum_{m,n=0}^{N}\left[\ln k_n + \ln k_m - \ln\xi_{nm}\right]^2. \end{aligned} \quad (13)$$

Using logarithms rather than the values themselves in (13) allows us to find the minimum of (13) with respect to k_n analytically as a solution of the system of linear equations

$$(N+1)\ln k_j + \sum_{n=0}^{N}\ln k_n = \sum_{n=0}^{N}\ln\xi_{jn}. \quad (14)$$

This gives

$$\ln k_j = \frac{1}{(N+1)}\left[\sum_{n=0}^{N}\ln\xi_{jn} - \frac{1}{2(N+1)}\sum_{m=0}^{N}\sum_{n=0}^{N}\ln\xi_{mn}\right],$$
$$j = 0, \ldots, N, \tag{15}$$

where recovering k_n from $\ln k_n$ is trivial.

While this algorithm requires neither iterations nor initial values of the scale factors, its serious disadvantage is the large dimension of the system of equations (12), the need to use a square matrix of the dimension $\frac{1}{2}(N+1)(N+2)$, and sensitivity to rounding errors. This makes it impractical when applied to real structures and we describe it here for the sake of completeness.

2.3.4. Algorithm 4: iterative phased search. With this algorithm, we try to avoid both an iterative minimization of a function of many variables (algorithm 2) and the use of a large system of equations (algorithm 3). To do so, instead of comparison of intensities, we compare structure factors as complex values. The generally unknown phase values $\varphi_{\rm obs}(\mathbf{s})$ can be approximated as those of the model structure factors (4),

$$\varphi_{\rm obs}(\mathbf{s}) \simeq \varphi_{\rm model}(\mathbf{s}), \tag{16}$$

which is a reasonable assumption for a nearly finalized model, the scenario when the multi-component model is expected to be used. We express the best fit of the complex structure factors as a function to be minimized with respect to k_n,

$$\begin{aligned} LF_F &= \frac{1}{2}\sum_{\mathbf{s}}\left[\sum_{n=0}^{N}k_n\tilde{\mathbf{F}}_n(\mathbf{s}) - \mathbf{F}_{\rm obs}(\mathbf{s})\right]^*\left[\sum_{n=0}^{N}k_n\tilde{\mathbf{F}}_n(\mathbf{s}) - \mathbf{F}_{\rm obs}(\mathbf{s})\right] \\ &= \frac{1}{2}\sum_{n,m=0}^{N}k_nk_m\left[\sum_{\mathbf{s}}G_{nm}(\mathbf{s})\right] - \sum_{n=0}^{N}k_n\left[\sum_{\mathbf{s}}H_n(\mathbf{s})\right] \\ &\quad + \frac{1}{2}\sum_{\mathbf{s}}\left[F_{\rm obs}(\mathbf{s})\right]^2, \end{aligned} \tag{17}$$

where $G_{nm}(\mathbf{s})$ are defined previously by (8) and $H_n(\mathbf{s})$ are defined similarly as

$$\begin{aligned} H_n(\mathbf{s}) &= \frac{1}{2}\left[\tilde{\mathbf{F}}_n^*(\mathbf{s})\mathbf{F}_{\rm obs}(\mathbf{s}) + \tilde{\mathbf{F}}_n(\mathbf{s})\mathbf{F}_{\rm obs}^*(\mathbf{s})\right] \\ &= \Re\left[\tilde{\mathbf{F}}_n^*(\mathbf{s})\mathbf{F}_{\rm obs}(\mathbf{s})\right] \\ &= \tilde{F}_n(\mathbf{s})F_{\rm obs}(\mathbf{s})\cos\left[\varphi_n(\mathbf{s}) - \varphi_{\rm obs}(\mathbf{s})\right]. \end{aligned} \tag{18}$$

Minimization of (17) results in a system of $N+1$ linear equations with respect to k_n:

$$\sum_{n=0}^{N}k_n\left[\sum_{\mathbf{s}}G_{jn}(\mathbf{s})\right] = \sum_{\mathbf{s}}H_j(\mathbf{s}), \quad j = 0, \ldots, N, \tag{19}$$

which, similarly to (12), can be solved using a standard approach. Several iterations, typically up to a few dozens, may be required to solve (17), with each iteration improving model structure factors (4) and respective phase values (16) and updating $H_n(\mathbf{s})$ (18).

3. Testing algorithms 2 and 4

3.1. Generalities

As discussed in the Introduction, the multi-component model may be applied to the solution of various problems and, generally, it consists of two stages: (i) defining these components and calculation of structure factors from them, and (ii) combining these structure factors together into the total model structure factor (4). The first stage (defining components) is very problem specific. The components may arise as a result of annotation of macromolecular cavities (Matthews & Liu, 2009), or map analysis to find and model regions of semi-ordered lipid layers (Sonntag *et al.*, 2011), or from calculating blurred binary masks to account for the bulk solvent (Jiang & Brünger, 1994), or use a large-Gaussian model to approximate yet unmodelled parts of the macromolecule (Lunin *et al.*, 1995), and so on. The second stage (combining structure factors from multiple components) is not problem specific: it is independent of how the components and their structure factors were obtained. Since in this work we describe the algorithms that address the second stage, the test calculations described below have been done using a simple self-contained model to prove that the algorithms can find accurate values of the multi-component optimal scale functions $k_n(\mathbf{s})$ in (4). In what follows, we focus on algorithms 2 (iterative one-step search) and 4 (iterative phased search) as algorithms 1 (sequential search) and 3 (non-iterative two-step search) are much less likely to find practical application. Also, as stated in Section 2.2, during the search for the scale factors k_n all structure factors, $\mathbf{F}_{\rm calc}(\mathbf{s})$ and $\mathbf{F}_n(\mathbf{s})$, remain unchanged.

3.2. Error-free test with a few components representing isolated regions inside a protein

To test the performance of these algorithms, the following numeric experiment was set up. The Ypd1p model [PDB (Protein Data Bank) code 1c03, Song *et al.*, 1999] was obtained from the PDB (Burley *et al.*, 2021) and the bulk solvent mask was calculated using the standard approach (Jiang & Brünger, 1994). This mask has one large isolated region that constitutes about 57% of the unit-cell volume (174 794 Å^3) and six much smaller regions with the volume varying between 50 and 190 Å^3. Each of these regions was considered as an individual solvent region with its own binary mask, 1 inside the region and 0 outside. The total model structure factor for this system was defined according to (3) as

$$\mathbf{F}_{\rm model}(\mathbf{s}) = \mathbf{F}_{\rm calc\_atoms}(\mathbf{s}) + \exp\left(-\frac{Bs^2}{4}\right)\sum_{n=1}^{N}k_n\mathbf{F}_{\rm mask\_}n(\mathbf{s}). \tag{20}$$

Here, $N = 7$ is the number of regions, and the exponential resolution-dependent scale factor was introduced similarly to the flat bulk solvent model to smooth the sharp boundaries of masks with the smearing B factor of 50 Å^2 (Fokine & Urzhumtsev, 2002). Each region was assumed to have its own individual scale factor k_n, and their values were assigned randomly in the range between 0 and 1. For each trial choice

of k_n the corresponding set of structure factors (20) was calculated and their absolute values were then referred to as error-free 'observable data' $F_{obs}(\mathbf{s})$. These $F_{obs}(\mathbf{s})$, $\mathbf{F}_{calc\_atoms}(\mathbf{s})$ and the set of smeared $\mathbf{F}_{mask\_n}(\mathbf{s})$ were subjected to algorithms 2 and 4 and the obtained values of k_n were compared with the known values using relative error as a measure. Additionally, the crystallographic R factor was calculated using the known exact $F_{obs}(\mathbf{s})$ and model structure factors (2) calculated with k_n values recovered by one of the two algorithms. Since the outcome of the procedure can potentially depend on the choice of k_n used to calculate $F_{obs}(\mathbf{s})$ and the initial k_n values used by algorithms, the procedure was repeated 1000 times, each time using the different set of k_n and varying the initial values for k_n within about an order of magnitude from the known values. In all cases, both algorithms recovered the k_n values almost exactly, within 0.0001% error, regardless of the choice of k_n and the initial values.

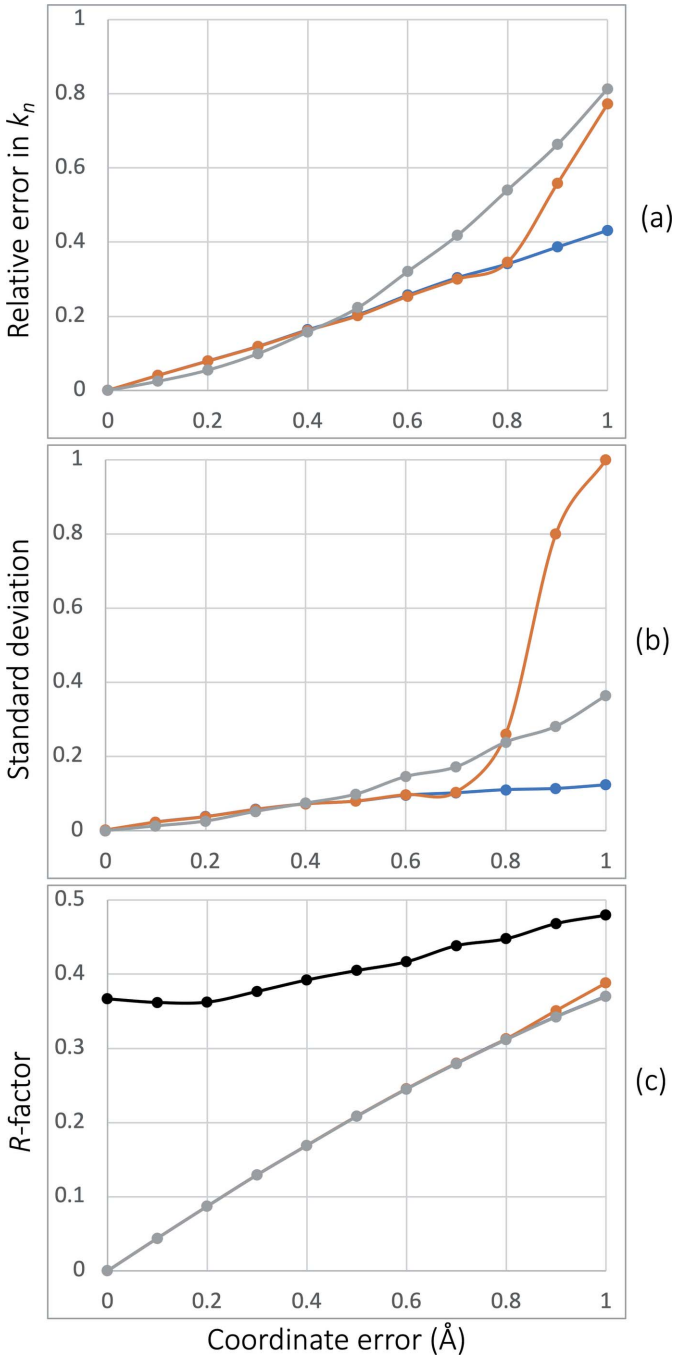


Figure 1
Relative mean error in k_n (*a*), its standard deviation (*b*) and (*c*) R factor between error-free simulated $F_{obs}(\mathbf{s})$ and $|\mathbf{F}_{model}(\mathbf{s})|$ (20) computed from an atomic model with indicated mean coordinate errors using k_n values recovered by algorithm 4 (gray), algorithm 2 without second derivatives (orange) and algorithm 2 using second derivatives (blue). The black line in (*c*) shows the initial R factor calculated assuming all k_n values are zero.

3.3. Robustness with respect to errors in the atomic model

Additionally, the performance of the algorithms was assessed in the presence of random errors in atomic model coordinates using the same test setup as in Section 3.2.

Generally, the errors can be of several types (*e.g.* systematic, random) and have many sources, such as errors in atomic model parameters (coordinates, B factors, occupancies) or model incompleteness, as well as errors in experimental data (measurement errors, completeness). Here we only focus on removable model errors (Lunin *et al.*, 2002), which do not prevent the model eventually reproducing the experimental data accurately if all model parameters have their exact values. This is fundamentally different to the case of irremovable errors. An example of irremovable errors is crystal structure model incompleteness, when the model describes only a part of the entire unit-cell content. In this case no choice of model parameters can fully compensate for the missing scattering and the best fit of model parameters to the data does not necessarily lead to accurate model parameters, in fact, the opposite (Lunin *et al.*, 2002). This problem is typically addressed by the appropriate choice of refinement target function and not by the optimization procedure itself (Lunin *et al.*, 2002).

Provided the model completely describes the unit-cell content, errors in atomic coordinates are an example of removable errors that we consider in what follows. Also, simulation of random errors in atomic coordinates can be thought of as somewhat similar to the simulation of correlated random errors in the experimental data (Lunin *et al.*, 2002; Holton *et al.*, 2014). Thus, in the following test random errors of different magnitude were introduced to atomic coordinates leading to the root-mean-squared deviation (RMSD) between initial unperturbed and perturbed models in the range between 0 and 1 Å with a step of 0.1 Å. The unperturbed atomic model, the set of mask structure factors calculated for each of seven regions and the known values of k_n were used to generate $I_{obs}(\mathbf{s})$ using formula (20). The perturbed model was used to calculate $\mathbf{F}_{calc}(\mathbf{s})$ during the search. For each perturbation dose, 1000 trials of running algorithms 2 and 4 were performed as described above for the error-free case, and the mean of the relative error in k_n and the standard deviation were calculated across all 1000 trials [Figs. 1(*a*), 1(*b*)]. Additionally, the crystallographic R factor was calculated [Fig. 1(*c*)]. Both algorithms perform similarly up to the coordinate error of 0.4 Å, leading to the relative error under 20%; this

coordinate error is within various estimates reported in the literature [see, for example, pp. 658–662 in Rupp (2009), and references therein]. After that limit, algorithm 2 performs systematically better. For large coordinate errors, using second derivatives of (6) explicitly calculated and supplied to L-BFGS

$$\frac{\partial^2 LS_I}{\partial k_l \partial k_j} = \sum_{m=0}^{N}\sum_{n=0}^{N} k_n k_m \left[\sum_{\mathbf{s}} G_{jl}(\mathbf{s}) G_{nm}(\mathbf{s})\right] + 2\sum_{n=0}^{N} k_n k_m \left[\sum_{\mathbf{s}} G_{jm}(\mathbf{s}) G_{lm}(\mathbf{s})\right] - \sum_{\mathbf{s}} G_{jl}(\mathbf{s}) I_{\rm obs}(\mathbf{s}) \quad (21)$$

improved the performance of algorithm 2 further. Overall, algorithm 2 with second derivatives seems to perform best across all trials in terms of yielding the lowest relative error [Fig. 1(*a*)] and more consistently [Fig. 1(*b*)] compared with other algorithms. However, given that errors of magnitude 0.5 Å or larger are rather rare and unrealistic, and algorithm 2 is much slower than algorithm 4, the latter may be the default option of choice for practical applications.

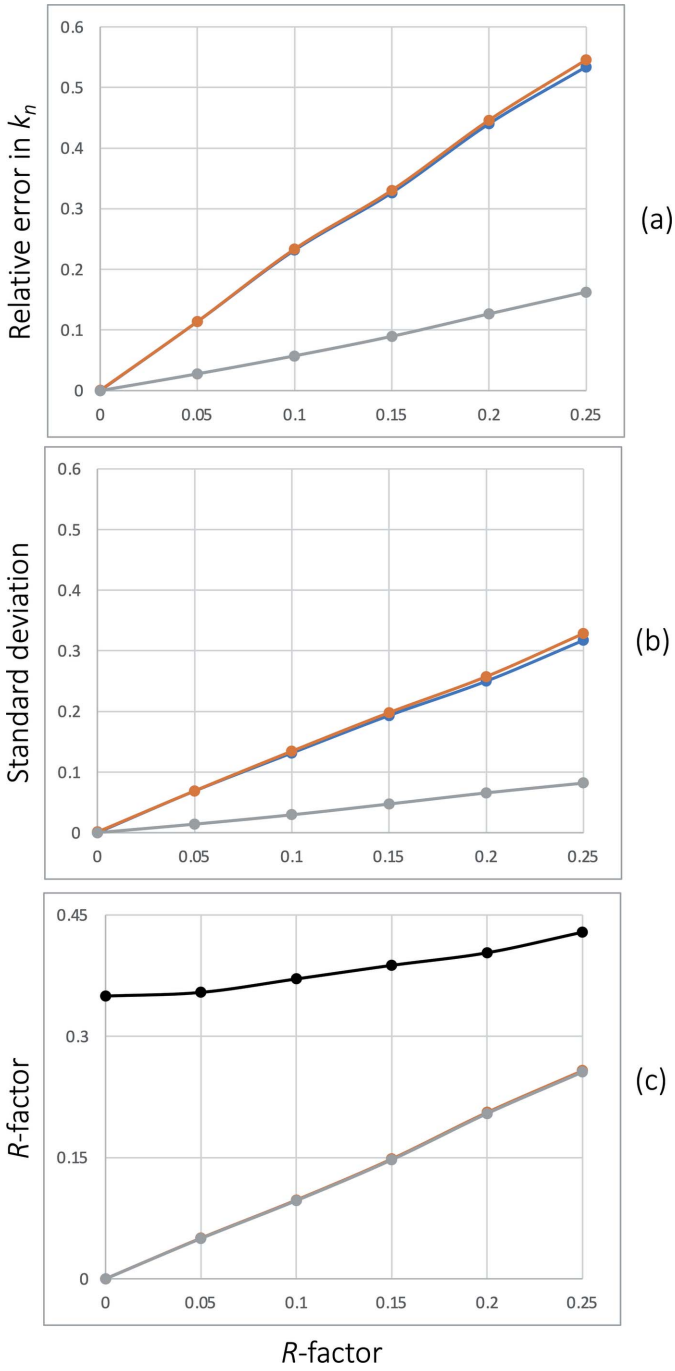


Figure 2
Relative mean error in k_n (*a*), its standard deviation (*b*) and (*c*) *R* factor between error-free simulated $F_{\rm obs}(\mathbf{s})$ and $|\mathbf{F}_{\rm model}(\mathbf{s})|$ (20) computed using k_n values recovered by algorithm 4 (gray), algorithm 2 without second derivatives (orange) and algorithm 2 using second derivatives (blue), plotted as a function of a random Gaussian error introduced to observed $I_{\rm obs}$ which is expressed through the respective *R* factor. The black line in (*c*) shows the initial *R* factor calculated assuming all k_n values are zero.

3.4. Robustness with respect to the number *N* of components

In the tests above, the rather small number of components contributing to the total model structure factor (3–4) were defined by the atomic model of choice and remained the same in all calculations. However, the number and size (especially relative to the macromolecule and to each other) of these components can potentially affect the performance of the algorithms. To explore this, the following numeric experiment was set up. The lysozyme model (PDB code 1jkb, Muraki *et al.*, 1997) was obtained from the PDB (Burley *et al.*, 2021) and placed in the middle of a virtual *P*1 unit-cell box. The atomic model occupied 25% of the unit cell, which corresponds to a somewhat above average solvent content. Individual regions that contribute to the total model structure factor were mimicked by spheres placed in the solvent region of the unit cell such that they occupied the entire solvent region and did not overlap with the protein and themselves. The size (radius R_n) and occupancy k_n of each sphere were chosen randomly between 3 and 10 Å and 0.1 and 100, correspondingly. This typically generated between 30 and 50 spheres. Using spheres as individual mask components allowed a fast calculation of their structure factors analytically using the same *B* factor equal to 50 Å^2 as in the previous tests:

$$\mathbf{F}_{\rm model}(\mathbf{s}) = \mathbf{F}_{\rm calc\_atoms}(\mathbf{s}) + \exp\left(-\frac{Bs^2}{4}\right) \times \sum_{n=1}^{N} k_n \left[\frac{\sin(2\pi s R_n) - (2\pi s R_n)\cos(2\pi s R_n)}{2\pi^2 s^3}\right] \times \exp(i2\pi \mathbf{s}\mathbf{r}_n), \quad (22)$$

where R_n is the sphere radius and $\mathbf{r}_n$ is its center (Appendix *A*). The rest of the test was performed exactly as in the previous example and yielded essentially similar results (not shown).

The Python code of the numeric test described above is part of the *CCTBX* distribution and is used as a regression test for algorithms 2 and 4; it is located in the mmtbx.bulk_solvent module of *CCTBX*.

3.5. Errors in experimental data

So far, all tests described above have been done using the model-simulated error-free experimental data. While modeling the experimental data which include many various sources of errors, *e.g.* those discussed by Borek *et al.* (2003) and Pozharski (2012), is a challenging task, here we focused on the simplest and most straightforward case of independent random errors distributed using the Gaussian law. These errors were introduced into model-calculated values of I_{obs} such that the resulting values of I_{obs} with errors match the exact error-free values up to specified R factors of 0 (no errors), 5, 10, 15, 20 and 25%. This mimics the typical R-factor values in macromolecular crystallography performed at a broad range of resolutions of the experimental data: from ultra-high to mid-low (*e.g.* Urzhumtsev *et al.*, 2009). Similarly to Section 3.3, 1000 runs were done for each of six error doses introduced to I_{obs}. In terms of robustness and consistency of recovering k_n, algorithm 4 performed notably better than either of the two versions of algorithm 2 [Figs. 2(*a*), 2(*b*)]. This is likely because algorithm 4 uses model phases and in this test model phases were kept error free.

3.6. Test with real (not simulated) experimental data

For this test we have selected a model and experimental data from the PDB (PDB code: 4gu0, Chen *et al.*, 2013) and focused on an isolated region inside the protein near residue 131 in chain H [Fig. 3(*a*)]. The residual density map still shows a rather strong peak in this region [Fig. 3(*b*)] after solvent and all scales have been accounted for using the standard approach as implemented in *CCTBX* (Afonine *et al.*, 2013), which suggests that the region is occupied by either a disordered ligand or by a solvent other than the bulk solvent everywhere else. This region is considered as an independent component in (4) and its scale factor k_n was obtained using both algorithms 2 and 4. The inclusion of this region in the total model structure factor (4) with refined k_n (both algorithms yielded virtually the same value) flattened out the residual density map [Fig. 3(*c*)].

4. Discussion

The multi-component approach to modeling the crystal content provides an opportunity for a more complete and accurate description. The model described here allows for explicit inclusion of semi-ordered solvent, disordered ligands and parts of the macromolecule as well as the features in the bulk solvent that deviate from the flat solvent model. In this approach each feature being modeled, which is not a part of the atomic model nor bulk solvent, is treated individually and its contribution to the total model structure factor is added as a correction term with a refinable resolution-dependent scale factor. Calculating these scale factors in a numerically efficient and stable manner is an algorithmic challenge to which we provide a solution. Algorithm 1 is the most straightforward in terms of implementation but at the same time it is the most runtime expensive and offers no guarantee of convergence to the correct result. Algorithm 3 does not require iterations and leads to the solution analytically; however, it is sensitive to rounding errors and is very computer memory expensive. While we found that both algorithm 2 (using second derivatives) and algorithm 4 perform almost identically in terms of recovering parameters in our tests with reasonable-size errors, algorithm 2 requires substantially more calculations and thus it is more runtime expensive. Therefore, algorithm 4 is suggested as the default choice. All the algorithms described here are implemented in *CCTBX* (mmtbx.bulk_solvent module) and are available in the *Phenix* suite starting from version 1.20rc4-4425. Putting these algorithms in production to automatically model non-uniform features of the bulk solvent and disordered parts of the atomic model, both macromolecule and ligands, is an ongoing effort within the *Phenix* team and collaborators.

APPENDIX *A*

Let us define a binary mask 1/0 of a sphere at the origin and with radius R. Its Fourier transform (scattering function, or structure factors if we consider $\mathbf{s}$ as points of the reciprocal-space grid)

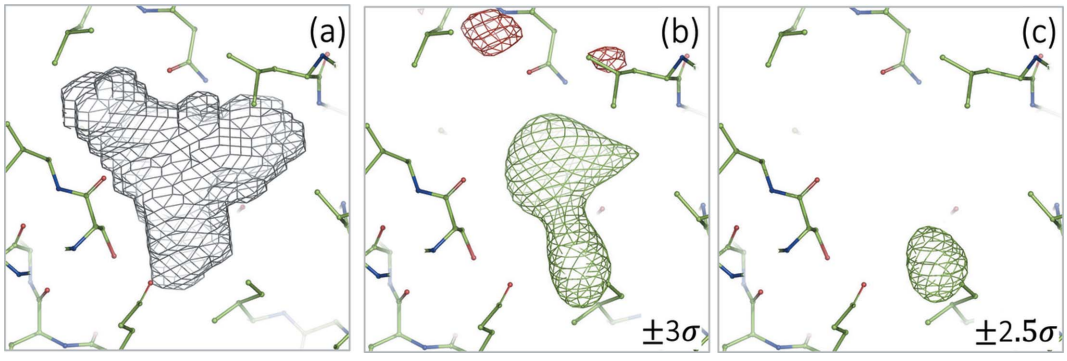


Figure 3
Bulk solvent mask (*a*) outlining a pocket inside the protein (PDB: 4gu0, near residue 131 in chain H) and the weighted difference map (Read, 1986) calculated assuming this pocket is empty (*b*) or filled with a solvent that was modeled using algorithm 4 (*c*). Map contouring levels are indicated on the figure.

$$\mathbf{F}(\mathbf{s}) = \int\limits_{|\mathbf{r}|\leq R} \exp(2\pi i \mathbf{rs})\,\mathrm{d}V \tag{23}$$

is spherically symmetric. Being expressed in spherical coordinates with $r = |\mathbf{r}|$, $s = |\mathbf{s}|$ and θ the angle between the vectors $\mathbf{r}$ and $\mathbf{s}$, its radial component

$$\bar{F}(s) = \int\limits_0^R \int\limits_0^\pi \int\limits_0^{2\pi} r^2 \sin\theta \exp(2\pi i r s \cos\theta)\,\mathrm{d}\varphi\,\mathrm{d}\theta\,\mathrm{d}r \tag{24}$$

becomes equal to the 3D interference function times the volume of the integration sphere:

$$\begin{aligned}\bar{F}(s) &= 2s^{-1} \int\limits_0^R r \sin(2\pi r s)\,\mathrm{d}r \\ &= \frac{\sin(2\pi s R) - (2\pi s R)\cos(2\pi s R)}{2\pi^2 s^3} \\ &= \frac{4\pi R^3}{3}\left[3\frac{\sin(2\pi s R) - (2\pi s R)\cos(2\pi s R)}{(2\pi s R)^3}\right].\end{aligned} \tag{25}$$

Funding information

PVA and PDA thank the NIH (grants R01GM071939, P01GM063210 and R24GM141254) and the PHENIX Industrial Consortium for support of the PHENIX project. This work was supported in part by the US Department of Energy under contract No. DE-AC02-05CH11231. AGU acknowledges Instruct-ERIC and the French Infrastructure for Integrated Structural Biology FRISBI (ANR-10-INBS-05).

References

Afonine, P. V., Grosse-Kunstleve, R. W., Adams, P. D. & Urzhumtsev, A. (2013). *Acta Cryst.* D**69**, 625–634.

Blanc, E., Roversi, P., Vonrhein, C., Flensburg, C., Lea, S. M. & Bricogne, G. (2004). *Acta Cryst.* D**60**, 2210–2221.

Borek, D., Minor, W. & Otwinowski, Z. (2003). *Acta Cryst.* D**59**, 2031–2038.

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., Bhikadiya, C., Bi, C., Bittrich, S., Chen, L., Crichlow, G. V., Christie, C. H., Dalenberg, K., Di Costanzo, L., Duarte, J. M., Dutta, S., Feng, Z., Ganesan, S., Goodsell, D. S., Ghosh, S., Green, R. K., Guranović, V., Guzenko, D., Hudson, B. P., Lawson, C., Liang, Y., Lowe, R., Namkoong, H., Peisach, E., Persikova, I., Randle, C., Rose, A., Rose, Y., Sali, A., Segura, J., Sekharan, M., Shao, C., Tao, Y., Voigt, M., Westbrook, J., Young, J. Y., Zardecki, C. & Zhuravleva, M. (2021). *Nucleic Acids Res.* **49**, D437–D451.

Chen, F., Yang, H., Dong, Z., Fang, J., Wang, P., Zhu, T., Gong, W., Fang, R., Shi, Y. G., Li, Z. & Xu, Y. (2013). *Cell Res.* **23**, 306–309.

Fokine, A. & Urzhumtsev, A. (2002). *Acta Cryst.* D**58**, 1387–1392.

Holton, J. M., Classen, S., Frankel, K. A. & Tainer, J. A. (2014). *FEBS J.* **281**, 4046–4060.

Jiang, J. S. & Brünger, A. T. (1994). *J. Mol. Biol.* **243**, 100–115.

Liebschner, D., Afonine, P. V., Baker, M. L., Bunkóczi, G., Chen, V. B., Croll, T. I., Hintze, B., Hung, L.-W., Jain, S., McCoy, A. J., Moriarty, N. W., Oeffner, R. D., Poon, B. K., Prisant, M. G., Read, R. J., Richardson, J. S., Richardson, D. C., Sammito, M. D., Sobolev, O. V., Stockwell, D. H., Terwilliger, T. C., Urzhumtsev, A. G., Videau, L. L., Williams, C. J. & Adams, P. D. (2019). *Acta Cryst.* D**75**, 861–877.

Liu, D. C. & Nocedal, J. (1989). *Math. Program.* **45**, 503–528.

Lunin, V. Y., Afonine, P. V. & Urzhumtsev, A. G. (2002). *Acta Cryst.* A**58**, 270–282.

Lunin, V. Yu., Lunina, N. L., Petrova, T. E., Vernoslova, E. A., Urzhumtsev, A. G. & Podjarny, A. D. (1995). *Acta Cryst.* D**51**, 896–903.

Matthews, B. W. & Liu, L. (2009). *Protein Sci.* **18**, 494–502.

Meckes, E. S. & Meckes, M. W. (2018). *Linear Algebra*. Cambridge University Press.

Moews, P. C. & Kretsinger, R. H. (1975). *J. Mol. Biol.* **91**, 201–225.

Muraki, M., Goda, S., Nagahora, H. & Harata, K. (1997). *Protein Sci.* **6**, 473–476.

Murshudov, G. N., Skubák, P., Lebedev, A. A., Pannu, N. S., Steiner, R. A., Nicholls, R. A., Winn, M. D., Long, F. & Vagin, A. A. (2011). *Acta Cryst.* D**67**, 355–367.

Pozharski, E. (2012). *Acta Cryst.* D**68**, 1077–1087.

Read, R. J. (1986). *Acta Cryst.* A**42**, 140–149.

Roversi, P., Blanc, E., Vonrhein, C., Evans, G. & Bricogne, G. (2000). *Acta Cryst.* D**56**, 1316–1323.

Rupp, B. (2009). *Biomolecular Crystallography: Principles, Practice and Applications to Structural Biology*. New York: Garland Science, Taylor and Francis Group.

Sheldrick, G. M. (2008). *Acta Cryst.* A**64**, 112–122.

Sheriff, S. & Hendrickson, W. A. (1987). *Acta Cryst.* A**43**, 118–121.

Song, H. K., Lee, J. Y., Lee, M. G., Moon, J., Min, K., Yang, J. K. & Suh, S. W. (1999). *J. Mol. Biol.* **293**, 753–761.

Sonntag, Y., Musgaard, M., Olesen, C., Schiøtt, B., Møller, J. V., Nissen, P. & Thøgersen, L. (2011). *Nat. Commun.* **2**, 304.

Spek, A. L. (2015). *Acta Cryst.* C**71**, 9–18.

Tronrud, D. E. (1997). *Methods Enzymol.* **277**, 306–319.

Urzhumtsev, A., Afonine, P. V. & Adams, P. D. (2009). *Acta Cryst.* D**65**, 1283–1291.

Urzhumtsev, A. & Podjarny, A. D. (1995). *Jnt CCP4/ESF–EACMB Newsl. Protein Crystallogr.* **31**, 12–16.