Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Multidimensional molecular replacement

Nicholas M. Glykos[a]* and Michael Kokkinidis[a,b]

[a]IMBB, FORTH, PO Box 1527, 71110 Heraklion, Crete, Greece, and [b]Department of Biology, University of Crete, PO Box 2208, 71409 Heraklion, Crete, Greece

Correspondence e-mail: glykos@crystal2.imbb.forth.gr

Received 10 February 2001
Accepted 24 May 2001

A method is described which attempts to simultaneously and independently determine the positional and orientational parameters of all molecules present in the asymmetric unit of a target crystal structure. This is achieved through a reverse Monte Carlo optimization of a suitable statistic (such as the R factor or the linear correlation coefficient between the observed and calculated amplitudes of the structure factors) in the $6n$-dimensional space defined by the rotational and translational parameters of the n search models. Results from the application of this stochastic method – obtained with a space-group-general computer program which has been developed for this purpose – indicate that with present-day computing capabilities the method may be applied successfully to molecular-replacement problems for which the target crystal structure contains up to three molecules per asymmetric unit. It is also shown that the method may be useful in cases where the assumption of topological segregation of the self- and cross-vectors in the Patterson function is violated (as may happen, for example, in closely packed crystal structures).

1. Introduction

The classical approach to the problem of placing n copies of a search model in the asymmetric unit of a target crystal structure is to divide this problem into a succession of three-dimensional searches (rotation-function followed by translation-function searches for each of the models), as described by Rossmann & Blow (1962), Rossmann (1972, 1990), Machin (1985), Dodson *et al.* (1992) and Carter & Sweet (1997). A more recently developed class of algorithms attempts to improve the sensitivity and accuracy of the molecular-replacement method by increasing the dimensionality of the parameter space explored simultaneously. This is achieved by performing successive six-dimensional searches for each of the molecules present in the crystallographic asymmetric unit. Published examples of such methods include a genetic algorithm approach (Chang & Lewis, 1997), an evolutionary search methodology (Kissinger *et al.*, 1999) and a systematic six-dimensional search using a fast translation function (Sheriff *et al.*, 1999).

We have recently described (Glykos & Kokkinidis, 2000*a*) an alternative $6n$-dimensional molecular-replacement procedure which is based on the simultaneous determination of the rotational and translational parameters of all molecules present in the crystallographic asymmetric unit of a target structure. In this communication, we present an overview of the current state of the method, its practical implementation in the form of a space-group-general computer program and the

application of this program to molecular-replacement problems of varying complexity.

2. Methods and algorithms: an overview

2.1. Stating the problem

If there are n copies of a search model in the asymmetric unit of the target crystal structure then in general there are $6n$ parameters whose values are to be determined by molecular replacement (three rotational and three translational parameters for each of the search models). These $6n$ parameters in turn define a $6n$-dimensional configurational space in which each and every point corresponds to a possible configuration for the target crystal structure; therefore, for each and every of these points it is possible to calculate the value of a suitable statistic (such as the R factor or the linear correlation coefficient) measuring the agreement between the experimentally observed and the calculated structure-factor amplitudes. By assuming that the correct solution corresponds to the global optimum of this statistic, the molecular-replacement problem is reduced to one of the unconstrained global optimization of the chosen statistic in the $6n$-dimensional space defined by the rotational and translational parameters of the molecules. Stated in simpler terms, the aim of the proposed method is to find which combination of positions and orientations of the n molecules optimizes the value of the R factor or correlation coefficient between the observed and calculated data. In this respect (and by performing the search in a continuous parameter space), the method views molecular replacement as a generalized rigid-body refinement problem.

2.2. Method of solution

The volume of the configurational space defined by the rotational and translational parameters of the molecules present in the asymmetric unit of a target crystal structure is so large that a systematic examination of all possible combinations of their positions and orientations is beyond present-day computing capabilities (however, see Sheriff *et al.*, 1999 for an example of a systematic six-dimensional search). On the other hand, stochastic methods (such as simulated annealing or genetic algorithms) have repeatedly been shown to be able to deal with multidimensional combinatorial optimization problems in near-optimal ways and in a fraction of the time required for a systematic search (Kirkpatrick *et al.*, 1983; Press *et al.*, 1992).

We have chosen to use a modification of the reverse Monte Carlo technique (McGreevy & Pusztai, 1988; Keen & McGreevy, 1990), where instead of minimizing the quantity $\chi^2 = \sum_{hkl}[(F_o - F_c)/\sigma(F_o)]^2$, one minimizes any of the following (user-defined) target functions: (i) the conventional crystallographic R factor, $R = \sum_{hkl} |F_o - F_c| / \sum_{hkl} F_o$, (ii) the quantity $1.0 - \mathrm{Corr}(F_o, F_c)$ and (iii) the quantity $1.0 - \mathrm{Corr}(F_o^2, F_c^2)$, where Corr() is the linear correlation coefficient function, F_o and F_c are the observed and calculated structure-factor amplitudes of the hkl reflection and $\sigma(F_o)$ is the standard uncertainty of the corresponding measurement. To avoid unnecessary repetition and to simplify the discussion that follows, we will hereafter refer only to the R-factor statistic, on the understanding that any of the correlation-based targets can be substituted for it.

The minimization procedure follows closely the original Metropolis algorithm (Metropolis *et al.*, 1953) and its basic steps are outlined below. Random initial orientations and positions are assigned to all molecules present in the crystallographic asymmetric unit of the target structure and the R factor ($= R_{\rm old}$) between the observed and calculated structure-factor amplitudes is noted. In the first step of the basic iteration, a molecule is chosen randomly and its orientational and translational parameters are randomly altered. The R factor ($= R_{\rm new}$) corresponding to this new arrangement is calculated and compared with $R_{\rm old}$: if $R_{\rm new} \leq R_{\rm old}$, then the new configuration is accepted and the procedure is iterated with a new (randomly chosen) molecule. If $R_{\rm new} > R_{\rm old}$ (that is, if the new configuration results in a worse R factor), the new configuration is accepted with probability $\exp[(R_{\rm old} - R_{\rm new})/T]$, where T is a control parameter which plays a role analogous to that of temperature in statistical mechanical simulations. This probabilistic treatment again relies on the random-number generator: if $\exp[(R_{\rm old} - R_{\rm new})/T] > \xi$, where ξ is a random number between 0.0 and 1.0, the new configuration is accepted and the procedure iterated. If $\exp[(R_{\rm old} - R_{\rm new})/T] \leq \xi$, we return to the previous configuration (the one that resulted in a R factor equal to $R_{\rm old}$) and reiterate. Given enough time, this algorithm is guaranteed to find the global optimum of the target function (Ingber, 1993).[1]

By trading computer memory for speed of execution, the CPU time required per iteration of the Monte Carlo algorithm can be made to be only linearly dependent on the number of reflections of the target structure expanded to space group $P1$. This is achieved by calculating (and storing in memory) the molecular transform of the search model before the actual minimization is started. For the rest of the simulation, to calculate a structure-factor amplitude $F_c(hkl)$, we only have to add the (complex) values of the molecular transform at the coordinates that the hkl reflection would take if rotated accordingly to the orientation of each molecule in the unit cell (a detailed account on the usage of the molecular transform to accelerate the structure-factor calculation for this type of problem can be found in §2.1 of Chang & Lewis, 1997). Additionally, and in order to avoid a dependence on the number of molecules present in the asymmetric unit of the target structure, the contribution of each molecule to every reflection is also stored in memory and so at each iteration we only have to recalculate the contribution from the molecule that is being tested.

[1] Strictly speaking, simulated annealing is guaranteed to find the global optimum of the target function only in the case of the so-called Boltzmann annealing, for which the temperature $T(k)$ at each step k of the simulation is given by $T(k) = T_0/\log(k)$, where T_0 is the starting temperature (Ingber, 1993). Only with this annealing schedule and with T_0 'sufficiently high' is the algorithm guaranteed to find the global optimum of the target function. In this respect, the linear slow-cooling protocol discussed in §3.1 of this paper is more accurately described by the term 'simulated quenching' than the conventionally used term 'simulated annealing'.

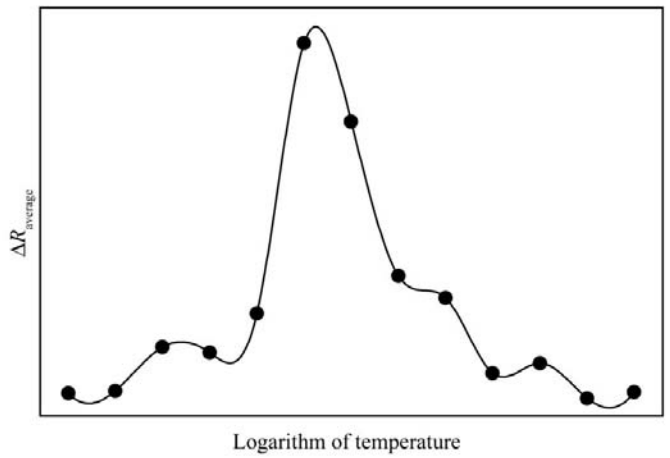

Figure 1
Variation of the average values of the target function as a function of temperature during an automatic temperature-limit determination for a problem with one (pronounced) phase transition. The filled circles correspond to the actual measurements made by the program; the continuous line is the natural cubic spline interpolation of these points. All graphs were prepared using the program *Xmgr*, available from http://plasma-gate.weizmann.ac.il/Xmgr/.

2.3. Methodological limitations

Three salient features of the method proposed in the previous sections are worth discussing in more detail. The first is that all configurations are treated as *a priori* equally probable, without reference to whether their packing arrangement is physically and chemically sensible. Although it is in principle possible to include a van der Waals repulsion term in the method (to take into account bad contacts between symmetry-related molecules), this would destroy the ergodicity property of simulated annealing; that is, it will no longer be possible to guarantee that each and every state of the system can be reached within a finite number of moves. This is a consequence of the fact that once an arrangement is found that allows the efficient packing of the search models and their symmetry equivalents in the target unit cell, no further major rearrangements of the molecular configuration will be possible (especially in tightly packed crystal forms) and the minimization would come to a halt.

A second limitation of the method is that by optimizing a global statistic such as the correlation coefficient or the R factor, it tries to simultaneously match both the self vectors (of the search models) and all of the cross vectors (between search models and their crystallographically equivalent molecules). The problem with this approach is that as the search model is becoming worse and worse, the agreement for the cross vectors (which are on the average longer) deteriorates much faster than for the (shorter) self vectors, thus reducing the effective signal-to-noise ratio for the correct solution. In contrast, the traditional rotation function (possibly because it restricts itself to a self-vector-enriched volume of the Patterson function) is expected to be able to sustain a recognisable solution even for quite inaccurate starting models, increasing in this way the probability that a subsequent translation function will also be successful. The implication of this analysis is that when a sufficiently accurate search model is not available this stochastic method may be less sensitive (compared with the conventional Patterson-based methods) in identifying the correct solution.

The third (and most important) limitation of this method is that by treating the problem as $6n$-dimensional, it ignores all the information offered by the properties of the Patterson function. This includes information about the probable orientations of the molecules (usually presented in the form of the cross-rotation function) and of the relationships between them (usually in the form of the self-rotation function). The method as described above also fails to automatically take into account cases of pure translational non-crystallographic symmetry (Navaza *et al.*, 1998), although it is relatively easy to account for such forms of non-crystallographic symmetry through the incorporation of additional fixed symmetry elements. It is worth mentioning here that if the assumption of topological segregation of the self- and cross-vectors in the Patterson function holds, then molecular-replacement problems are not $6n$-dimensional but rather two $3n$-dimensional problems: the first $3n$-dimensional problem is a generalized cross-rotation function which would attempt to determine the orientation of all n molecules simultaneously (by taking into account not only the agreement between the observed Patterson function and an isolated set of self vectors from just one of the search models, but also the interactions between the n copies of self-vector sets that are necessarily present in the observed Patterson function). The second $3n$-dimensional problem is a generalized translation function which would attempt to simultaneously determine the positions of all n properly oriented (from the first step) search models. For this reason, and as long as the assumptions behind Patterson-based methods hold, $6n$-dimensional searches 'overkill' the molecular-replacement problem by unnecessarily doubling the dimensionality of the search space.

It should be mentioned, however, that this very property of ignoring evidence obtained from the Patterson function makes these methods more robust and suitable for problems where the assumptions behind the Patterson-based methods are not satisfied. One such example will be presented later.

3. Implementation

A space-group-general computer program has been developed which implements the method described in the previous sections (see §6 for information about how to obtain a copy of the program). As is always the case with Monte Carlo algorithms, the efficiency of the minimization depends greatly on the optimal (or otherwise) choice of (i) an annealing schedule which specifies how the temperature of the system will vary with time, (ii) the temperature (or temperature range) that will be used during the simulations, (iii) a set of moves that determine how the next configuration (the one that will be tested) can be obtained from the current configuration (the one that has already been tested) and (iv) a suitable (for the problem under examination) target function whose value is to

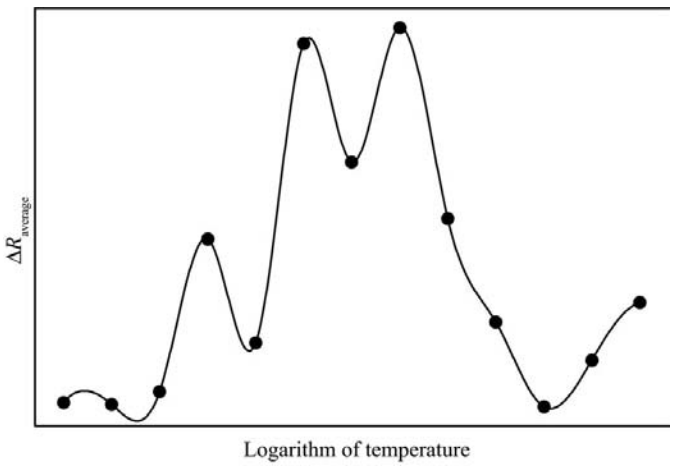

Figure 2
Variation of the average values of the target function as a function of temperature during an automatic temperature-limit determination for a problem with at least two phase transitions. The filled circles correspond to the actual measurements made by the program; the continuous line is the natural cubic spline interpolation for these points.

be optimized. The following sections discuss these four points in more detail and present additional information about two other issues that are important for the specific application, namely scaling of the observed and calculated data, and bulk-solvent correction.

3.1. Annealing schedules

The current implementation of the program supports four annealing modes. In the first mode the temperature is kept constant throughout the minimization. The second is a slow-cooling mode, with the temperature linearly dependent on the simulation time. The third mode supports a logarithmic schedule for which the temperature T(k) at each step k of the simulation is given by $T(k) = T_0/\log k$, where T_0 is the starting temperature. In the last mode, the temperature of the system is automatically adjusted in such a way as to keep the fraction of moves made against the gradient of the target-function constant and equal to a user-defined value. This is achieved as follows: the program counts the number of times that a new configuration is accepted even though it results in a worse value for the target function. After a predefined number of iterations, the fraction of moves that have been made against the function gradient is calculated and if it is less than a target value (defined by the user) the temperature is increased, otherwise it is decreased.

3.2. Automatic temperature-limit determination

It is possible to automatically obtain reasonable estimates of the temperature required for a constant and logarithmic temperature run and of a temperature range for a slow-cooling run. This, as shown in Fig. 1, is achieved by monitoring the variation of the average value of the target function as a function of the temperature during a short slow-cooling simulation which is started from a sufficiently remote (high) temperature (this is similar to a specific heat plot from statistical mechanics, see Kirkpatrick *et al.*, 1983). The temperature [$T_{\max(\Delta R)}$] at which the average of the target function shows the greatest reduction is selected for a constant-temperature run and is also the starting temperature for a slow-cooling run. In the case of the logarithmic schedule, the starting temperature is set to a value T_0 such that the temperature $T_{\max(\Delta R)}$ will be reached only after a fraction of ($1/e$) of the total number of moves has already been performed.

A PostScript file containing a graph (similar to the one shown in Fig. 1) is automatically produced by the program. The reason for this is not cosmetic: depending on the nature of the problem, there may well be more than just one phase transition of the system as the temperature is reduced. Fig. 2 shows one such example for a problem with at least two phase transitions. Obviously, for such problems the default treatment of selecting the maximum of this curve (as a starting temperature) may well fail and user intervention would be required.

3.3. Move size control

In this section, we discuss how the current version of the program deals with the problem of how to generate the next configuration (the one that will be tested) from the current configuration (the one that has already been tested). Unfortunately, the selection of an optimal set of possible moves and the control of their magnitudes depends on the nature of the individual problems, making it difficult to find a satisfactory solution without losing generality. Instead of artificially making the optimization problem discontinuous (by restricting the configurational parameters to take values from a predefined fixed grid), we have chosen to work with the continuous case (in which any parameter can take any value from within its defining limits). The program stores the orientational parameters of the search models using the polar angles (ω, φ, κ) convention, with ω defining the latitude and φ the longitude of a rotation axis about which the molecule is rotated by κ°. The translational parameters are stored in terms of the fractional coordinates of the geometrical centres of the molecules in the crystallographic frame of the target structure.

The choice of polar angles simplifies the task of updating and controlling the orientational parameters: for the whole length of the minimization, an orientation for the rotation axis is chosen randomly and uniformly from the full-half sphere (that is $0 \leq \omega \leq \pi/2$ and $0 \leq \varphi < 2\pi$), leaving only the rotational offset $\Delta\kappa$ and the translational offsets Δx, Δy, Δz to be specified before a new configuration can be obtained from the current one. The program supports two modes of move-size control. In the first, the maximum possible values that the random offsets $\Delta\kappa$, Δx, Δy and Δz can take are kept constant throughout the simulation with $\max(\Delta\kappa) = 2d_{\min}$ (in $^\circ$) and $\max(\Delta x, \Delta y, \Delta z) = 2d_{\min}/\max(a, b, c)$, where $d_{\min}$ is the minimum Bragg spacing of the input data and a, b, c are the unit-cell translations of the target structure (in Å). In the second mode, the maximum move sizes (as defined above) are linearly dependent on both time and the current R factor, with

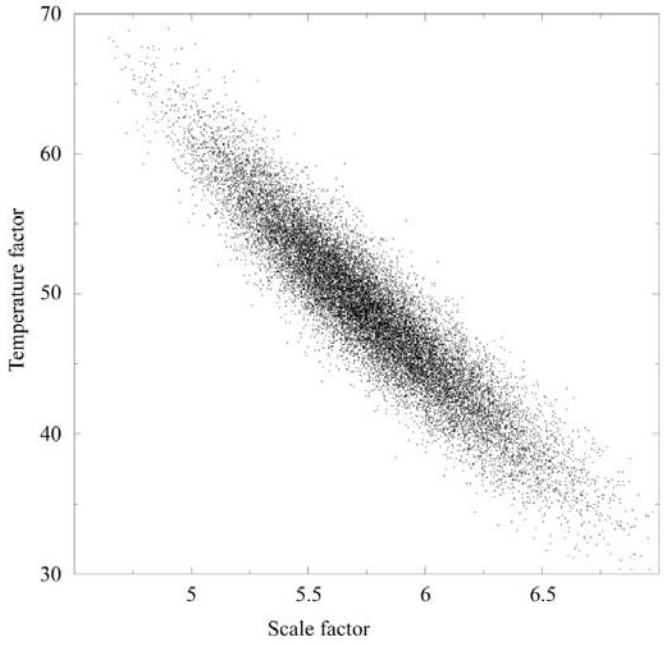

Figure 3
Scatter plot of 20 000 scale and temperature-factor pairs from a minimization performed against real 15–4 Å data obtained from the PDB (see text for details).

$\max(\Delta\kappa) = \pi Rt/t_{\rm total}$ and $\max(\Delta x, \Delta y, \Delta z) = 0.5Rt/t_{\rm total}$, where R is the current R factor, t is the current time step and $t_{\rm total}$ is the total number of time steps for the minimization. The dependence on the R factor is justified on the grounds that as we approach a minimum of the target function, we should be sampling the configurational space on a finer grid.[2] The time dependence follows from a similar argument.

3.4. Target-function selection

In other simulated-annealing problems the target function (whose value is to be optimized) is an integral part of the problem and is thus not a matter of choice. In crystallographic problems, however, the issue of which function to optimize has been (and is some cases, still is) hotly debated. The current thinking in the field clearly points the way to the theoretical (and, nowadays, practically achievable) superiority of a maximum-likelihood function (see, for example, Bricogne, 1988, 1992 and a whole series of papers presented in Dodson *et al.*, 1996). The major problems with the implementation of a maximum-likelihood target in the context of the stochastic multidimensional search described in this communication are that (i) it is not clear how to estimate the σ_A curve (Read, 1997) based on the necessarily small number of reflections (especially for the free R, C-value set; Brünger, 1997) used by this method, (ii) that the σ_A curve would have to be recalculated at each and every step of the algorithm and (iii) that for most of the time these calculations would be pointless given

[2] The word 'grid' is used here metaphorically. For all practical purposes, the values of $\Delta\kappa$, Δx, Δy and Δz returned by the random-number generator are continuous [if, for example, the generator returns values in the range 0–$2^{31}-1$ and $\max(\Delta\kappa) = \pi$, then the 'grid size' on $\Delta\kappa$ is less than 9×10^{-8} °].

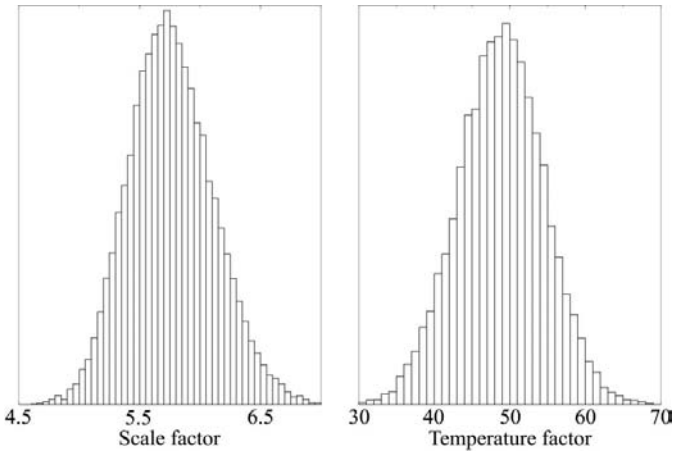

Figure 4
Marginal distributions for the scale and temperature factor obtained from the distribution shown in Fig. 3.

that the majority of the sampled configurations during a minimization are completely wrong (random) structures. Additionally, it is not clear whether the use of a maximum-likelihood target (even if correctly implemented) would indeed offer a significant improvement in the discrimination capabilities of the algorithm. The reason for this is that in contrast with the situation encountered with macromolecular refinement, this method is blessed with an extremely high ratio of observations to parameters (usually in the order of a few hundred reflections per parameter) and that the model is (by being the result of an independent structure determination) totally unbiased towards the observed data.

As was mentioned in §2.2, the currently distributed version of the program supports three user-selectable target functions: the conventional crystallographic R factor and two correlation-based targets, the first of which is calculated over the amplitudes and the second over the intensities of the reflections. In agreement with other studies in the field (Navaza & Saludjian, 1997), we have found that the amplitude-based correlation target appears to perform better than the intensity-based target. Our practical experience has been that when a reasonably accurate starting model is available, there is not a great difference between the performance of the R factor and the amplitude-based correlation target. We suspect that the reason is that with such an overdetermined problem there is little to chose between an accuracy indicator (such as the correlation coefficient; Hauptman, 1982) and a precision indicator (such as the R factor). In an attempt to substantiate this argument, we have performed a series of minimizations during which a modified version of the program was calculating (and writing out) at each step both the R factor and the linear correlation coefficient between the observed and calculated amplitudes. These two sets of statistics were then compared: the linear correlation coefficient between 25 000 pairs of (R factor, 1.0 – Corr) values was found to be 0.682. Given that the R factor is sensitive to the application of an accurate overall scale factor and the fact that all Monte Carlo moves (and not just the accepted ones) were included in

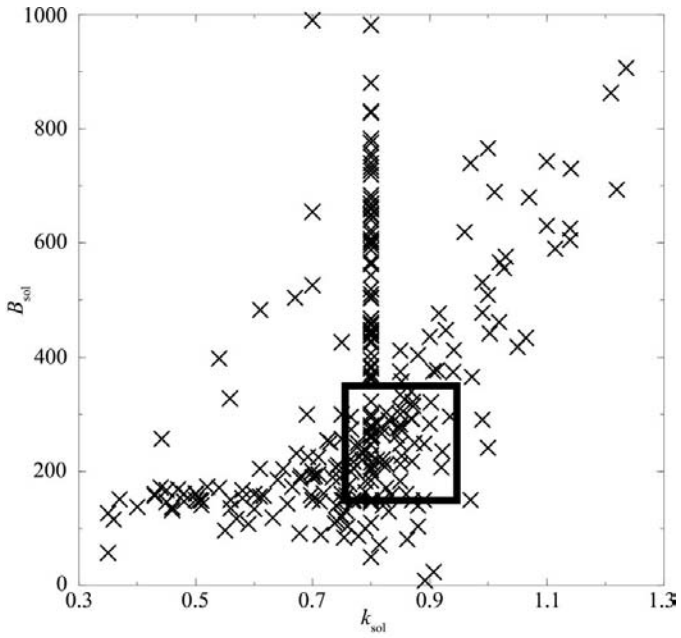

Figure 5
Distribution of the bulk-solvent correction parameters for 301 structures deposited with the PDB. The small rectangular area encloses the usually cited range of values for the two parameters.

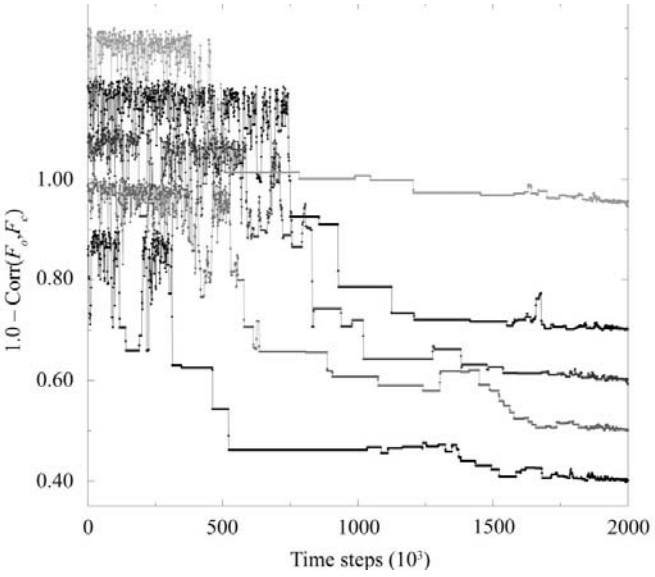

Figure 6
Evolution of the average $1.0 - \mathrm{Corr}(F_o, F_c)$ values for five minimizations from a six-dimensional problem using real data. The $(1 - C)$ values refer only to the lower curve, with the other four curves translated by +0.1, 0.2, 0.3 and 0.4 units to improve clarity. See text for details.

the calculation, the similarity between the two statistics implies that, at least for the case considered here, they provide more-or-less equivalent information about the minimization. This is not to imply that we advocate a return to the *R* factor as a crystallographic target function, nor that we doubt years of accumulated experience on the relative merits of the various functions. All that the preceding analysis suggests is that in the case of the problem under examination and for the specific method of solution, the efficiency of the algorithm appears to be not critically dependent on the choice of the target function (but we should reiterate here that if a good starting model is not available, choosing a precision indicator like the *R* factor as a target function would only exacerbate the problems mentioned in the second paragraph of §2.3).

3.5. Scaling

As discussed in §2.2, the decision as to whether to accept (or reject) a move is based on the difference of the values of the target function before and after this move. Because these differences can be quite small, this stochastic method is sensitive to the algorithm used for scaling the observed and calculated data. The two basic problems with scaling are (i) whether to refine an overall temperature factor or not and (ii) how to correct for the presence of bulk solvent which usually spoils scaling at low resolution (see Fig. 1 of Tronrud, 1997 for an analysed example). The second of these problems will be dealt with in the next section. The question of whether to refine an overall temperature factor, especially for the relatively low resolution ranges used for molecular-replacement calculations, is rather open-ended (clearly, the application of an overall scale factor only matters when the target function for the minimization is the *R* factor. The application of an overall temperature factor affects all three target functions). In our experience, refining both an overall scale and an overall temperature factor is advantageous even at resolutions as low as 4 Å. We will illustrate this with an example using real data obtained from the PDB (Bernstein *et al.*, 1977) (PDB entry 1tgx). Fig. 3 shows the distribution of scale and temperature-factor pairs for the first 20 000 moves of a minimization performed against 15–4 Å data and Fig. 4 shows the marginal distributions for the two parameters.

Not unexpectedly, the distribution in Fig. 3 is skewed, indicating that the two parameters are correlated. What is important, however, is that even though the parameters are correlated, their individual distributions are relatively well behaved (keeping in mind also that this distribution was obtained by determining pairs of scale and temperature factors from an ensemble of effectively random structures): if the marginal distributions (shown in Fig. 4) are least-squares fitted with a Gaussian, then for the scale factor we obtain a mean value of 5.7 with a standard deviation of only 0.35. Similarly, for the temperature factor we obtain a value of 48 ± 5.6. If an overall temperature factor was not being refined, then we would be using an effective $B = 0$, which is approximately 9σ away from the current mean. It can correctly be argued, however, that a suitable value for the overall temperature factor could have been obtained from a Wilson plot. The problem with this approach is that the linear part of a Wilson plot usually does not coincide with the resolution ranges used for molecular-replacement calculations. Additional problems may occur when very low resolution data are used for the calculation and a bulk-solvent correction is not being applied. The default behaviour for the

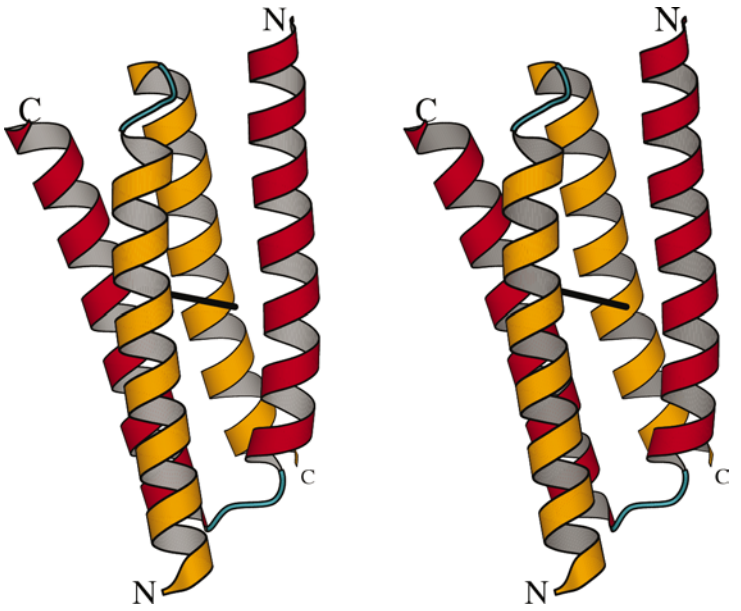

Figure 7
Schematic stereo diagram of the target structure for a six-dimensional problem (1b6q). The colour coding is red and yellow for the helices of each monomer, blue for the connective strands. The position of the intramolecular (crystallographic) dyad axis is also noted. Figure prepared with the program *BOBSCRIPT* (Esnouf, 1997).

currently distributed version of the program is to refine both an overall scale and an overall temperature factor.

3.6. Bulk-solvent correction

The absence of a bulk-solvent correction from the type of calculations described in the previous sections is a serious problem. Not only does it introduce a systematic error for all data to approximately 6 or 5 Å resolution, but it also necessitates the application of a low-resolution cutoff (commonly at ∼15 Å) to compensate for the absence of a suitable correction. This low-resolution cutoff in turn introduces series-termination errors and further complicates the target-function landscape, making the identification of the global minimum more difficult.

Because at each and every step of the minimization we have a complete model for the target crystal structure, it is (at least in principle) possible to perform a proper correction for the presence of bulk solvent, as described for example by Jiang & Brünger (1994) and Badger (1997). The problem, of course, is that if at each step we had to calculate a mask for the protein component, followed by several rounds of refinement for the parameters of the solvent, the resulting program would be too slow to be practical. There is, however, a much faster (but less accurate; Jiang & Brünger, 1994; Kostrewa, 1997) bulk-solvent correction method (known as the exponential scaling model algorithm), which is based on Babinet's principle and is fully described by just one equation,

$$F = F_P(1.0 - k_{\rm sol}\exp\{-B_{\rm sol}[\sin(\theta)/\lambda]^2\}),$$

Table 1
Results from nine constant-temperature minimizations for a 12-dimensional problem.

The solution shown in bold (compared in Fig. 10 with the final structure) is the correct solution.

Minimization	1.0 − Corr(F_o, F_c)	Free value
1	0.2778	0.3162
2	0.2744	0.6903
3	**0.2407**	**0.3305**
4	0.2639	0.3656
5	0.2632	0.8358
6	0.2473	0.4466
7	0.2590	0.4330
8	0.2937	0.2821
9	0.2725	0.6402

where F is the corrected structure-factor amplitude, F_P is the amplitude of the protein component alone, $\sin(\theta)/\lambda$ is reciprocal resolution, $k_{\rm sol}$ is the ratio of the mean electron densities of the solvent and macromolecule and $B_{\rm sol}$ is a measure of the diffuseness (or sharpness) of the boundary between the two components (Moews & Kretsinger, 1975; Tronrud, 1997). This physical interpretation of the meaning of $k_{\rm sol}$ and $B_{\rm sol}$ is only valid when the initial assumption is satisfied; that is, when the electron-density distribution for both the macromolecular and solvent components is uniform. Because this can only be true at very low resolution, it is common practice (at least in the case of macromolecular refinement) not to fix their values, but instead to allow $k_{\rm sol}$ and $B_{\rm sol}$ to enter the refinement as two independent (adjustable) parameters whose values determine the contribution from the bulk solvent (see Tronrud, 1997 for a discussion of the refinement procedure).

Unfortunately, addition of several rounds of non-linear least-squares refinement of the $k_{\rm sol}$ and $B_{\rm sol}$ parameters in the proposed molecular-replacement method would make the resulting program rather impractical. Nevertheless, given the physical meaning of the two parameters and the fact that the great majority of proteins crystallize under rather similar conditions led us to believe that a reasonable trade-off between speed and accuracy could be achieved: this we intended to do by fixing the values of $k_{\rm sol}$ and $B_{\rm sol}$ to the centroid of the distribution obtained from all deposited (in the PDB) pairs of values for the two parameters (and for structures refined with the exponential scaling model algorithm). By doing so, not only we could avoid continuous cycles of parameter refinement, but we could actually calculate the value of the correction term $\{1.0 - k_{\rm sol}\exp[-B_{\rm sol}(\sin(\theta/\lambda)^2]\}$ even before the minimizations begin. The result would be that for the whole length of the calculations the computational cost of performing a bulk-solvent correction would be just one multiplication per reflection per cycle.

Our hope that this would be a viable method to correct for the bulk solvent was reinforced by the fact that the usually quoted range of values for the parameters is rather narrow, 0.75–0.95 for $k_{\rm sol}$ and 150–350 Å^2 for $B_{\rm sol}$, as given by Tronrud (1997), Badger (1997) and Kostrewa (1997). Unfortunately, as

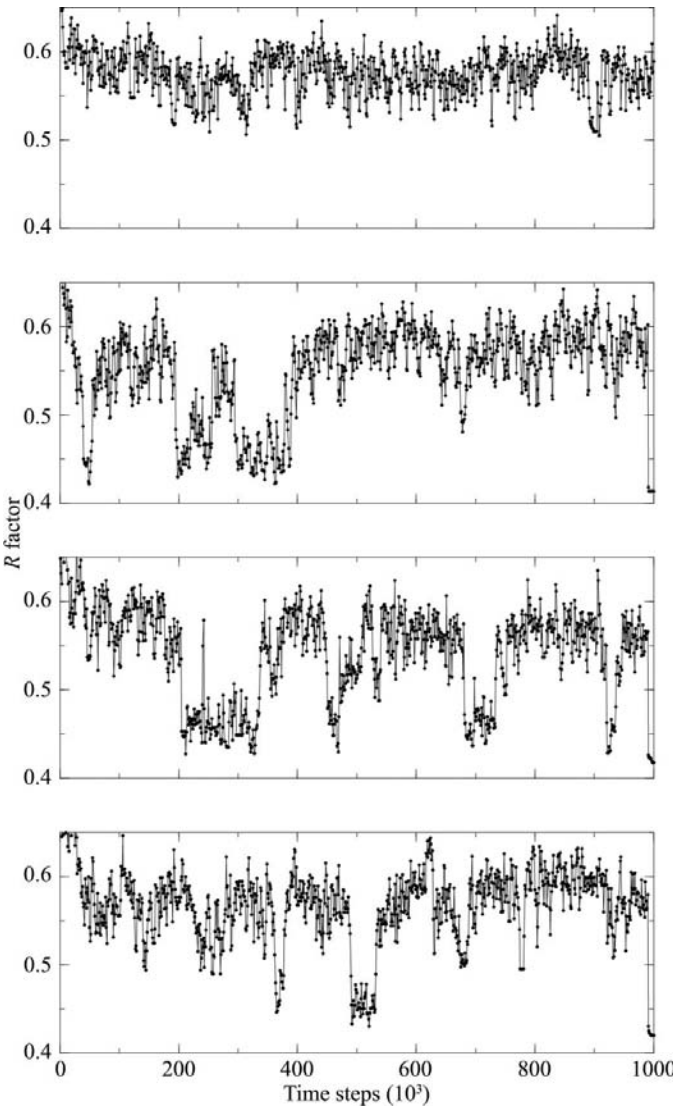

Figure 8
Evolution of the average *R*-factor values for four minimizations from a six-dimensional problem. The minimization shown in the top graph failed to find the correct solution, whereas the other three minimizations converged to the correct solution with a *R* factor of about 0.42. See text for details.

shown in Fig. 5, the distribution obtained from 301 structures deposited with the PDB showed anything but a tight clustering around a value in this range. Because this distribution (and some of its possible interpretations) has already been discussed extensively (Glykos & Kokkinidis, 2000*b*), it suffices here to say that the currently distributed version of the program can perform a bulk-solvent correction (given a pair of values for the two solvent parameters), but the feature is not turned on by default and its application has to be explicitly requested by the user.

3.7. Implementation-specific limitations

There are several limitations of the program described in the previous sections which arise not from the method *per se*, but from its practical implementation. The most important is that the current version of the program is limited to target crystal structures consisting exclusively of only one molecular species. The reason for not implementing a more general treatment is that the physical memory requirements for storing simultaneously two (or more) molecular transforms would make the program impractical for most users (but this is slowly changing). A second (not unrelated) limitation is that the molecular structure of the search model is kept fixed throughout the calculation. Again, there is no practical way for modifying the search model during the calculation without losing the advantages offered by a pre-calculated molecular transform. (The obvious solution would be to treat the individual domains or other substructure as independent search models, but this would not only be impractical owing to physical memory limitations, but would also unnecessarily increase the number of free parameters and the dimensionality of the problem).

Another limitation concerns the automatic temperature determination algorithm presented in §3.2. The problem with the approach presented there, is that too much faith is placed on the behaviour of the average value of the target function as observed in just one quickly performed slow-cooling protocol. If the behaviour of the target function is not typical of a set of non-productive random walks on the target function landscape (as the current implementation assumes), then the algorithm presented above will be at the mercy of the idiosyncratic peculiarities encountered during this specific simulation.

One final problem concerns the incorporation of known non-crystallographic symmetry elements (determined, for example, from the self-rotation or Patterson functions) in the calculations described above. In the case of exclusively translational non-crystallographic symmetry, this prior knowledge can be directly incorporated in the current implementation of the program (in the form of additional fixed symmetry elements). Incorporation of general non-crystallographic symmetry elements restraints is not possible with the current implementation of the program, as this would entail independent refinement of the positions of all non-crystallographic symmetry axes with a known orientation.

4. Results

In this section, we present results from the application of the program to molecular-replacement problems of varying complexity. In all cases, the results will be presented in the form of graphs on which the horizontal axis is time steps (number of iterations) of the Monte Carlo algorithm and the vertical axis is the value of the target function for the minimization.

4.1. A six-dimensional problem (1)

The first example shows results from a slow-cooling six-dimensional search performed using real data obtained from the PDB entry for the orthorhombic form of chicken egg-

white lysozyme (PDB entry 1aki). The space group of the target structure is $P2_12_12_1$, with unit-cell parameters $a = 59.06$, $b = 68.45$, $c = 30.51$ Å and one molecule per asymmetric unit. The search model for this calculation was Japanese quail lysozyme (PDB entry 2ihl) which has an r.m.s. deviation from the target structure of 1.2 Å and a maximum displacement of 8.7 Å. Fig. 6 shows the evolution of the average $1.0 - \mathrm{Corr}(F_o, F_c)$ values from five independent minimizations using the 558 strongest reflections between 15 and 4.0 Å resolution (about 50% of all data to this resolution).

As is obvious from this figure, four out of five minimizations converged to a deep minimum of the target function, corresponding to the correct solution [with $(1 - C)$ values of about 0.39]. The total CPU time for each minimization was 118 min, with the four solutions of the successful runs appearing after 30, 36, 59 and 64 min, respectively.[3]

4.2. A six-dimensional problem (2)

The example presented in the previous section could have trivially been solved using any of the standard Patterson-based molecular-replacement programs and in a fraction of the time required by this six-dimensional search [*AMoRe*, for example (Navaza & Saludjian, 1997; Navaza, 1994), solves this same problem in less than 3 min of CPU time]. In this section, we present results from a problem that defeats traditional methods by violating the assumption of topological segregation of the self-and cross-vectors. The target structure for this problem is 1b6q, a homodimeric 4-α-helical bundle (a schematic diagram of which shown in Fig. 7) which crystallizes in space group $C222_1$, with unit-cell parameters $a = 30.4$, $b = 42.1$, $c = 81.4$ Å, one monomer (half a bundle) in the asymmetric unit and very low solvent content (approximately 30%).

As can be seen from Fig. 7, the self-vectors between, for example, the two helices of the red-coloured monomer, are on the average longer that the cross vectors between helices belonging to different monomers (a red and a yellow helix in this figure). Because this is also a tightly packed structure, the result is that the cross vectors within any chosen integration radius around the origin of the Patterson function will be approximately as numerous as the self-vectors. The search molecule used for the calculations was an essentially perfect polyalanine model of the two helices (with an r.m.s. deviation of less that 0.2 Å for the included atoms) and we used real data (collected on a CAD4 diffractometer) between 15 and 4 Å resolution. Although the search model is exceptionally accurate and the data of high quality, conventional methods [program *MOLREP* (Vagin & Teplyakov, 1997) from the *CCP*4 suite of programs (Collaborative Computational Project, Number 4, 1994)] cannot identify the correct solution during the default run.

In contrast, a six-dimensional search which was performed using the same search model and data successfully identified the correct solution. Fig. 8 shows the evolution of the average *R*-factor values from five independent minimizations using the 353 strongest reflections between 15 and 4.0 Å resolution (about 70% of all data). As it is obvious from this figure, three out of four minimizations converged to a deep minimum of the *R* factor that corresponds to the correct solution (with *R* values of about 0.42). The total CPU time for each these simulations was 23 min.

4.3. A 12-dimensional problem

Although the example presented in the previous section was sufficiently complex to defy solution by Patterson function based methods, it was still far from realistic: in a real molecular-replacement problem we would hardly expect to have such an accurate starting model, both for the individual helices and especially with respect to their relative positions and orientations. A far more demanding (and thus realistic) problem would be to try to determine this structure using as a starting model just one polyalanine helix taken from an independently determined structure. An (unsuccessful) attempt to find a solution to this problem through an exhaustive search performed with *AMoRe* has already been described in the original structure determination of this protein (Glykos & Kokkinidis, 1999).[4] Here, we show that the structure could have been solved (although not trivially) with a full 12-dimensional search performed with this stochastic method. The search model was the helical polyalanine part of residues 4–29 of 1rpo, amounting to a total of only 130 atoms (less than 25% of the total number of ordered atoms in the structure). The r.m.s. deviation between the search model and the two target helices were 0.7 and 0.9 Å; we only used data between 15 and 4 Å resolution.

Table 1 shows the final (best) values of the target function [in this case, $1.0 - \mathrm{Corr}(F_o, F_c)$] and its corresponding free set for the nine minimizations performed; Fig. 9 shows the evolution of the average $1.0 - \mathrm{Corr}(F_o, F_c)$ values for two of these simulations (3 and 4 in Table 1). The solution corresponding to minimization number 3 is compared in Fig. 10 with the final structure. As can be seen from this figure, for the minimization with the best value of the target function the two search models are approximately correctly placed and oriented (and within the convergence radius of rigid-body refinement at that resolution). Furthermore, the polarity (direction) of the helices is also correctly predicted and would

[3] All references to physical time measurements of the program's speed of execution refer to a UNIX workstation which in single-user mode gave the following SPEC95 benchmark results: SPECint95 = 16.6, SPECint_rate95 = 149, SPECfp95 = 21.9 and SPECfp_rate95 = 197 (Standard Performance Evaluation Corporation, 10754 Ambassador Drive, Suite 201, Manassas, VA 21109, USA; http://www.specbench.org/). UNIX is a registered trademark of UNIX System Laboratories, Inc.

[4] This search was conducted as follows: one polyalanine helix was fixed in orientation and position by combining the best 99 orientations from its cross-rotation function with the top 20 peaks from each of the corresponding translation functions, giving a total of 1980 starting models for the first helix. For each of these models, we calculated the translation functions corresponding to each and every of the 99 best orientations for a second copy of the model, giving a grand total of $1980 \times 99 = 196\,020$ translation functions or a list of 3 920 400 correlation coefficients. This search resulted in a more or less uniform distribution of the linear correlation coefficients from the translation functions, with the best solutions being clearly wrong as judged by packing considerations.

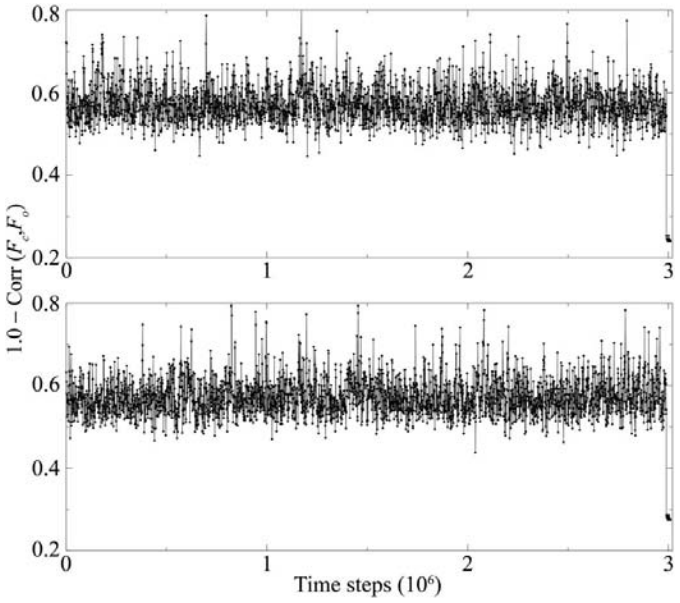

Figure 9
Evolution of the average 1.0 − Corr(F_o, F_c) values for two constant-temperature minimizations (entries 3 and 4 in Table 1) from a 12-dimensional problem. The sudden drop of the target function near the end of the simulations corresponds to the beginning of the final few thousand cycles of refinement of the best solution encountered during the main length of the minimization.

probably have allowed the structure determination of this protein to proceed to completion.

We should not like, however, to leave an impression that determining this structure would have been trivial with this 12-dimensional search. As Table 1 (and Fig. 9) show, the correct solution is hardly identifiable from the set of the other (wrong) solutions. There are several reasons for this. The first is that all nine minimizations converged to closely related solutions, with very similar packing arrangements. Their differences are mostly accounted for by rotations about the helical axes combined by translations approximately equal to the length of one helical turn. A second reason is that the search model is rather incomplete and that relatively low-resolution data are used. Under those circumstances, the value of the target function for the correct solution is not significantly lower than its value for some of the wrong (local) minima. Furthermore, owing to the necessarily small number of reflections that enter the calculation, the standard uncertainty of the free R (or 1.0 − C) value is so large that even wrong solutions can give quite respectable free values (for example, minimizations 1 and 8 in Table 1), further complicating the identification of the correct solution.

4.4. A three-body problem

With this last example we attempt to address the question of what is the practical limit for the number of search models per asymmetric unit (of the target crystal structure) that can be tackled with this method. Clearly, the answer to this question depends so much on the specifics of the problem under examination that it is impossible to justifiably give a single answer that would cover all cases. To reinforce this statement about the dependence on the characteristics of the individual problems, we present in Fig. 11 results from a trivial, but nonetheless 15-dimensional, three-body problem which this program can solve in less than 20 min of CPU time per minimization. Data for this example were calculated from the PDB entry 1a7y containing the atomic coordinates of the 11-residue antibiotic actinomycin D which crystallizes in space group $P1$ with three molecules per asymmetric unit and unit-cell parameters a = 15.737, b = 15.887, c = 25.156 Å, α = 85.93, β = 86.19, γ = 69.86°. The three molecules in the asymmetric unit were forced to be identical by replacing chains B and C with the coordinates of chain A; we only used data between 25 and

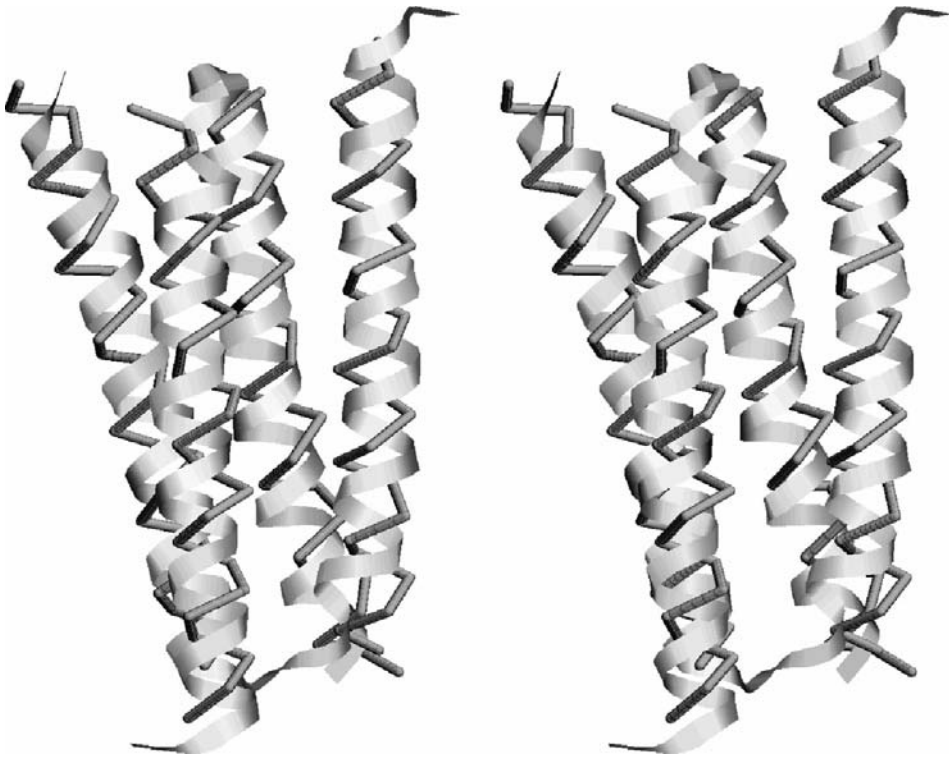

Figure 10
Schematic stereo diagram of the target structure (1b6q, shown as a ribbon diagram) and of the best solution from a 12-dimensional search performed using as model one polyalanine helix (stick models). The orientation of the molecule is identical to the one shown in Fig. 7. Figure prepared with the program *RASMOL* (Collaborative Computational Project, Number 4, 1994).

2 Å resolution. To make the example somewhat more realistic, the input data were modified by adding an offset ranging randomly and uniformly from −20 to +20% of their modulus. This 'noisy' data set was treated as the observed data set (of the target structure). As can be seen from Fig. 11, all three minimizations converged to deep minima of the *R* factor, with each minimization taking approximately 58 min of CPU time and individual solutions appearing after 10, 15 and 17 min. Two simulations (upper two graphs in Fig. 11) converged to an *R* factor of ∼20%, whereas the last solution (lower curve) converged to an *R* factor of 11%. The reason for the difference between the values of the target function is that the search model has an approximate internal dyad axis of symmetry, which at the resolution used for this calculation gives rise to two approximately equivalent orientations for each molecule, with one orientation slightly better than the other. The difference of the *R* factors reflects a difference in the proportion of the search models that have been placed in one (or the other) of the two orientations.

Although this hypothetical structure is by all accounts a rather trivial problem to solve, it does make the point that the high dimensionality of the search space is not in itself sufficient for invalidating the application of this method. We should stress, however, that our practical experience with the application of this program is that when there are more than three molecules per asymmetric unit, the so called 'curse of dimensionality', combined with less-than-ideal search model and data, makes the application of this 6*n*-dimensional procedure unjustifiably expensive in terms of computational requirements [and as a last cautionary tale, we should add here that this program has never (at least to our knowledge) been able to solve a problem with more than three molecules per asymmetric unit].

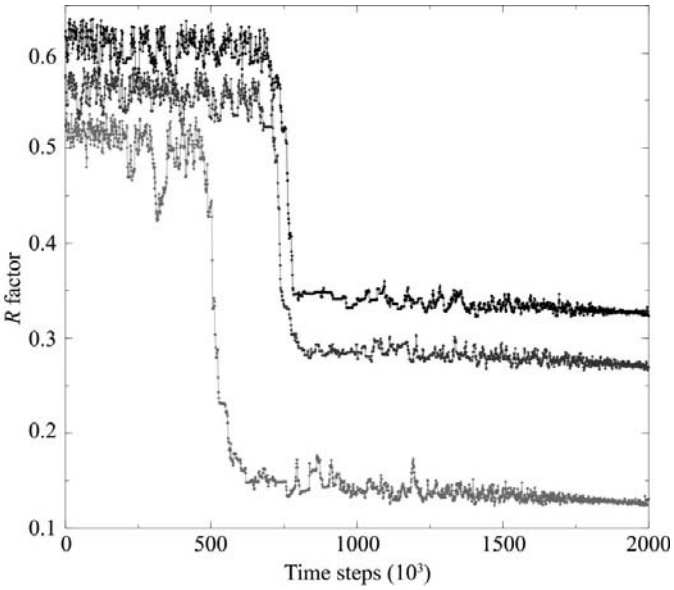

Figure 11
Evolution of the average *R* factors for three slow-cooling simulations from a 15-dimensional problem. The *R*-factor values refer only to the lower curve, with the other two curves translated by +0.05 and +0.10 units to improve clarity. See text for details.

5. Discussion

We showed that a stochastic molecular-replacement method which is able to determine the rotational and translational parameters of all search models simultaneously is not only feasible but also practical for molecular-replacement problems ranging in complexity from relatively straightforward six-dimensional optimizations, to quite complex 12- and even 15-dimensional problems. In this final section, we discuss the status of the method from the viewpoint of the practising crystallographer and present what we think are the future perspectives for this class of algorithms.

We do not believe that this method could (or should) compete with the well established Patterson-based molecular-replacement programs. These methods (and the corresponding programs), when combined with careful thinking and examination of all available evidence, have repeatedly been shown to be able to solve problems far more difficult and demanding than the examples presented in this communication and at a fraction of the computational cost required by this method. Nevertheless, as the examples in §§4.2 and 4.3 illustrated, there do exist classes of problems which are tractable through this multidimensional approach but defeat most other methods. For this reason, we view our method as a last-ditch effort to solve by pure computational means molecular-replacement problems that resisted solution by other automated methods (but we feel that we should add that substituting computing for thinking has repeatedly been shown to fail even for problems much simpler that those presented in the previous sections).

As was discussed in §§2.3 and 3.7, there is a significant number of limitations of this approach, both methodological and implementation-specific. However, what we think that is really missing from the method is the ability to integrate and take into account all the evidence and information available for any given problem. To give just one example: instead of treating all orientations of the search model as *a priori* equally probable, it should be possible to treat the cross-rotation function (calculated for the given search model and data) as a probability-distribution function and then force the search model(s) to sample the orientational space in such a way as to keep the fraction of time spent at each orientation proportional to the value of the cross-rotation function for this orientation. In this way, and by reducing the amount of time spent on exploring combinations of parameters which are deemed improbable by the evidence in hand (in this case, the cross-rotation function), we would be performing an 'importance sampling' on the orientational parameters of the search model(s). Additional information which may enter this type of calculations can come, for example, from the self-rotation function (which could relatively easily be used to enforce non-crystallographic symmetry in the orientational probability-distribution function mentioned above) or from the presence of purely translational non-crystallographic symmetry

detectable from the native Patterson function.[5] Clearly, keeping track of all these disparate sources of information and combining them in a meaningful and computationally robust algorithm is not a trivial task, but we believe that such a method would open the way to structure determinations which are outside the reach of the currently implemented molecular-replacement techniques.

6. Program availability

The program (*Queen of Spades*) described in this paper is open-source software which is distributed free of charge to both academic and non-academic users and is immediately available for download from http://origin.imbb.forth.gr/~glykos/. The distribution contains source code, documentation (manual page, PostScript and html), example scripts, test files and stand-alone executable images suitable for the majority of the most commonly used workstation architectures.

References

Badger, J. (1997). *Methods Enzymol.* **277**, 344–352.

Bernstein, F. C., Koetzle, T. F., Williams, G. J., Meyer, E. F. Jr, Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535–542.

Bricogne, G. (1988). *Acta Cryst.* A**44**, 517–545.

Bricogne, G. (1992). *Proceedings of the CCP4 Study Weekend. Molecular Replacement*, edited by E. J. Dodson, S. Gover & W. Wolf, pp. 62–75. Warrington: Daresbury Laboratory.

[5] It should be mentioned, however, that even this simple proposition, *i.e.* to use the cross-rotation function as an orientational probability distribution, is still an oversimplification for problems with more than one molecule per asymmetric unit. The reason is that for such problems the probability distribution for the orientation of one molecule ought to be treated as conditional on the orientation of the other search models. One way that this could be achieved is through the active use of the self-rotation function as a means to calculate, based on the probability distribution for the orientation of one of the search models, the orientational probability distributions for the rest of the molecules (which is a generalization of the principle behind the locked rotation function; Tong & Rossmann, 1990). It goes without saying that the computational cost for performing such a calculation (which would involve updating the orientational probability distributions at each and every step) would be prohibitive with present-day computing capabilities.

Brünger, A. T. (1997). *Methods Enzymol.* **277**, 366–396.

Carter, C. W. Jr & Sweet, R. M. (1997). *Methods Enzymol.* **276**, 558–618.

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

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

Dodson, E. J., Gover, S. & Wolf, W. (1992). Editors. *Proceedings of the CCP4 Study Weekend. Molecular Replacement.* Warrington: Daresbury Laboratory.

Dodson, E. J., Moore, M., Ralph, A. & Bailey, S. (1996). Editors. *Proceedings of the CCP4 Study Weekend. Macromolecular Refinement.* Warrington: Daresbury Laboratory.

Esnouf, R. M. (1997). *J. Mol. Graph.* **15**, 132–134.

Glykos, N. M. & Kokkinidis, M. (1999). *Acta Cryst.* D**55**, 1301–1308.

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

Glykos, N. M. & Kokkinidis, M. (2000*b*). *Acta Cryst.* D**56**, 1070–1072.

Hauptman, H. (1982). *Acta Cryst.* A**38**, 289–294.

Ingber, L. A. (1993). *J. Math. Comput. Model.* **18**, 29–57.

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

Keen, D. A. & McGreevy, R. L. (1990). *Nature (London)*, **344**, 423–425.

Kirkpatrick, S., Gelatt, C. D. Jr & Vecchi, M. P. (1983). *Science*, **220**, 671–680.

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

Kostrewa, D. (1997). *CCP4 Newsl.* **34**.

McGreevy, R. L. & Pusztai, L. (1988). *Mol. Simul.* **1**, 359–367.

Machin, P. A. (1985). Editor. *Proceedings of the CCP4 Study Weekend. Molecular Replacement.* Warrington: Daresbury Laboratory.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953). *J. Chem. Phys.* **21**, 1087–1092.

Moews, P. C. & Kretsinger, R. H. (1975). *J. Mol. Biol.* **91**, 201–228.

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

Navaza, J., Panepucci, E. H. & Martin, C. (1998). *Acta Cryst.* D**54**, 817–821.

Navaza, J. & Saludjian, P. (1997). *Methods Enzymol.* **276**, 581–593.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992). *Numerical Recipes in C. The Art of Scientific Computing*, 2nd ed. Cambridge University Press.

Read, R. J. (1997). *Methods Enzymol.* **277**, 110–128.

Rossmann, M. G. (1972). *The Molecular Replacement Method.* London: Gordon & Breach.

Rossmann, M. G. (1990). *Acta Cryst.* A**46**, 73–82.

Rossmann, M. G. & Blow, D. M. (1962). *Acta Cryst.* **15**, 24–31.

Sheriff, S., Klei, H. E. & Davis, M. E. (1999). *J. Appl. Cryst.* **32**, 98–101.

Tong, L. & Rossmann, M. G. (1990). *Acta Cryst.* A**46**, 783–792.

Tronrud, D. E. (1997). *Methods Enzymol.* **276**, 306–319.

Vagin, A. & Teplyakov, A. (1997). *J. Appl. Cryst.* **30**, 1022–1025.