Automated search-model discovery and preparation for structure solution by molecular replacement

**Ronan M. Keegan and
Martyn D. Winn***

Computational Science and Engineering
Department, CCLRC Daresbury Laboratory,
Daresbury, Warrington WA4 4AD, England

Correspondence e-mail: m.d.winn@dl.ac.uk

A novel automation pipeline for macromolecular structure solution by molecular replacement is described. There is a special emphasis on the discovery and preparation of a large number of search models, all of which can be passed to the core molecular-replacement programs. For routine molecular-replacement problems, the pipeline automates what a crystallographer might do and its value is simply one of convenience. For more difficult cases, the pipeline aims to discover the particular template structure and model edits required to produce a viable search model and may succeed in finding an efficacious combination that would be missed otherwise. The pipeline is described in detail and a number of examples are given. The examples are chosen to illustrate successes in real crystallography problems and also particular features of the pipeline. It is concluded that exploring a range of search models automatically can be valuable in many cases.

1. Introduction

The number of models of macromolecular structures deposited in the Protein Data Bank (PDB; Berman *et al.*, 2000) continues to grow apace. This mostly reflects advances in crystallography, although other techniques such as NMR are also maturing. As well as providing input to further theoretical modelling or experiment design, the PDB increasingly supports further structure determination through the technique of molecular replacement (MR). As the size of the PDB increases, the relative significance of the MR technique is widely expected to grow.

While there continue to be advances in the underlying algorithms, the software effort in macromolecular structure determination is increasingly turning towards the development of automation schemes which link together several steps of the structure-solution pipeline. Part of the reason is to make the application of crystallographic techniques easier so that scientists can focus on the biology, but automation also allows a more exhaustive search of relevant methods and parameters which may increase the quality of the solution. In the context of MR, a crystallographer frequently has to try a large number of search models and MR programs before finding the optimum solution, or indeed any solution, and this aspect is clearly ripe for automation. For example, Jaskólski *et al.* (2006) provided results for a range of search models and solution methods for the case of a retroviral protease HTLV-1 and concluded that

> when many possible models are available, all should be investigated as potential starting points.

In recent years, a number of automated pipelines and services based around or including the MR technique have

been developed. These include those developed to support structural genomics consortia (for example, Rupp *et al.*, 2002; Fu *et al.*, 2005). Publically available services include the TB Consortium Bias Removal Server (Reddy *et al.*, 2003), *CaspR* (Claude *et al.*, 2004) and *BRUTEPTF* (Strokopytov *et al.*, 2005). Other developments include *Auto-Rickshaw* (Panjikar *et al.*, 2005), which is principally for experimental phasing but covers phased MR as well, *Balbes* (Long *et al.*, 2007) and a scheme for using comparative models in MR (Giorgetti *et al.*, 2005).

In this article, we describe an automated package for structure solution by molecular replacement, called *MrBUMP*. In essence, the package consists of a set of Python scripts which link together established programs. We believe that *MrBUMP* differs from other molecular-replacement pipelines in two ways. Firstly, there is a greater emphasis on the discovery of potential search models. A list of search models is generated and many are tried in MR with the aim of finding the optimum model. Secondly, *MrBUMP* is intended for general use within the *CCP*4 software suite (Collaborative Computational Project, Number 4, 1994) and is thus designed to be portable and flexible.

2. Design issues

2.1. Philosophy of *MrBUMP*

There is a wealth of literature devoted to elucidating the optimum approach to MR. Nevertheless, for each structure solution, the crystallographer is interested in finding a search model that is as close to his/her target structure as possible and requires an approach suited to that particular (unknown) target, rather than necessarily the most widely applicable approach. From the outset, therefore, we chose to develop a framework which allowed a range of techniques to be employed and strove not to impose any preconceived ideas. Such a framework approach also means that new programs can be incorporated relatively easily.

For a given target, *MrBUMP* tries a long list of potential search models based on different proteins and on different search model-generation techniques. The search is exhaustive rather than fast, but we feel that in the context of a crystallography project this is not a major limitation. Search models are ranked so that there is a reasonable chance that good solutions will appear early, but unexpected hits are allowed for.

In favourable cases, this approach gives a 'one-button' solution, with the output of *MrBUMP* ready for model completion and submission. In unfavourable cases, the results of *MrBUMP* will suggest likely search models for further manual investigation.

2.2. Scope of the automation scheme

The aim of *MrBUMP* is to start from native structure factors and a target sequence and deliver a positioned and partly refined model suitable for further model rebuilding, model completion and/or refinement. In the highest level view,

the process consists of three stages: discovery of search-model templates, construction of search models from these templates and molecular replacement itself. In this section, we give an overview of these stages, with more details being given in §3.

In the first stage, the target sequence is used to search for related proteins in the PDB, using a simple pairwise alignment as implemented in the *FASTA* package (Pearson & Lipman, 1988). *MrBUMP* does not currently support the detection of remote homologues using iterative searches such as those implemented in *psiBLAST* (Altschul *et al.*, 1997) and *FFAS* (Jaroszewski *et al.*, 2005). If such remote homologues are known, or if there are models held in local PDB files, they can be entered manually by the user and added explicitly to the template list.

An optional following step is to submit the structure of the top *FASTA* hit to the *SSM* service (Krissinel & Henrick, 2004), which may find additional PDB entries that were not picked up in the initial sequence search. Such entries are structurally similar (based on the secondary-structure elements) to the top match of the *FASTA* search; the hope is that such structures are also structurally similar to the target.

These first searches give a list of complete protein chains that are templates for search models. However, the target may consist of or include a domain that is a constituent domain of another protein and in this case the optimum search model would be based on the domain only. Therefore, the SCOP database (Murzin *et al.*, 1995; Lo Conte *et al.*, 2002) is searched to see if any of the current list of templates contain known domains which could be used as search-model templates. This approach is conservative, in that the SCOP database only contains domains that appear in more than one PDB structure and does not include domains which may be obvious from the structure but which are unique in the PDB. In addition, the domains are derived from chains found in the original *FASTA* search, rather than being searched for directly.

Similarly, the PQS database (Henrick & Thornton, 1998) is searched to see if any of the search templates are predicted to exist in a multimeric form. The aim of the PQS database is to identify multimers that are biologically relevant, rather than simply artefacts of the crystal packing, and such multimers might be expected to be transferrable between crystal structures. The advantage of using a multimer as a search model is that the larger structure should give a larger signal in the MR search, which may be critical in finding a solution, especially if the target asymmetric unit contains many protein chains. However, if the arrangement of monomers in the multimeric search model differs from that in the target, then the MR search will fail. This is seen, for example, in a trial solution of 1vlw, which contains a trimer in the asymmetric unit. Trimeric search models based on several related proteins, such as 1eua, fail, whereas solution based on monomeric search models is straighforward. Manual comparison of the trimeric search models against the deposited 1vlw and against the successful monomer-based solutions shows that the relative positions of the monomers within the trimer differ significantly, possibly because of differing crystal-packing environments.

The set of sequences of the template chains and domains are aligned using a multiple alignment program such as *MAFFT* (Katoh *et al.*, 2005) or *ClustalW* (Chenna *et al.*, 2003). From this multiple alignment, pairwise alignments between the target and each of the templates are extracted. These alignments are expected to be more accurate than the simple pairwise alignments given by the initial *FASTA* search and are used by *CHAINSAW* in the model-building step (see below).

The multiple alignment is also used to score the template models. The score is based on the sequence identity of the template to the target and the extent and nature of the alignment (see §3). While the sequence identity between the target and the search model is a well known criterion for success in MR, the role of the alignment extent is less well characterized. While a longer alignment is obviously better, there are many cases of successful MR-based structure solution using a search model that represents only a fraction of the target. The current scoring function in *MrBUMP* attempts to achieve a balance between sequence identity and alignment extent, but this is certainly worthy of further study.

The list of template models is ranked according to this score and a subset is passed to the search model-preparation stage of *MrBUMP*, which generates an actual MR search model from the template structure. *MrBUMP* currently implements four methods, which differ in the use of alignment between target and template and in the treatment of side chains. The method based on *CHAINSAW* (Stein, 2007) uses the alignment to the target generated previously, while the method based on *MOLREP* (Vagin & Teplyakov, 1997) derives its own alignment. In both cases, sections of the template structure which do not align to the target are removed, while in the remaining two methods all the template main chain is retained. The latter are the PDBclip method, which simply tidies the template structure, and a standard polyalanine model.

The search model-generation methods based on *CHAINSAW* and *MOLREP* are the most sophisticated of the four used by *MrBUMP*, but PDBclip and poly-alanine models are also commonly used. These methods can be run in parallel in order to determine without prejudice which is the most useful in a given case. In addition to the set of single search models, an ensemble of the top search models is also created for input to the program *Phaser* (McCoy *et al.*, 2005).

The list of search models is passed to the MR stage, which uses *MOLREP* (Vagin & Teplyakov, 1997) or *Phaser* (McCoy *et al.*, 2005) or both. If the MR program finds a solution, then the positioned model is passed to *REFMAC* (Murshudov *et al.*, 1997) for 30 cycles of restrained refinement. A criterion based on the behaviour of the free R factor is used to alert the user to a possible successful structure solution. Other statistics, such as the scores from the MR program, are available to the user but are not currently used directly by *MrBUMP*.

2.3. Remote services *versus* local calculations

The early stages of the *MrBUMP* process rely on bioinformatics tools and databases. There is a choice to be made here on whether to implement everything locally or to use existing services available over the internet. In general, we have chosen to take the latter route, in the first instance using services provided by the European Bioinfor-
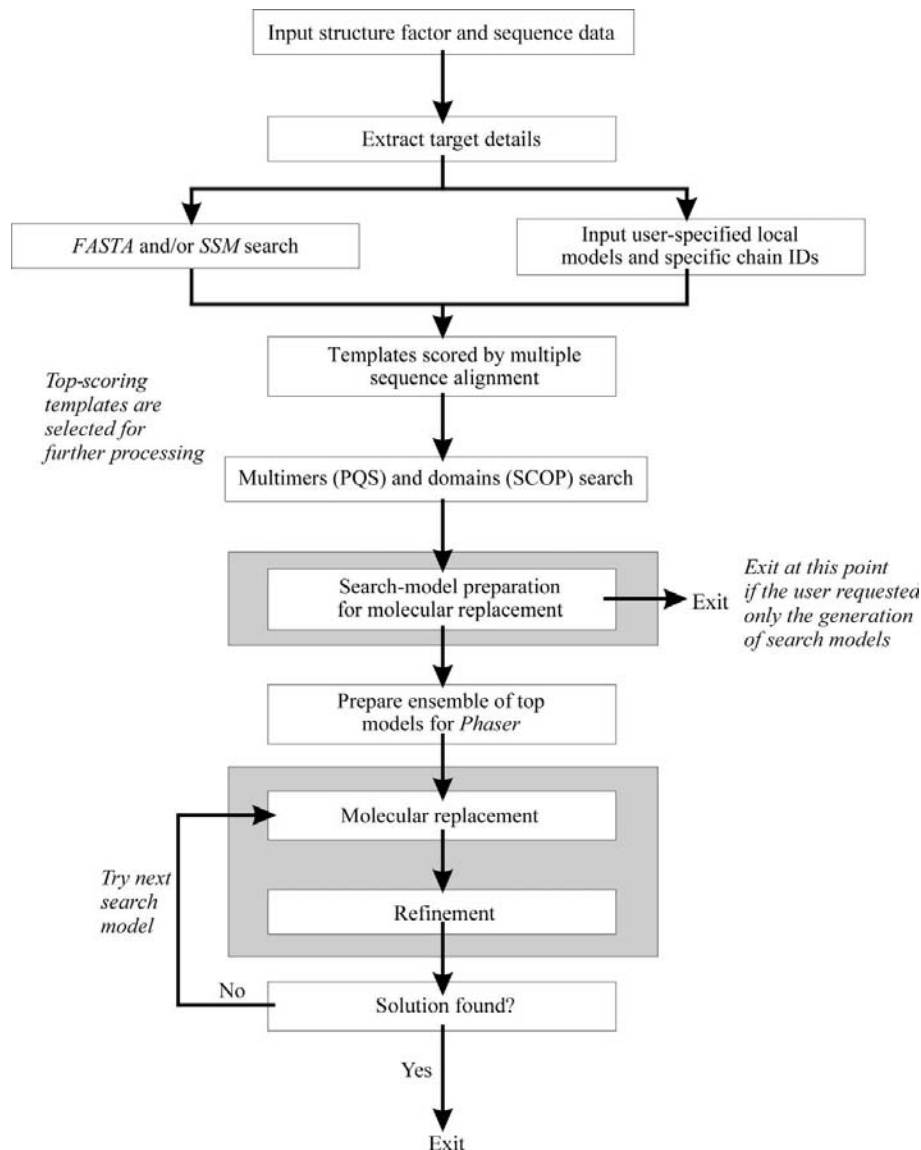

Figure 1
Flow diagram of the steps performed in a full run of *MrBUMP*.

matics Institute (EBI). The advantage is clearly that such services are actively maintained and are likely to be reliable and up to date. The disadvantages are the dependency on the network, the lack of control of the form of the data and possible privacy issues.

Privacy is particularly relevant for the first step in which the target sequence is used to search for possible homologues. Therefore, *MrBUMP* includes the option to perform this step locally, provided a local version of the *FASTA* program is installed. Steps which still require the network are the download of the PDB files, the *SSM* search, download of the PQS and SCOP database files and download of the PQS multimers. All these steps concern the putative search models and contain no direct information about the target.

From the point of view of bioinformatics service providers, *MrBUMP* is not expected to be too demanding. The searches are relatively small and are performed only once. The highest load comes from the download of a possibly large number of coordinate files. *MrBUMP* is therefore organized so that coordinate files are only actually downloaded if they are going to be used as search models in the MR step.

3. Implementation in *MrBUMP*

MrBUMP is a Python program which accesses several of the most commonly used programs in the *CCP*4 suite for performing molecular replacement, along with some additional applications for retrieving, scoring and preparing search models to be used in the MR step. Fig. 1 shows a flow diagram for *MrBUMP*. The flow through the program can be broken down into several steps or tasks as follows.

3.1. Input data processing and database updates

The only mandatory input required for the program is the merged reflection data in the form of an MTZ file and the amino-acid sequence of the target structure. These inputs are processed at the beginning of the pipeline to extract relevant information about the target. This includes details about the molecular weight of the target and the number of molecules that are likely to be in the asymmetric unit. The latter is based on the Matthews coefficient probabilities proposed recently by Kantardjieff & Rupp (2003).

The template-search step uses data from several databases, represented in the form of flat files. The files are provided by the EBI, apart from the SCOP file which comes from the MRC-LMB in Cambridge, and these files are updated regularly with data from newly solved structures. To take advantage of the latest data, the versions of these files provided with *MrBUMP* can be compared at runtime with the latest available, and new files downloaded if necessary.

3.2. Template search

Using the target sequence, a *FASTA* search is carried out using either the web-based OCA service at the EBI (Boutselakis *et al.*, 2003) or on the local machine if the *FASTA*34 program from the *FASTA* package (Pearson &

Lipman, 1988) is installed. If a local search is chosen, this is performed against a file of sequences generated from the coordinates of all files in the PDB. These sequences thus only include residues for which coordinates have been determined. Template-search models can also be specified explicitly, either *via* PDB codes or by uploading local PDB files. If no templates are specified explicitly, then the *FASTA* search is compulsory.

The search gathers a list of PDB chains that are potentially suitable as the basis for search models in MR. There is also the option to carry out a secondary-structure-based search using the EBI *SSM* web service (Krissinel & Henrick, 2004). The top-scoring structure from the *FASTA* search step is used as the query structure in the *SSM* search and any new structures found that were not found in the *FASTA* sequence-based search are added to the overall list of potential search models.

At this point in the program, a list of PDB chains has been garnered from the *FASTA* and *SSM* searches and any chains that have been inputted or specified by the user. The list may need to be pruned to reduce the computational overhead and the templates need to be ordered so that the search models more likely to produce a solution are used first in the MR step. To achieve this, the sequences of the templates are aligned and scored against the target sequence, with the alignment being carried out by either the *MAFFT* (Katoh *et al.*, 2005) or *ClustalW* (Chenna *et al.*, 2003) multiple alignment programs. This processing is carried out on the local machine and requires that one of these programs is installed.

3.3. Template scoring

For each template, the pairwise alignment against the target is extracted from the full multiple alignment and condensed to remove positions where there is a gap in both the target and template. The template is then scored according to the expression

$$\text{score} = \text{seq\_id} \times \frac{(L_{\text{align}} - P_0 \times \text{NGO}_{\text{temp}} - P_1 \times \text{NGE}_{\text{temp}})}{L_{\text{targ}}},$$

$$(1)$$

$$\text{seq\_id} = N_{\text{Cons}}/(L_{\text{align}} - \text{NG}_{\text{temp}}),\qquad(2)$$

where seq_id is the ungapped sequence identity between target and template sequences, L_{targ} is the length of the target sequence, L_{align} is the length of the target sequence included in alignment (usually equal to L_{targ}), N_{Cons} is the number of conserved residues in the alignment, $\text{NG}_{\text{temp}} = \text{NGO}_{\text{temp}} + \text{NGE}_{\text{temp}}$ is the total number of gap characters within the template sequence, NGO_{temp} is the number of gaps opened in template sequence by alignment, NGE_{temp} is the total number of gap extensions in the template sequence, P_0 is the gap-opening penalty (2.0) and P_1 is the gap-extension penalty (0.5).

The sequence identity (2) is calculated as the fraction of the number of target residues that are aligned against a template residue rather than a gap, *i.e.* it is the ungapped sequence identity. This measure is chosen because we are not interested

in the quality of the alignment *per se*, but rather what fraction of the residues in our search model have the correct side chain.

The second factor in (1) represents the extent of the alignment, *i.e.* how much of target is actually being modelled. In the case that $P_0 = P_1 = 1$, the second factor reduces to $(L_{align} - NG_{temp})/L_{targ}$, which is the fraction of target residues that are aligned against a template residue rather than a gap. However, such a measure does not distinguish between cases where a fraction of the target, say a large domain, aligns very well with the template and cases where there are many small insertions and deletions throughout the template. We believe that the former scenario will work better in MR and therefore penalize gap opening more than gap extension by setting $P_0 = 2.0$ and $P_1 = 0.5$. Thus, if a template is found for the first half of the target, then the second factor in (1) works out to be about 0.75, so that the score is less than the sequence identity but is reduced by less than a factor of two. Conversely, in the limit that the template aligns against every second target residue ($NGE_{temp} = 0$ and $NGO_{temp} = L_{align}/2$), the score reduces to zero.

The scoring system aims to take into account the factors which are relevant to MR and aims to be more sophisticated than simply looking at the sequence identity, but is nevertheless fairly *ad hoc*. We expect to have to refine this scheme, for example by adjusting the parameters P_0 and P_1. Note also that the score depends on the alignment used and may therefore be affected by the choice of alignment program.

After scoring all the templates, the highest scoring template structures go forward to the model-preparation and molecular-replacement steps. The number taken forwards defaults to ten, but can be altered by the user. The coordinate files are now downloaded from the PDB for the templates that have been retained. If a set of coordinates are derived from NMR data, then the first model from the NMR file is extracted for use as the search model. In future versions of the program it will be possible to use all of the models in an NMR PDB file as input search models for MR, if desired.

3.4. Domain and multimer search

The list of the highest scoring template structures is now augmented by any associated domains or multimers. The SCOP database (Murzin *et al.*, 1995; Lo Conte *et al.*, 2002) is searched for domains derived from the PDB codes in the template list. Domains which correspond to full chains are ignored, since they add nothing new. Otherwise, if a domain is found, its coordinates are extracted from the PDB file of the parent structure using the domain definition in the SCOP database and the domain is added to the list of search-model templates to be used in molecular replacement.

Similarly, the PQS database (Henrick & Thornton, 1998) is used to identify possible multimeric structures based on the template structure. Putative multimers are checked against the target asymmetric unit to see if they are appropriate as search models. If a multimer is identified, its coordinate file is retrieved from the PQS online database of multimers and added to the list of models to be used in molecular replace-ment. The constituent chains of the multimer are edited according to the *CHAINSAW* protocol (see below).

3.5. Model preparation

For each template chain, up to four search models are generated by applying different editing procedures. The simplest procedure is the 'PDBclip' method, which retains most of the template coordinates. The input PDB file is tidied by the removal of water molecules and H atoms, the selection of the most probable conformations for side chains and the correction of any anomalies in the PDB file such as missing chain names. These steps are performed with the *CCP*4 programs *Pdbset*, *Coord_format* and *Pdbcur* (Collaborative Computational Project, Number 4, 1994). The PDBclip procedure is also a precursor to the other search model-generation methods.

The second method is the 'polyALA' procedure, which truncates all side chains in the template PDB file to at most the β position. Such polyalanine models are quite commonly used in MR. Both the PDBclip and polyALA procedures retain the full main chain of the template molecule and no attempt is made to align this with the target sequence. Consequently, insertions in the template molecule relative to the target are retained in the search model.

The third method is based on the model-improvement functions of *MOLREP* (Vagin & Teplyakov, 1997). The template structure and the target sequence are passed to *MOLREP*, which generates an alignment and edits the template structure accordingly. This alignment takes into account the three-dimensional structure of the template model. Gaps and insertions are considered to be unlikely within sequence fragments that correspond to helices and strands and are considered to be more probable in the loops and at the surface of the template model. Parts of the template structure which do not align with the target are removed. For those parts which do align, the side chain is retained if the residue is conserved and is truncated to a common part if it is not conserved.

The final method is based on the *CCP*4 program *CHAINSAW* (Stein, 2007). This edits the template structure in a similar way to *MOLREP*, but differs in two respects. Firstly, it uses an externally provided alignment between the template and the target. In the general case, this allows the user full control over the alignment. In the context of *MrBUMP*, the pairwise alignment between target and template is extracted directly from the multiple alignment performed previously (see §3.2). Secondly, the side-chain pruning is more severe for nonconserved residues, with the side chain being truncated to at most the γ position. *CHAINSAW* implements the 'mixed' model of Schwarzenbacher *et al.* (2004).

In a loose sense, the *MOLREP* and *CHAINSAW* methods fill the gap between the PDBclip and polyALA methods. The PDBclip method should do well when the sequence identity between the target and the template is very high and one wants to include most of the atoms in the template structure. When the sequence identity is high, the *MOLREP* and

CHAINSAW methods will in fact produce a PDBclip-like model, since most residues are aligned and conserved. Conversely, the polyALA method applies when the sequence identity is low and at most the overall fold is conserved. In such cases, the *MOLREP* and *CHAINSAW* methods will truncate most of the side chains since most aligned residues are not conserved. Therefore, the PDBclip and polyALA methods are possibly redundant, but are included as they are commonly used methods for preparing search models for molecular replacement.

3.6. Molecular replacement and refinement

For each of the prepared search models in the sorted list, an MR job is initiated. The ordering of the jobs depends on the sorted order of the search-model list. In the case where the search involves several molecules in the asymmetric unit, any multimeric search models will be used first as this will reduce the search time in MR. Either *MOLREP* (Vagin & Teplyakov, 1997) or *Phaser* (McCoy *et al.*, 2005) can be used to perform the molecular replacement. If an MR solution is found, then the resulting positioned model is put through restrained refinement in *REFMAC* (Murshudov *et al.*, 1997) to see how well it refines. The MR solution may contain fewer copies of the search model than predicted.

As a general rule, we aim to use as many program defaults as possible, thereby allowing improvements in the underlying programs to take effect immediately. For *Phaser*, we use the automated mode MR_AUTO. Parameters such as the reflection-file column labels, the molecular weight of the target and the estimated sequence identity between the model and the target are generated automatically. The number of molecules to search for is estimated automatically, but can be overridden by the user, as can the number of packing clashes allowed. For *MOLREP*, we again supply the reflection-file column labels, the estimated sequence identity and the number of molecules to search for, but otherwise use the defaults.

Both MR programs use the sequence identity, supplied by *MrBUMP*, to effectively downweight the high-resolution terms. Similarly, the model completeness is used to downweight the low-resolution terms. Finally, both programs by default apply a suitable high-resolution cutoff. No explicit resolution limits on the reflection data are imposed by *MrBUMP*.

For *REFMAC*, we find that 30 cycles of restrained refinement may be needed to produce a model which can be used as a starting point in *ARP/wARP* (Bahar *et al.*, 2006). We use the WEIGHT AUTO option of *REFMAC* to ensure reasonable weighting between the X-ray and geometry terms. Otherwise, we use the program defaults. Since the MR model contains only peptide residues, the initial refinement should proceed without difficulties.

The success criterion for each search model is based on the behaviour of the R_{free} value during the restrained refinement. If the value drops by 20% of the initial value during restrained refinement and the final value is less than 0.5, we judge this to

be a likely success. With a very good search model, R_{free} may start from a good value and stay constant or even become slightly worse. In such cases, we rely on the absolute value of R_{free} and judge the model a success if the final R_{free} is less than 0.35. Otherwise, if the value drops by 5% of the initial value during restrained refinement and the final value is less than 0.52, or the final value is less than 0.48, then we judge this to be a marginal success. Both successes and marginal successes are flagged by the program and the program will optionally finish if a success is found. Alternatively, the user can request that all of the search models are tested in molecular replacement.

The chosen success criterion is meant as a rough guide. The user can and should manually inspect search models labelled as a likely or marginal success. In particular, the unrefined model is available as well as the refined one. We find that a partially refined model works better as a starting point for rebuilding in *ARP/wARP* (Bahar *et al.*, 2006), whereas an unrefined model may be better for identifying model bias.

3.7. User interfaces

The pipeline is implemented as a Python script which can be accessed in a number of ways. *MrBUMP* can be run from the command line with keyworded input, in a similar manner to other *CCP4* programs, or can be run *via* a *ccp4i* task interface. *MrBUMP* has also been implemented as a web service and is used as such within the e-HTPX project (Allan *et al.*, 2005). In all cases, *MrBUMP* can be run with minimal direction from the user, as is the desire for modern automation schemes. The output consists of a summary log file, a positioned and
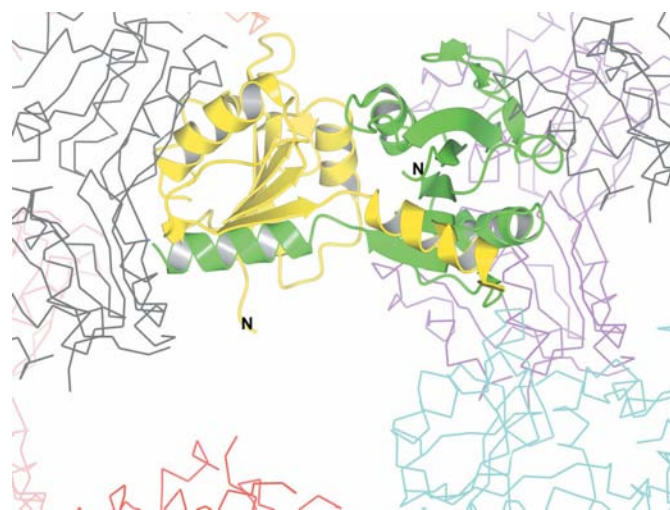

Figure 2
Illustration of the packing of model 1xcb_f2_MOLREP after MR using *Phaser* (see §4.1). In the centre are the two copies located by *Phaser* in the asymmetric unit. The domain-swapped helices are clearly visible. The surrounding C^α traces represent symmetry-related molecules. Large gaps in the crystal packing indicate where the N-terminal domains were subsequently placed. The first domain of the dimer gave a *Phaser* solution with RFZ = 4. 6 and TFZ = 3.5 and the second RFZ = 4.3 and TFZ = 5.4. The partial structure refined to $R_{\mathrm{free}} = 0.516$. Despite the poor statistics, the solution is correct and was sufficient to proceed to the full model. This figure was prepared using *CCP4mg* (Potterton *et al.*, 2004)

Table 1
Selected results for transcription factor from *B. subtilis* (§4.1).

The first column gives the search model used, following the notation described in §4. The column 'Seq. id.' gives the appropriate sequence identity against the target for the chain or domain, as determined from the multiple alignment step. The column 'R.m.s.d.$_{model}$' gives the r.m.s.d. of C$^\alpha$ atoms between the search model and the full chain *A* of 1xcb, after superposition with *SUPERPOSE* (Krissinel & Henrick, 2004). The number in parentheses is the number of C$^\alpha$ atoms included in the r.m.s.d. calculation. The column 'RFZ/TFZ' gives the *Z* scores from the *Phaser* rotation and translation functions for the second copy located. The column 'R_{free}(i)/R_{free}(f)' gives the initial and final R_{free} values from restrained refinement in *REFMAC*. The correctness of the solution indicated in the final column is based on correct packing of domain-swapped helix, as well as comparison with the final structure.

Search model	Seq id. (%)	R.m.s.d.$_{model}$ (Å)	RFZ/ TFZ	R_{free}(i)/R_{free}(f)	Solution
1xcb_f2_MOLREP	35.0	0.667 (125)	4.3/5.4	0.523/0.516	Yes
1xcb_f2_CHNSAW		0.670 (125)	5.1/6.2	0.531/0.538	Yes
1xcb_f1_MOLREP	36.2	0.828 (69)	4.1/5.6	0.546/0.549	No
1xcb_f1_CHNSAW		0.828 (69)	4.0/4.6	0.542/0.544	No
1xcb_F_MOLREP	29.9	1.340 (197)	3.6/4.6	0.540/0.556	No
1xcb_F_CHNSAW		1.072 (86)	4.6/5.8	0.559/0.544	No
1xcb_d2_MOLREP	35.0	0.663 (125)	4.8/4.6	0.535/0.526	No
1xcb_d2_CHNSAW		0.664 (125)	3.5/4.9	0.512/0.534	No

partially refined model and a subdirectory containing all intermediate results available for inspection.

4. Examples

At a workshop held in July 2005, *MrBUMP* (alongside several other programs and automation schemes) was tested against a set of 17 targets from the Structural Proteomics in Europe (SPINE) project. The results are presented in a separate publication (Bahar *et al.*, 2006). *MrBUMP* found solutions in the majority of cases, although many of these had good homologues available in the PDB and the advantage of using *MrBUMP* was simply one of convenience.

Here, we describe in detail three different examples, two of which come from early users of *MrBUMP*. These are more advanced examples than those described in Bahar *et al.* (2006). It should be stressed that we aim to illustrate the usage and advantages of *MrBUMP*, rather than give a detailed comparison of the various methods and programs involved. These are individual case studies and no general conclusions can be
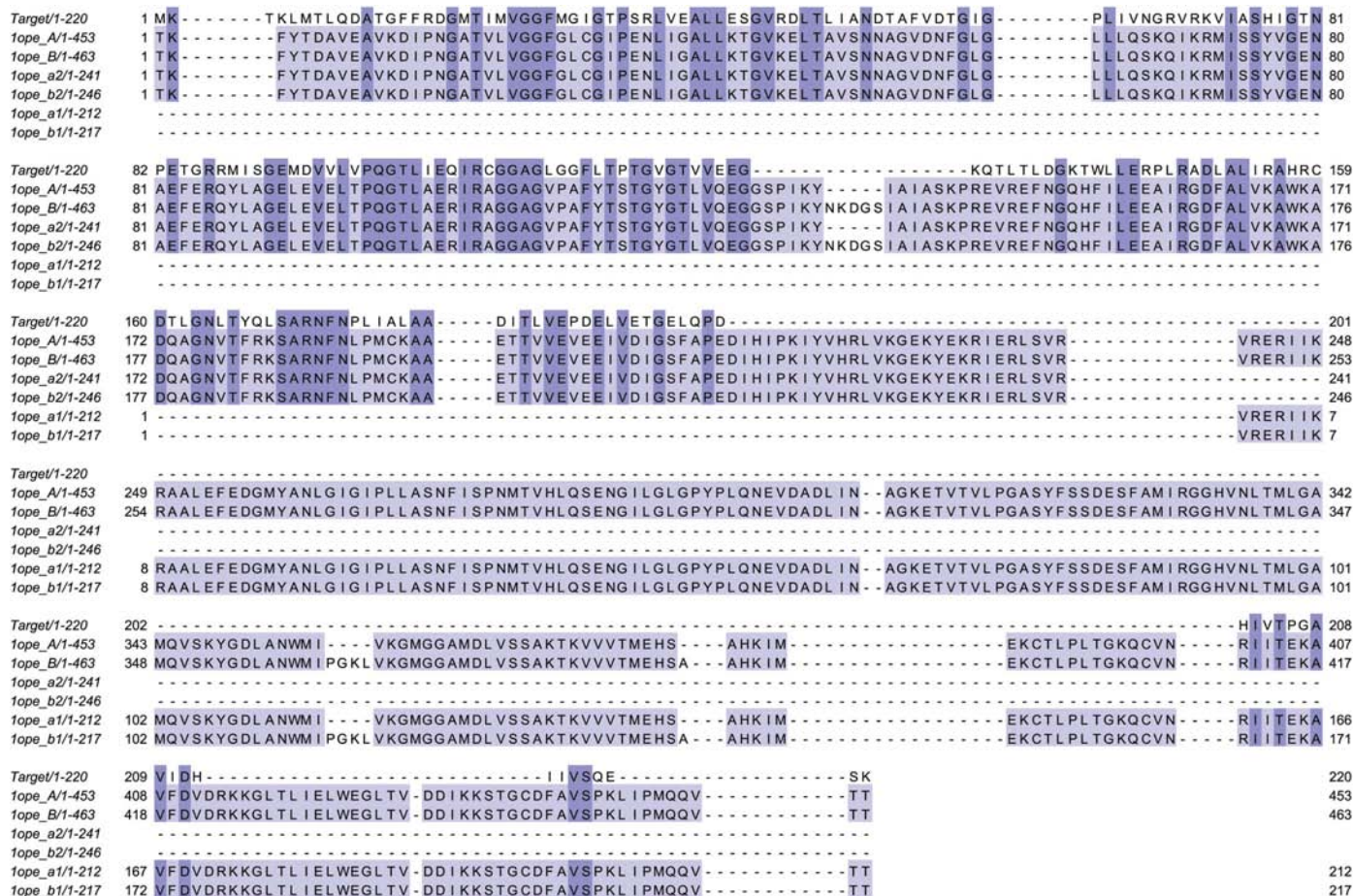

Figure 3
Multiple alignment results for the α subunit of acetate CoA-transferase (1k6d; §4.2). The multiple alignment included 54 template chains and domains; for clarity, only the six sequences from 1ope are shown here. Regions where only gaps are visible are aligned against residues of sequences that are not displayed. The figure shows that 201 of the 220 target residues are aligned against the N-terminal domains (1ope_a2 and 1ope_b2), while 19 are incorrectly aligned against the C-terminal domain. The alignment was produced with *MAFFT* (Katoh *et al.*, 2005) and the figure was produced with *Jalview* (Clamp *et al.*, 2004).

drawn from the specific results. A more systematic comparison will be the subject of a future study.

In testing *MrBUMP* in the context of automation, we may classify a particular model as a failure if it does not give a refinable positioned model in the output of *MrBUMP*. That is not to say that a solution could not be obtained by further inspection of the *MrBUMP* result and in fact we see such lead generation as an important application of *MrBUMP*. In the examples given here, final models have been submitted to the PDB and we can compare the results of *MrBUMP* directly against these models.

In the following, we refer to specific search models using the nomenclature <PDB_code>_<subunit_ID>_<model_preparation_method>, where a subunit can be a chain or a domain. For example, 1jza_B_CHNSAW refers to chain *B* of 1jza prepared using the program *CHAINSAW*. Note also that the sequence identity of a model to the target depends on the specific sequence alignment and may therefore vary between different stages of the pipeline as different alignments are generated. It also depends on the definition of sequence identity used, *e.g.* gapped *versus* ungapped.

4.1. Transcription factor from *Bacillus subtilis*

The first example is a transcription factor from *B. subtilis* with an N-terminal DNA-binding domain of around 85 residues and a C-terminal ligand-binding domain of around 125 residues (Logan, 2007). There are two copies in the asymmetric unit and data were collected to 2.1 Å.

The initial *FASTA* search found only one PDB entry, 1xcb, a Rex-family transcriptional repressor from *Thermus aquaticus* (Sickmier *et al.*, 2005) consisting of seven chains in the asymmetric unit (3.5 dimers) with approximately 36% sequence identity to the target. SCOP has entries for N-terminal and C-terminal domains (referred to in the following as a1 and a2, respectively, for chain *A* and so on). *MrBUMP* constructed models based on the 14 individual domains and seven monomers and processed with the *MOLREP* and *CHAINSAW* model-preparation methods.

Of the 42 models tried, only those based on the C-terminal domain of chain *F* of 1xcb (with either the *MOLREP* or *CHAINSAW* model-preparation methods) or that based on the C-terminal domain of chain *B* (with the *CHAINSAW* model only) produced solutions (see Table 1). In fact, the final R_{free} from *REFMAC*5 was only 0.516, 0.538 and 0.540, respectively, for the three models and these were not identified as solutions by *MrBUMP*. Nevertheless, the R_{free} of 0.516 for model 1xcb_f2_MOLREP was clearly better than other solutions and prompted closer investigation.

The C-terminal domains of 1xcb include a helix which binds to the C-terminal domain of the other member of the dimer and such domain-swapped helices are also expected in the target structure. This is indeed the case, although the configuration of the helices is found to be slightly altered in the final structure. The presence or absence of domain-swapped helices thus becomes a convenient diagnostic when inspecting the MR solutions and many solutions can be rejected on this basis.

Conversely, for the three successful models, the packing of the MR solution clearly shows this feature (see Fig. 2).

The final solution was based on model 1xcb_f2_MOLREP using *Phaser* as the MR engine. Having used *MrBUMP* to locate all C-terminal domains, a manual MR job was used to fit the N-teminal domains. While the position of the domain-swapped helices clearly indicated correct packing of the C-terminal domains, there appeared to be some rearrangement in the target compared with the search model. Therefore, these helices were removed and rebuilt after refining the rest of the structure.

This turned out to be a difficult MR problem owing to variations between the chains of 1xcb and owing to the rearrangement of the domain-swapped helix between 1xcb and the target. Table 1 lists the r.m.s.d. of the models compared with chain *A* of 1xcb (chosen arbitrarily as a reference structure) and shows that there are reasonably substantial differences between the search models. The larger r.m.s.d. for the full chain is indicative of a slight variation of the inter-domain angle between the different chains of 1xcb. The resolution of the search template 1xcb is only 2.9 Å, which may partially explain the structural variations between the chains.

The usefulness of *MrBUMP* in this case was the ability to process a large number of potential models. Although *MrBUMP* was not able to identify the correct solution automatically, it provides enough information so that further manual processing is straightforward. Given that only three out of 42 models appeared to produce a solution, it is possible that a manual search would have missed these.

4.2. Acetate CoA-transferase from *Escherichia coli*

PDB entry 1k6d is the α subunit of acetate CoA-transferase (ACT) from *E. coli*, with an asymmetric unit containing two chains of 220 residues. It was solved originally using native data to 1.9 Å and phases from a selenomethionine-labelled protein to 3.4 Å (Korolev *et al.*, 2002).

For *MrBUMP*, we start with a target sequence of 220 residues for the α subunit (Swiss-Prot entry P76458). The *FASTA* search picks up templates based on 1ooz, 1ooy, 1ope, 1o9l and 1m3e, with sequence identities of around 37%. These are porcine succinyl-CoA:3-ketoacid CoA transferases (SCOT), with longer chains collinear with the bacterial α and β subunits. The other member of the 'CoA transferase α-subunit-like' SCOP family, 1poi, has a lower sequence identity of 26% and is further down the *FASTA* list, below unrelated proteins. Only the 1poi structure was known at the time of the original structure determination.

For the porcine SCOT templates, the SCOP search returns both α-subunit-like (N-terminal domain, denoted a2) and the β-subunit-like (C-terminal domain, denoted a1) domains. The former is structurally related to the target and is expected to form a good search model. The latter is more distantly related and is not expected to form a good search model. The model-preparation methods which use an alignment to the target, *viz.* *CHAINSAW* and *MOLREP*, are expected to select the correct domain from the full-chain template and form good

search models, whereas the other methods, *viz*. PDBclip and polyALA, will produce models which are too large.

MrBUMP scores the porcine SCOT search models with the α-subunit-like domain higher than the β-subunit-like domain, because of the higher sequence identity. The scoring of the full chain depends on the quality of the alignment against the target, which in turn is sensitive to the number of sequences included in the multiple sequence alignment. With a sufficiently large number of sequences included, the alignment of the target to the N-domain region of the full chain is good (Fig. 3) and the full chain obtains a similar score to the α-subunit-like domain. Consequently, the two search models generated by *CHAINSAW* are very similar. With a smaller number of sequences included, the alignment is poorer, the ranking of the full chain is lower and the *CHAINSAW* search model contains fragments from both domains and performs poorly. For comparison, the alignment generated by *MOLREP* for the full chain aligns 189 of the 220 target residues to the N-domain and therefore generates a reasonable search model.

Sample results are given in Table 2. Models 1ope_a2_CHNSAW and 1ope_a2_MOLREP both give good solutions which refine well. 1ope_a2_PLYALA also gives a solution which refines, although not so convincingly since all side chains are missing, while 1ope_a2_PDBCLP fails to find a solution. Models 1ope_A_CHNSAW and 1ope_A_MOLREP derived from the full-chain template also give solutions, though not so convincing as those based on the N-domain. Finally, models 1ope_a1_CHNSAW and 1ope_a1_MOLREP fail to give solutions, as expected. Results from other porcine SCOT search models are similar.

In conclusion, *MrBUMP* solves this structure straightforwardly using porcine SCOT structures that have become available since the original structure determination. The structure solution is most robust with search models based on the α-subunit-like N-domain, which are generated automatically *via* the SCOP procedure of *MrBUMP*. Nevertheless, *MrBUMP* will also solve the structure from full-chain templates, using alignment to the target to generate the correct search model, as implemented in *CHAINSAW* and *MOLREP*.

4.3. LqhIT2

The final example is *Leiurus quinquestriatus hebraeus* insect toxin 2 (LqhIT2), which is one of about 200 short toxic proteins produced by an Israeli yellow scorpion and belongs to the so-called 'depressant' type of toxin (Karbat *et al.*, 2007). Data had been collected to 1.2 Å and there is one molecule in the asymmetric unit. Despite the existence of three structures at around 58% sequence identity, namely 1jza, 1jzb and 2sn3, the target had resisted solution since 2002.

Sample results from a standard run of *MrBUMP* are given in Table 3. *MrBUMP* identifies two marginal solutions, one based on model 1jzb_A_MOLREP using *Phaser* as the MR program and the other based on model 1jza_A_MOLREP, again using *Phaser* as the MR program (the first and third

Table 2
Selected results from 1k6d (§4.2).

The column 'Contrast' gives the contrast output by *MOLREP*, while column 'CC$_1$/CC$_2$' gives the correlation coefficients of the first and second peaks of the translation function for the second copy located. Other columns are as in Table 1.

Search model	Seq id. (%)	Contrast	CC$_1$/CC$_2$	R_{free}(i)/ R_{free}(f)	Solution
1ope_a2_CHNSAW	40.2	5.84	0.464/0.410	0.533/0.396	Yes
1ope_a2_MOLREP		3.10	0.496/0.455	0.528/0.391	Yes
1ope_a2_PDBCLP		1.33	0.424/0.421	0.538/0.537	No
1ope_a2_POLYALA		7.09	0.420/0.370	0.548/0.452	Yes
1ope_A_CHNSAW	39.7	5.73	0.439/0.385	0.540/0.451	Yes
1ope_A_MOLREP		6.16	0.449/0.398	0.538/0.442	Yes
1ope_a1_CHNSAW	35.0	No solution			No
1ope_a1_MOLREP		1.30	0.378/0.375	0.565/0.579	No

Table 3
Selected results for target LqhIT2 (§4.3; since deposited with PDB code 2i61).

Templates 1jza and 1jzb have a sequence identity of 58% to the target, while template 2sn3 has a sequence identity of 57%. The column 'R.m.s.d._model' gives the r.m.s.d. of C$^\alpha$ atoms between the search model and the final model after superposition with *LSQKAB* from *CCP*4. The column 'RFZ/TFZ' gives the Z scores from the *Phaser* rotation and translation functions for the top solution. The column 'R_{free}(i)/R_{free}(f)' gives the initial and final R_{free} values from restrained refinement in *REFMAC*. The column 'r.m.s.d._sol' gives the r.m.s.d. of C$^\alpha$ atoms between the partially refined model and the final model. The program *REFORIGIN* from *CCP*4 is used to move the partially refined model to the nearest symmetry equivalent, but the r.m.s.d. is calculated without superposing the models. The column 'ARP/wARP' gives the number of residues rebuilt by *ARP/wARP* (Perrakis *et al.*, 1999) and the number in parentheses is the number of side chains docked.

Search model	R.m.s.d._model (Å)	RFZ/ TFZ	R_{free}(i)/ R_{free}(f)	R.m.s.d._sol (Å)	ARP/ wARP
1jzb_A_MOLREP	2.522	3.6/3.2	0.537/0.491	2.621	60 (60)
1jzb_A_CHAINSAW	2.497	4.0/3.3	0.542/0.536	No match	0 (0)
1jza_A_MOLREP	2.563	4.6/2.9	0.545/0.498	2.479	58 (58)
1jza_A_CHAINSAW	2.546	4.3/2.8	0.560/0.547	No match	0 (0)
1jza_B_MOLREP	2.942	4.6/2.4	0.529/0.503	2.759	60 (60)
1jza_B_CHAINSAW	2.543	2.9/3.2	0.538/0.529	3.135	58 (58)
2sn3_A_MOLREP	3.229	3.8/2.7	0.522/0.548	No match	0 (0)
2sn3_A_CHAINSAW	2.459	4.1/3.8	0.539/0.535	No match	0 (0)

models in Table 3). Both solutions were rebuilt using *ARP/wARP* (Perrakis *et al.*, 1999), which built and docked 60 residues of the former and 58 residues of the latter out of 61 residues in the target protein. Inspection of the rebuilt models in *Coot* (Emsley & Cowtan, 2004) confirms the structure to be essentially solved.

More detailed inspection of the results from *MrBUMP* shows that other models also give solutions, despite *MrBUMP* not identifying them as such. In fact, the structure was initially solved using model 1jza_B_MOLREP and using *MOLREP* as the MR program. Although this is not classified as a solution or a marginal solution, *ARP/wARP* will build and dock 60 residues. Conversely, other solutions, such as that from *Phaser* using search model 1jzb_A_CHAINSAW, fail to rebuild in *ARP/wARP*.

The second column of Table 3 shows the r.m.s.d. of C$^\alpha$ atoms between the search model and the final refined structure after superposition with the program *LSQKAB* from the *CCP*4 suite (Collaborative Computational Project, Number 4, 1994)

and it is clear that the structural differences between the search models and the target are greater than might be expected for a sequence identity of 58%. For comparison, the study of Lesk & Chothia (1980) on globins suggests that an r.m.s.d. of 3.0 Å corresponds to a sequence identity of 25–30%. Fig. 4 illustrates the structural differences between two representative models and the final refined structure. There is reasonable agreement in the conserved α/β core of the protein, but many small changes across the rest of the protein.

The Z scores from *Phaser* are in general poor and do not really indicate a success. Nevertheless, some of the solutions from *Phaser* do refine in *REFMAC*. Manual comparison of the partially refined structures against the final model (column r.m.s.d.$_{sol}$ in Table 3) reveals that half of the models shown in Table 3 are essentially correct and this is confirmed by successful rebuilding in *ARP/wARP*. In the case of 1jza_A_CHNSAW, the correct MR solution is actually the fourth in the list of solutions output by *Phaser*, but in the other failed cases the correct MR solution does not appear at all.

Whether or not a solution is obtained is thus sensitive to the details of the search model used because, as a group, the search models are borderline in their structural similarity to the target. On the other hand, because of the good quality of the data, *ARP/wARP* is able to build a good model given an approximately correct MR solution. For LqhIT2, structure solution also appears to be sensitive to the details of the MR step. In earlier manual attempts to solve this structure, a high-resolution cutoff of 3 Å had been applied and no solution was found. It seems that for this case at least, inclusion of high-resolution terms (albeit suitably downweighted) is critical.

5. Discussion and future directions

The philosophy of *MrBUMP* is to perform an exhaustive search over a large number of possible search models. We do not attempt to determine the single best protocol, as this is likely to be problem-dependent. In simple cases, this approach would be inefficient, so we have included the option to streamline the search in such cases. In more difficult cases, the exhaustive search does sometimes find the particular combination which is required for a solution. *MrBUMP* has already proved useful in a number of recent structure solutions (Obiero *et al.*, 2006; El Omari *et al.*, 2006; Karbat *et al.*, 2007; Logan, 2007). Nevertheless, there are potential improvements to many of the steps in the *MrBUMP* pipeline.

The initial *FASTA* search could be supplemented by profile-based searches, as implemented for example in *psiBLAST* (Altschul *et al.*, 1997) and *FFAS* (Jaroszewski *et al.*, 2005). In general, these would be expected to discover more remote homologues. Although it is debatable whether such remote homologues would be structurally similar enough to work as templates for MR search models, the possibility remains that in particular cases this approach could be the difference between finding a solution and not.

As discussed by Schwarzenbacher *et al.* (2004), the alignment between the template structure and the target may be crucial in finding a solution. For the *CHAINSAW*-based

search models, we derive this alignment from a multiple alignment against all template sequences. This step could be improved by embedding the alignment in a larger set of sequences, including homologues for which structures are not known, or by making use of structural information. Alternatively, the alignment could be imported from external sources, such as a run of *FFAS* or a user's own knowledge of a protein family.

Our use of the *SSM* search is one simple way of accessing families of related structures (which, since it is based on the top *FASTA* hit rather than the target, may or may not be relevant). Clearly, in order to generate a full list of putative search models, one should use any knowledge one has on the family of proteins that the target belongs to. We use the domain definitions held in the SCOP database, but do not use the SCOP hierarchy itself. There are obviously many other sources of such information, for example CATH (Orengo *et al.*, 1997) or Pfam (Finn *et al.*, 2006). In addition, our use of the PQS database (Henrick & Thornton, 1998) will shortly be replaced by the more recent PISA service (Krissinel & Henrick, 2005).

It has been long appreciated that a target structure may differ from a search model by a relatively large but simple configuration change, such as a hinge bend. A simplified normal-modes calculation using the Gaussian Network Model (Tirion, 1996; Haliloglu *et al.*, 1997) can suggest potential configuration changes which can be applied to the search model before molecular replacement. The generation of perturbed search models for subsequent use in MR has been
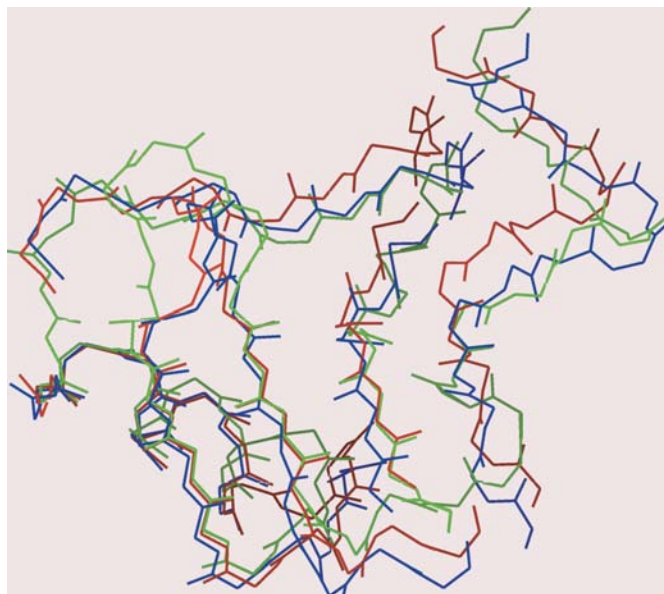

Figure 4
Two repesentative models of LqhIT2 (§4.3) compared with the final structure. The main chains of models 1jzb_A_MOLREP (blue) and 2sn3_A_MOLREP (red) are compared with the final model (green) after superposition with *SUPERPOSE* (Krissinel & Henrick, 2004). The conserved α/β core of the protein is towards the bottom left of the figure. *MrBUMP* finds a solution with 1jzb_A_MOLREP (r.m.s.d.$_{model}$ = 2.522 Å) but not with 2sn3_A_MOLREP (RMSD$_{model}$ = 3.229 Å). This figure was prepared using *CCP4mg* (Potterton *et al.*, 2004).

implemented in *El Nemo* (Suhre & Sanejouand, 2004*a*,*b*) and *Phaser* (McCoy *et al.*, 2005).

In conclusion, our test cases and the examples above demonstrate the utility of trying a range of search models, a protocol that can only be attempted adequately by automation. *MrBUMP* provides an implementation of such a potocol. For any particular template model, we would not expect *MrBUMP* to compete with careful analysis of the data and the model by an experienced crystallographer. However, it may succeed in difficult cases by finding a combination of models and protocols that would not otherwise have been tried. In more straightforward cases, the advantage is of course simply one of convenience.

6. Availability

MrBUMP is distributed under the *CCP*4 licence and is available for download from http://www.ccp4.ac.uk/MrBUMP. It runs under Linux/Unix, Mac OSX and Windows and comes complete with a *ccp*4*i* GUI.

References

Allan, R. J., Nave, C., Keegan, R. M., Meredith, D. J., Winn, M. D., Winter, G., Dolomanov, O., Launer, L., Young, P. & Berry, I. (2005). In *Proceedings of the UK e-Science All Hands Meeting.* http://www.allhands.org.uk/2005/proceedings/.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997). *Nucleic Acids Res.* **25**, 3389–3402.

Bahar, M. *et al.* (2006). *Acta Cryst.* D**62**, 1170–1183.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). *Nucleic Acids Res.* **28**, 235–242.

Boutselakis, H. *et al.* (2003). *Nucleic Acids Res.* **31**, 458–462.

Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T. J., Higgins, D. G. & Thompson, J. D. (2003). *Nucleic Acids Res.* **31**, 3497–3500.

Clamp, M., Cuff, J., Searle, S. M. & Barton, G. J. (2004). *Bioinformatics*, **20**, 426–427.

Claude, J.-B., Suhre, K., Notredame, C., Claverie, J.-M. & Abergel, C. (2004). *Nucleic Acids Res.* **32**, W606–W609.

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

El Omari, K., Dhaliwal, B., Lockyer, M., Charles, I., Hawkins, A. R. & Stammers, D. K. (2006). *Acta Cryst.* F**62**, 949–953.

Emsley, P. & Cowtan, K. (2004). *Acta Cryst.* D**60**, 2126–2132.

Finn, R. D., Mistry, J., Schuster-Böckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S. R., Sonnhammer, E. L. L. & Bateman, A. (2006). *Nucleic Acids Res.* **34**, D247–D251.

Fu, Z.-Q., Rose, J. & Wang, B.-C. (2005). *Acta Cryst.* D**61**, 951–959.

Giorgetti, A., Raimondo, D., Miele, A. E. & Tramontano, A. (2005). *Bioinformatics*, **21**, 72–76.

Haliloglu, T., Bahar, I. & Erman, B. (1997). *Phys. Rev. Lett.* **79**, 3090–3093.

Henrick, K. & Thornton, J. M. (1998). *Trends Biochem. Sci.* **23**, 358–361.

Jaroszewski, L., Rychlewski, L., Li, Z., Li, W. & Godzik, A. (2005). *Nucleic Acids Res.* **33**, W284–W288.

Jaskólski, M., Li, M., Laco, G., Gustchina, A. & Wlodawer, A. (2006). *Acta Cryst.* D**62**, 208–215.

Kantardjieff, K. A. & Rupp, B. (2003). *Protein Sci.* **12**, 1865–1871.

Karbat, I., Turkov, M., Cohen, L., Kahn, R., Gordon, D., Gurevitz, M. & Frolow, F. (2007). *J. Mol. Biol.* **366**, 586–601.

Katoh, K., Kuma, K., Toh, H. & Miyata, T. (2005). *Nucleic Acids Res.* **33**, 511–518.

Korolev, S., Koroleva, O., Petterson, K., Gu, M., Collart, F., Dementieva, I. & Joachimiak, A. (2002). *Acta Cryst.* D**58**, 2116–2121.

Krissinel, E. & Henrick, K. (2004). *Acta Cryst.* D**60**, 2256–2268.

Krissinel, E. & Henrick, K. (2005). *CompLife 2005*, edited by M. R. Berthold, R. Glen, K. Diederichs, O. Kohlbacher & I. Fischer, pp. 163–174. Berlin/Heidelberg: Springer-Verlag.

Lesk, A. M. & Chothia, C. (1980). *J. Mol. Biol.* **136**, 225–230.

Lo Conte, L., Brenner, S. E., Hubbard, T. J. P., Chothia, C. & Murzin, A. G. (2002). *Nucleic Acids Res.* **30**, 264–267.

Logan, D. (2007). Submitted.

Long, F., Vagin, A. A. & Murshudov, G. N. (2007). In preparation.

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

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

Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. (1995). *J. Mol. Biol.* **247**, 536–540.

Obiero, J., Bonderoff, S. A., Goertzen, M. M. & Sanders, D. A. R. (2006). *Acta Cryst.* F**62**, 757–760.

Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B. & Thornton, J. M. (1997). *Structure*, **5**, 1093–1108.

Panjikar, S., Parthasarathy, V., Lamzin, V. S., Weiss, M. S. & Tucker, P. A. (2005). *Acta Cryst.* D**61**, 449–457.

Pearson, W. R. & Lipman, D. J. (1988). *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.

Perrakis, A., Morris, R. & Lamzin, V. S. (1999). *Nature Struct. Biol.* **6**, 458–463.

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

Reddy, V., Swanson, S. M., Segelke, B. W., Kantardjieff, K. A., Sacchettini, J. C. & Rupp, B. (2003). *Acta Cryst.* D**59**, 2200–2210.

Rupp, B., Segelke, B. W., Krupka, H. I., Lekin, T. P., Schäfer, J., Zemla, A., Toppani, D., Snell, G. & Earnest, T. (2002). *Acta Cryst.* D**58**, 1514–1518.

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

Sickmier, E. A., Brekasis, D., Paranawithana, S., Bonanno, J. B., Paget, M. S., Burley, S. K. & Kielkopf, C. L. (2005). *Structure*, **13**, 43–54.

Stein, N. D. (2007). In preparation.

Strokopytov, B. V., Federov, A., Mahoney, N. M., Kessels, M., Drubin, D. G. & Almo, S. C. (2005). *Acta Cryst.* D**61**, 285–293.

Suhre, K. & Sanejouand, Y.-H. (2004*a*). *Acta Cryst.* D**60**, 796–799.

Suhre, K. & Sanejouand, Y.-H. (2004*b*). *Nucleic Acids Res.* **32**, W610–W614.

Tirion, M. M. (1996). *Phys. Rev. Lett.* **77**, 1905–1908.

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