Acta Crystallographica Section D
**Biological Crystallography**
ISSN 0907-4449

# *MrBUMP*: an automated pipeline for molecular replacement

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

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

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

Received 9 May 2007
Accepted 30 July 2007

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. An overview of *MrBUMP* is given and some recent additions to its functionality are highlighted.

## 1. *MrBUMP* as an automated pipeline

The 2007 CCP4 Study Weekend meeting has addressed many aspects of molecular replacement (MR), including how to prepare a search model and how to assess and process the output of MR programs. It is pertinent to ask what automation can do in this context.

An automation pipeline cannot do anything that could not in principle be done manually. In the context of MR, if there really is no suitable homologous protein to form the basis of a search model, then an automation pipeline will not create one. Does automation add anything at all? In fact, there are often a large number of potential search models to be tried and automation can clearly help in ensuring that all are tried. 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.

Earlier, Schwarzenbacher *et al.* (2004) also advocated the use of a range of search models together with the use of more than one MR program and suggested that

> The only practical solution for massive MR searches with different parameters is automation and parallelization.

An additional and important benefit of an automated scheme is that one gets data and file management for free.

Going to the other extreme, can automation now do everything, so that the practising protein crystallographer no longer need worry about the methodology? Sadly, this is not the case either. As we have heard during the meeting, difficult cases are still common and an automation scheme will not cover all the tricks needed to solve such cases. Moreover, the parameters and methods used in an automation scheme will be tuned to the test sets used and however extensive these test

doi:10.1107/S0907444907037195

sets are, the resulting parameters will not be appropriate in every case, even given searches over some parameters. Finally, finding a correct MR solution is not the end of the story: it is still necessary to complete the model and it may be necessary to rebuild parts of the model where there is bias towards the search model.

In this article, we describe *MrBUMP*, an MR automation scheme that we have been developing. 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; see, for example, Rupp *et al.* (2002) and Fu *et al.* (2005). Publicly 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, and a scheme for using comparative models in MR (Giorgetti *et al.*, 2005). More recently, the *Balbes* automated system for MR has been developed at York. *Balbes* is described elsewhere in this issue (Long *et al.*, 2008), along with the MR pipeline of the Joint Centre for Structural Genomics (JCSG).

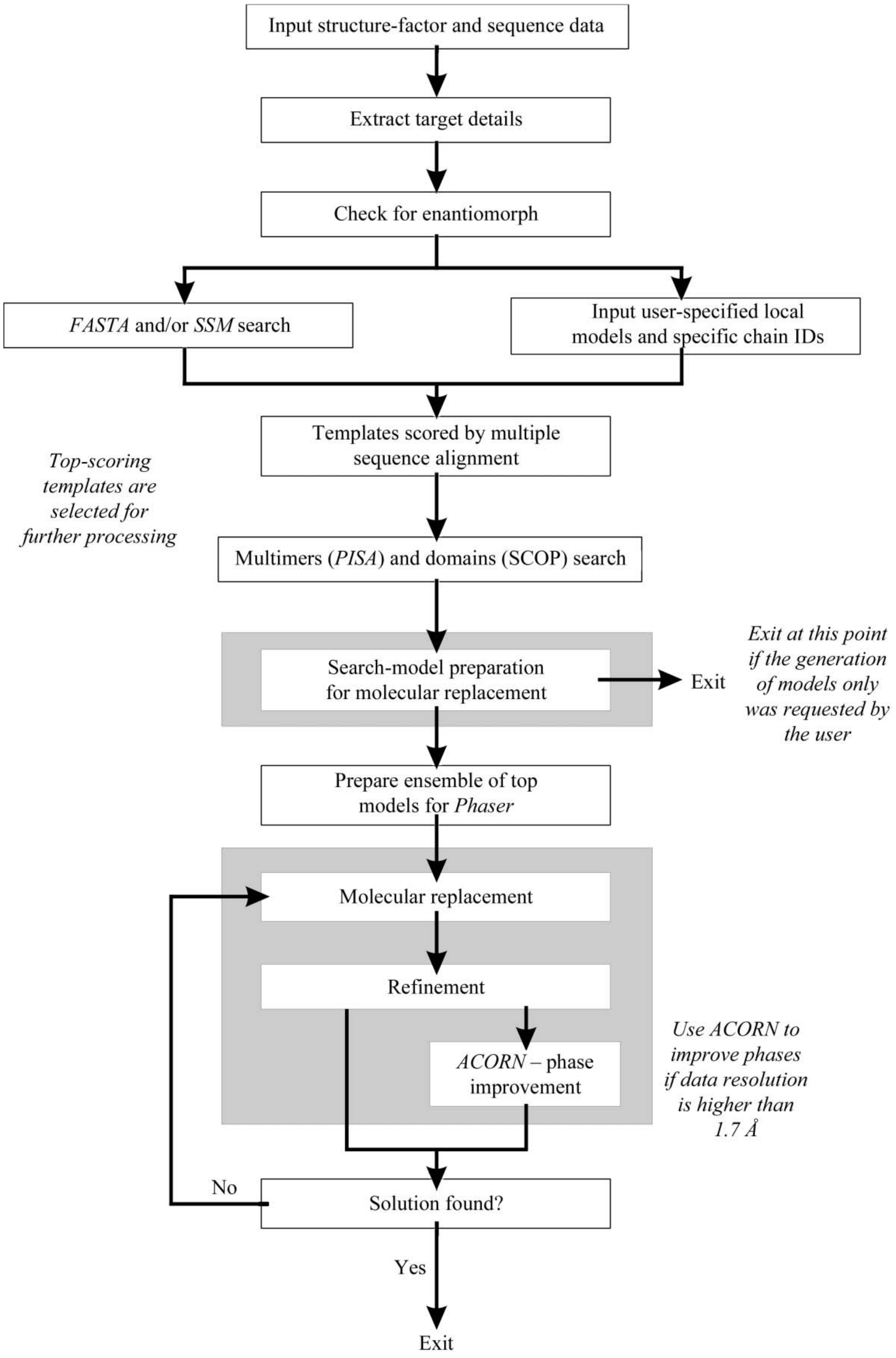
Input structure-factor and sequence data
Extract target details
Check for enantiomorph
FASTA and/or SSM search
Input user-specified local models and specific chain IDs
Templates scored by multiple sequence alignment
Top-scoring templates are selected for further processing
Multimers (PISA) and domains (SCOP) search
Search-model preparation for molecular replacement
Exit
Exit at this point if the generation of models only was requested by the user
Prepare ensemble of top models for Phaser
Molecular replacement
Refinement
ACORN – phase improvement
Use ACORN to improve phases if data resolution is higher than 1.7 Å
No
Solution found?
Yes
Exit

**Figure 1**
Flow diagram of the steps performed in a full run of *MrBUMP*. The starting point is an MTZ file containing structure factors and a single target sequence (the corresponding flow diagram for complexes, not yet implemented, is more complicated). Grey boxes indicate steps that can be run in parallel, for example making use of computer clusters.

*MrBUMP* is a framework which allows a range of techniques and programs to be employed, rather than relying on a single approach. 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. 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.

The current version of *MrBUMP* assumes a single target sequence, although there may be multiple copies of the target molecule in the asymmetric unit. *MrBUMP* does not currently address multi-component systems, *i.e.* complexes and multi-domain proteins where the domains need to be solved separately. In these cases, it can be and has been used to search for the components separately, with the best results being combined manually.

## 2. *MrBUMP* overview

The *MrBUMP* pipeline has been described in detail in a recent article (Keegan & Winn, 2007) and so we give only a brief overview here. Recent developments not described in the previous article are presented in the next section.

The overall scheme of *MrBUMP* is shown in Fig. 1. 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. Each of the stages can utilize a variety of techniques, giving a large degree of flexibility. The process is centred around a list of templates and a list of derived search models and the various techniques operate on these lists.

The first stage currently has three methods of acquiring search-model templates. Firstly, the target sequence is used to search for related proteins in the Protein Data Bank using a simple pairwise alignment as implemented in the *FASTA* package (Pearson & Lipman, 1988). The second method 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. Finally, templates may be specified manually if they are known, either by including a local PDB file or by specifying a PDB code.

In addition to complete chains, search models may be based on individual domains or on multimers. The SCOP database (Murzin *et al.*, 1995; Lo Conte *et al.*, 2002) is checked to see whether any of the templates under consideration includes domains; if so, a new template is constructed for those domains. Multimer templates are constructed if the multimer is biologically relevant (and therefore likely to be transferable between crystal structures) and if it will fit in the target asymmetric unit. The first implementation used the PQS database (Henrick & Thornton, 1998) to identify possible biological multimers. Recent usage of the *PISA* service (Krissinel & Henrick, 2005) is described in the next section.

The sequences of the set of template structures are aligned against the target sequence in a single multiple alignment step. The aim is twofold. Firstly, a template-to-target alignment is required for the *Chainsaw* model-generation step (see below) and that extracted from a multiple alignment is expected to be more reliable than a simple pairwise alignment of the sequences. Secondly, a score for each template is calculated from the multiple alignment and is used to rank the templates for subsequent steps.

In the second stage of *MrBUMP*, one or more search models are derived from each of the selected templates. Currently, four methods are implemented. The 'PDBclip' method simply tidies up the template PDB file, for example removing nonprotein atoms. This is a precursor for other methods, but can be used on its own. The second method generates a traditional polyalanine model from the template structure. The other two methods are more sophisticated and use an alignment to the target to remove sections of main chain that do not align to the target and to truncate side chains of aligned residues according to the conservation of the residue. The third method does this *via* the model-improvement functions of *MOLREP* (Vagin & Teplyakov, 1997), which uses an internal alignment that takes into account the secondary structure of the template. The fourth method uses the program *Chainsaw* (Stein, 2008), which uses an alignment extracted from the previous multiple alignment step. *MOLREP* and *Chainsaw* are similar in purpose, but differ in the details of the alignment used and the extent of side-chain truncation.

At this stage, *MrBUMP* can exit with a list of possible search models. Up to this point, there is no absolute requirement to have diffraction data and therefore *MrBUMP* can be used to generate search models before data collection. This may be useful to assess the need to collect derivative or anomalous data.

In the final stage of *MrBUMP*, the top search models are passed to *MOLREP* (Vagin & Teplyakov, 1997) and/or *Phaser* (McCoy *et al.*, 2005) for molecular replacement. *MrBUMP* passes the target data, a search model and additional information such as the molecular weight of the target to the molecular-replacement program. If the latter locates at least one copy of the search model, then the positioned model is passed to *REFMAC* (Murshudov *et al.*, 1997) for 30 cycles of restrained refinement. The purpose of the refinement step is to assess whether the positioned model is refinable, *i.e.* whether it is both positioned correctly and is a useful starting point for model completion. On the basis of the behaviour of $R_{\rm free}$, the MR solution for a particular search model is classified as a solution, a marginal solution or a failure.

The result of a run of *MrBUMP* is thus a set of search models and results from molecular-replacement trials. In favourable cases, *MrBUMP* will have produced a partially refined model that can be passed to final rounds of model editing and refinement or that can be passed to *ARP/wARP* (Perrakis *et al.*, 1999) for rebuilding. Otherwise, it is often clear that a particular search model is capable of solving the structure and the role of *MrBUMP* has been to direct manual efforts.

## 3. Recent developments

### 3.1. Enantiomorphic space groups

There are 11 pairs of enantiomorphic space groups containing screw axes of opposite handedness. One property of an enantiomorphic space group is that in the absence of anomalous scattering it is indistinguishable from the other member of the pair on the basis of its diffraction pattern. Therefore, unless one has prior knowledge of the space group, both enantiomorphic space groups of a pair need to be tested in MR. The collection of orientations of molecules in the unit cell does not differ between the two space groups, only the translations of these molecules with respect to the origin. Therefore, the rotation function is in principle identical for the alternative space groups and the correct space group is only indicated by the translation and packing functions.

*MrBUMP* has been extended to detect when an input MTZ file is in one of a pair of enantiomorphic space groups and then to give the user the option of testing both possible alternative space groups. *Phaser* already includes such an option (keyword SGAL HAND) and this is invoked by *MrBUMP*. The correct space group is inferred from the top solution provided by *Phaser*. *MOLREP* does not currently have such an option (although it does have an option to test all space groups in a given point group) and so *MrBUMP* runs the translation-function step of *MOLREP* twice, once for each possible space group. The correct space group is chosen as that which gives the best contrast value in *MOLREP*. The subsequent refinement step of *MrBUMP* is performed in the space group selected by the MR step.

When there is a good search model and hence a good MR solution, then the correct space group is obvious from the MR scores. For marginal search models, the discrimination is not so good and the incorrect space group may be chosen. Note that the choice is made independently for each search model.

### 3.2. Phase improvement with *ACORN*

*ACORN* is a program for phase improvement *via* dynamic density modification. The initial phase set can come from a variety of sources and can represent a small fraction of the target structure such as a single heavy atom. One application of *ACORN* is to the improvement of phase sets from MR and the reduction of model bias. Until recently, *ACORN* required atomic resolution, but the latest version of *ACORN* (Jia-xing *et al.*, 2005) has pushed the limit of applicability down to around 1.7 Å. This is achieved by artificially extending the reflection data to 1.0 Å, with several schemes for filling in missing observed amplitudes.

*MrBUMP* now has the option to run *ACORN* after a successful molecular replacement. One aim is to provide better maps for subsequent model completion. The correlation coefficient between the observed and calculated $E$ values in the set of medium reflections is also a good indicator of the quality and correctness of the MR solution. *ACORN* still requires reasonably good resolution and is invoked by default by *MrBUMP* if the target data resolution is better than 1.7 Å. Initial phases are provided by the partially refined search model from *REFMAC*. Trials have shown that refined coordinates provide a better starting phase set for *ACORN* than the unrefined coordinates direct from MR. §4.1 shows an example of the usage of *ACORN* within *MrBUMP*.

### 3.3. Quaternary structures with *PISA*

The *PISA* service (Krissinel & Henrick, 2005) considers all possible sets of protein assemblies that can be generated from the crystal structure and scores them according to an estimated free energy of dissociation. For a given template structure, *MrBUMP* queries the *PISA* server at the European Bioinformatics Institute and retrieves an XML file listing all possible multimers, together with their scores. *MrBUMP* selects those multimers considered to be stable and relevant to the current target and adds them to the list of templates. A coordinate file for each multimer is created using the chain and symmetry information provided by *PISA*. For each multimer template, a number of different MR search models may be generated using some or all of the four methods listed in §2.

The use of *PISA* has a couple of advantages over the earlier use of the PQS database. Firstly, the scoring is expected to be slightly more accurate and trials against a benchmark set of experimentally verified assemblies gave a slightly improved success rate (Krissinel & Henrick, 2005). Secondly, *PISA* gives a full list of possible multimers, so that *MrBUMP* could, for instance, generate both dimer and tetramer search models when trying to solve a tetrameric target.

### 3.4. Alternative multiple alignment programs

Multiple alignment is used in *MrBUMP* to generate scores for the template structures, which inform the order in which MR jobs are run, and to generate the pairwise alignments used by *Chainsaw*. Multiple alignment is still an active area of research, especially in the 'twilight zone' of low (<30%) sequence identities, and tests show that different multiple alignment programs can give significantly different results (see, for example, Ahola *et al.*, 2006). The quality of an alignment is sometimes assessed by comparison with reference alignments, such as BAliBASE (Thompson *et al.*, 1999), but the conclusions of such tests do not necessarily transfer to our particular usage of these programs.

The original version of *MrBUMP* supported two multiple alignment programs, namely *ClustalW* (Chenna *et al.*, 2003) and *MAFFT* (Katoh *et al.*, 2005). While *ClustalW* is the most widely used multiple alignment program, it is no longer considered to be the best available. Given the importance of alignment in generating MR search models it is useful to try other programs and to this end *MrBUMP* now also supports *PROBCONS* (Do *et al.*, 2005) and *T-COFFEE* (Notredame *et al.*, 2000).

In the 'twilight zone' where the alignment is not necessarily reliable, it may be necessary to try different alignments in the *Chainsaw* step, either from the same program or from different programs. This will be automated in future, but for the moment the user can experiment with different alignment programs.

### 3.5. *Smartie*: smart log-file browsing

A new mechanism for parsing the various log files produced by the underlying *CCP*4 programs (Collaborative Computational Project, Number 4, 1994) in *MrBUMP* has been incorporated into the program. Currently being developed by Peter Briggs of the CCP4 team, *Smartie* is a set of Python classes and methods intended to provide tools for parsing the content of *CCP*4 log files. The name '*Smartie*' reflects its origins as the driver for a 'smart logfile browser'. Amongst other things, *Smartie* allows the extraction of tables from the log files according to their title and their presentation in various formats such as log graph, marked up in HTML or in a plain text format. Its main advantages are its robustness and ease of use.

**Table 1**
Selected results for dUTPase from *C. jejuni* (§4.1).

The first column gives the search model used, following the notation described in §4. The column 'Seq. id.' gives the *FASTA* sequence identity against the target for the chain. The column 'RFZ/TFZ' gives the *Z* scores from the *Phaser* rotation and translation functions for the second copy located. The column '$R_{\rm free,i}/R_{\rm free,f}$' gives the initial and final $R_{\rm free}$ values from restrained refinement in *REFMAC*. The column '$\rm CC_i/CC_f$' gives the initial and final correlation coefficient for medium *E* values from *ACORN*. The correctness of the solution indicated in the final column is based on comparison with the final structure. See text for a discussion of the results.

| Search model | Seq id. (%) | RFZ/ TFZ | $R_{\rm free,i}$/ $R_{\rm free,f}$ | $\rm CC_i$/ $\rm CC_f$ | Solution |
|---|---|---|---|---|---|
| 1ogl_A_CHNSAW | 38.3 | 3.3/6.1 | 0.567/0.586 | — | No |
| 1ogk_A_CHNSAW | 37.3 | 3.3/5.0 | 0.564/0.572 | — | No |
| 1ogk_B_CHNSAW | 38.4 | 4.9/6.6 | 0.568/0.516 | 0.075/0.137 | Yes |
| 1ogk_D_CHNSAW | 38.3 | 4.2/10.1 | 0.562/0.509 | 0.109/0.174 | Yes |
| 1ogk_E_CHNSAW | 38.2 | 4.7/11.6 | 0.571/0.505 | 0.078/0.156 | Yes |

## 4. Case studies

Examples of the application of *MrBUMP* have been presented in two previous articles (Bahar *et al.*, 2006; Keegan & Winn, 2007). *MrBUMP* has also been used in a number of structure determinations (Obiero *et al.*, 2006; El Omari *et al.*, 2006; Karbat *et al.*, 2007; Logan, 2008).

Here, we give one example to illustrate the usage of *ACORN* within *MrBUMP*, as well as general features of *MrBUMP*.

### 4.1. dUTPase from *Campylobacter jejuni*

The example target is dUTPase from *C. jejuni* in complex with the substrate analogue dUpNHp, now deposited with PDB code 1w2y. It was solved originally by molecular replacement using *MOLREP* with the structure of the *Trypanosoma cruzi* dUTPase in complex with dUDP (PDB code 1ogk) as the search model (Moroz *et al.*, 2004). It was subsequently used as a test case for *ACORN* in Jia-xing *et al.* (2005). The target asymmetric unit contains two chains of 229 residues. Data are available to 1.65 Å, which is expected to be just good enough to apply the *ACORN* protocol.

The *FASTA* search in *MrBUMP* locates chain *A* of 1ogl and chains *A*, *B*, *D* and *E* of 1ogk, all of which have sequence identities to the target of around 38%. The SCOP search fails to find smaller constituent domains, but PQS indicates that 1ogl and 1ogk exist as dimers. The multiple alignment step aligns the five template chains against the target sequence and pairwise alignments are extracted for use in *Chainsaw*. Scoring based on the multiple alignment step ranks the chains in the order 1ogl_*A*, 1ogk_*B*, 1ogk_*E*, 1ogk_*D*, 1ogk_*A*, although in fact the scores are very similar in this case.

Table 1 shows sample results for monomer search models from a standard run of *MrBUMP*, invoking the *ACORN* option. When referring to search models, we use the nomenclature <PDB code>_<subunit ID>_<model preparation method>, where a subunit can be a chain or a domain. For example, 1ogk_B_CHNSAW refers to chain *B* of 1ogk prepared using the program *Chainsaw*.

*Phaser* finds clear solutions for *Chainsaw* search models based on chains *B*, *D* and *E* of 1ogk. The *Z* scores for the final translation function are large and the $R_{\rm free}$ drops sufficiently in restrained refinement to indicate a marginal solution. Conversely, the *Phaser* solutions for models 1ogk_A_CHNSAW and 1ogl_A_CHNSAW fail to refine. Similar results are obtained using *MOLREP* as the MR program. Unlike the other three chains, chain *A* of 1ogk is not complexed with dUDP in the template structure and adopts a significantly different conformation. Similarly, 1ogl is the crystal structure of the unliganded dUTPase from *T. cruzi* and adopts the same conformation as chain *A* of 1ogk. Since the target is in a bound form, chains 1ogk_*B*, 1ogk_*D* and 1ogk_*E* provide a better search-model conformation. As might be expected from these observations, a dimer search model based on chains *D* and *E* of 1ogk provides a solution, whereas one based on chains *A* and *B* does not.

*MrBUMP* passes the three successful solutions to *ACORN*. In each case, the correlation coefficient shows a clear increase, giving confidence that these solutions are correct. The low absolute values of the correlation coefficient reflect the use of medium-strength *E* values and are typical. Acorn outputs normalized structure factors, phases and weights to the

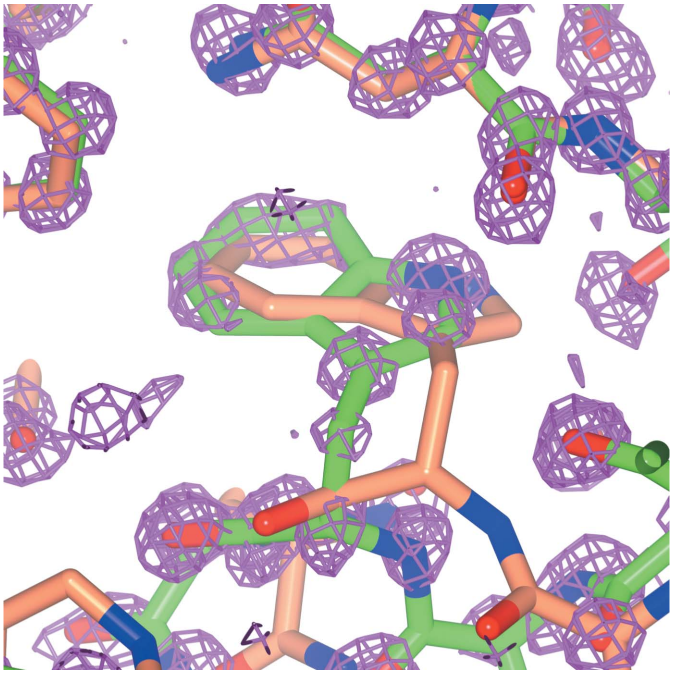

**Figure 2**
An example fragment showing the positioned and refined search model 1ogk_E_CHNSAW (coral) compared with the final deposited structure (PDB code 1w2y, green) for dUTPase from *C. jejuni* (§4.1). The map was generated using phases generated by *ACORN* density modification, starting from the search-model coordinates. At the top of the figure are two regions where the search model matches the final coordinates well. At the bottom of the figure is part of a loop which differs substantially between the search model and the final coordinates. The *ACORN* density clearly follows the correct coordinates rather than being biased to the search model. The map is atomic in nature, as it uses coefficients from the phase-extension procedure. It also noticeable that *ACORN* picks out O and N atoms better than C atoms. The figure was prepared using *CCP4mg* (Potterton *et al.*, 2004).

fictitious high-resolution limit and these can be used to generate atomic style maps which are better than maps calculated to the true resolution (Jia-xing *et al.*, 2005). Fig. 2 shows part of such a map, together with the positioned and refined search model used as input to *ACORN* and the final deposited structure. The search model 1ogk_E_CHNSAW possesses the same overall fold as the target, but nevertheless there are disagreements in some loop regions. At the bottom of the figure is part of a loop which differs substantially between the search model and the final coordinates. The *ACORN* density clearly follows the correct coordinates rather than being biased to the search model. Finally, the *ACORN* phases can be used as input to model-building programs such as *ARP*/*wARP* (Perrakis *et al.*, 1999), which rebuilds the model in the correct location.

## 5. 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.

This work was supported by the BBSRC through the e-HTPX and CCP4 grants. We thank all users of *MrBUMP* for useful feedback and encouragement.

## References

Ahola, V., Aittokallio, T., Vihinen, M. & Uusipaikka, E. (2006). *BMC Bioinformatics*, **7**, 484.

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

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

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.

Do, C. B., Mahabhashyam, M. S. P., Brudno, M. & Batzoglou, S. (2005). *Genome Res.* **15**, 330–340.

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

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.

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

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

Jia-xing, Y., Woolfson, M. M., Wilson, K. S. & Dodson, E. J. (2005). *Acta Cryst.* D**61**, 1465–1475.

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.

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

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.

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

Logan, D. (2008). Submitted.

Long, F., Vagin, A., Young, P. & Murshudov, G. N. (2008). *Acta Cryst.* D**64**, 125–132.

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

Moroz, O. V., Harkiolaki, M., Galperin, M. Y., Vagin, A. A., González-Pacanowska, D. & Wilson, K. S. (2004). *J. Mol. Biol.* **342**, 1583–1597.

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.

Notredame, C., Higgins, D. G. & Heringa, J. (2000). *J. Mol. Biol.* **302**, 205–217.

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

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., 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., 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.

Stein, N. D. (2008). *J. Appl. Cryst.* Submitted.

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

Thompson, J., Plewniak, F. & Poch, O. (1999). *Bioinformatics*, **15**, 87–88.

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