Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

BALBES: a molecular-replacement pipeline

Fei Long,‡ Alexei A. Vagin,‡ Paul Young and Garib N. Murshudov*

York Structural Biology Laboratory, Chemistry Department, University of York, Heslington, York, England

‡ These authors contributed equally to this work.

Correspondence e-mail: garib@ysbl.york.ac.uk

Received 8 May 2007
Accepted 12 October 2007

The number of macromolecular structures solved and deposited in the Protein Data Bank (PDB) is higher than 40 000. Using this information in macromolecular crystallography (MX) should in principle increase the efficiency of MX structure solution. This paper describes a molecular-replacement pipeline, *BALBES*, that makes extensive use of this repository. It uses a reorganized database taken from the PDB with multimeric as well as domain organization. A system manager written in Python controls the workflow of the process. Testing the current version of the pipeline using entries from the PDB has shown that this approach has huge potential and that around 75% of structures can be solved automatically without user intervention.

1. Introduction

The number of macromolecular structures deposited in the Protein Data Bank (PDB; Berman *et al.*, 2000) is increasing rapidly every year. For example, out of more than 40 000 entries, around 5500 (more than 12%) were deposited and released in 2006. X-ray crystal structure analysis (MX) is by far the most common technique used for the determination of three-dimensional structures (approximately 83%), followed by NMR with around 15%.

The PDB is a treasure of the structural biology community, the implications of which have yet to be fully appreciated. One can imagine the amount of information contained in this repository. How do we extract and analyse this information and use it to understand fundamental biological problems such as protein folding and protein evolution? This and other questions are the subject of many research disciplines, including bioinformatics. There have already been huge amounts of work carried out in this area. Two areas relevant to this paper are the classification of domains [CATH (Pearl *et al.*, 2005); SCOP (Murzin *et al.*, 1995)] and the extraction of biological oligomers from crystal structures (Krissinel & Henrick, 2005). While the domains defined by both CATH and SCOP are extremely useful for the biological community in general, our attempts to use them for molecular replacement did not produce consistent results. Therefore, we undertook to redefine the domains so that they could be used for molecular replacement and structure solution routinely and consistently.

One of the obvious applications of the PDB is the reuse of entries for macromolecular X-ray crystallography. The application of information derived from the PDB for molecular replacement, phase improvement (Terwilliger & Berendzen, 1999) and model building (Emsley & Cowtan, 2004; Jones *et al.*, 1991) now routinely takes place. In the near future, one can envisage that information that is invariant for all entries in the PDB (or classes of proteins) will be used during all stages of

structure analysis, thereby transferring information from high-resolution structures to new structure analysis, thus increasing the reliability of the derived models. Moreover, one can speculate that the celebrated phase problem may well be solved using substructure classes (*e.g.* domains) from the PDB by applying well established ideas such as the multi-solution techniques (Germain *et al.*, 1970) used in the small-molecular crystallographic world.

Analysis of the PDB shows that molecular replacement (MR) is the most widely used technique for macromolecular crystal structure solution. 67% of all X-ray structures released in 2006 were solved using this method (Fig. 1). It is expected that with (i) better organization of the database for molecular replacement, (ii) a better choice of protocols and (iii) improved algorithms in molecular replacement and refinement, this percentage will be significantly higher. However, it should be noted that the PDB reflects successful structure solution and therefore all statistical analysis derived from it will inevitably be biased.

In recent years, there has been an explosion of developments of automatic procedures for macromolecular X-ray structure solution. These approaches have already produced several highly automated and very popular software packages for automatic model building and refinement (*ARP*/*wARP*; Perrakis *et al.*, 1999) and for automatic phasing and model building [*SOLVE*/*RESOLVE* (Terwilliger & Berendzen, 1999), *CRANK* (Ness *et al.*, 2004) and *Auto-Rickshaw* (Panjikar *et al.*, 2005)]. Despite the high productivity of the molecular-replacement technique, until recently it was not applied in automation procedures. Nevertheless, several automated molecular-replacement pipelines have already been made available to the user community, including *NORMA* (Delarue, 2008), *MrBUMP* (Keegan & Winn, 2008) and part of the JSCS structure-solution pipeline (Schwarzenbacher *et al.*, 2008). All of these approaches are built around one or more of the popular molecular-replacement programs *AMoRe* (Navaza, 1987), *MOLREP* (Vagin & Teplyakov, 1997; Lebedev *et al.*, 2008) and *Phaser* (Storoni *et al.*, 2004).

This paper describes *BALBES*, a fully automatic molecular-replacement pipeline.

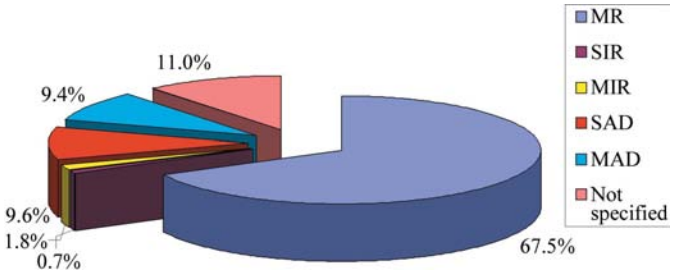


Figure 1
A pie chart showing the various methods used to determine X-ray structures for PDB entries released during 2006.

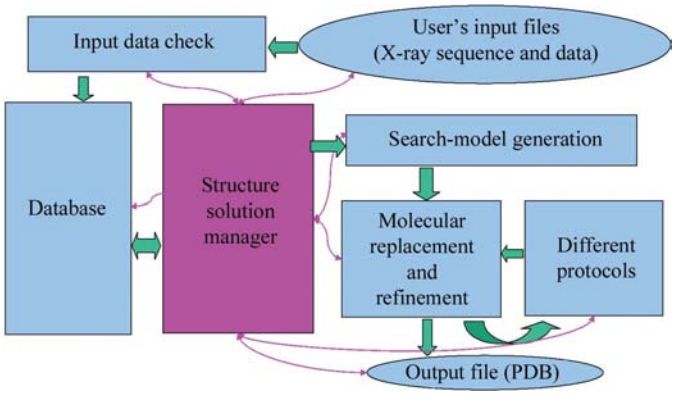


Figure 2
A schematic view of the *BALBES* workflow. All decisions are made internally according to the amount of data (reflections and sequence) and the stage of structure solution. The pink arrows show that the manager controls all the activities involved and the green arrows show the directions of the workflow.

2. Overall organization

BALBES, a system for fully automating molecular replacement, consists of three major components, which were developed independently of each other. These are (i) a reorganized database of protein structures, (ii) a system manager that controls the workflow and makes decisions according to the available information and (iii) scientific programs, which are the powerhouse of the system. The overall workflow of the system is shown in Fig. 2. Some details of these components are given in the following sections.

3. Database of macromolecular structures

3.1. Selection of entries

All protein entries from the PDB with a length greater than 15 amino-acid residues that had been solved using MX and had been refined against data higher than 3.5 Å resolution were selected to build the current database. A basic entry in the database was a macromolecular subunit. If two subunits had a sequence identity greater than 80% and a root-mean-square deviation (r.m.s.d.) between corresponding C^{α} atoms of less than 1 Å, then the one that had been refined against the higher resolution data was retained. This approach, while substantially reducing the number of subunits kept in the database, retained the conformational variability of the molecules. For example, if there were two copies of a subunit and there was a domain motion between these subunits, then both representatives were kept in the database even if the sequence identity was 100%.

For each entry sequence, information about the secondary structure, domains (see below) and potential to form multimers was also stored. Therefore, when an entry was extracted, all necessary information was immediately available.

All entries in the database (around 14 000 subunits) were aligned with each other using a modified version of the Needleman & Wunsch (1970) dynamic alignment algorithm. The result of this alignment was considered as a measure of similarity. Using this, a hierarchical database was organized with agglomerative clustering. The results were kept as a search tree.

3.2. Domains

All domains were analysed and checked manually. The main criteria for domain definition were three-dimensional compactness and separability from other parts of the subunit. However, if there was no well defined domain in a molecule then the whole molecule was considered as a domain. If a tentative domain contained completely exposed loops and N- or C-terminal stretches, they were considered as flexible parts and were removed from the domains. The result of this analysis was approximately 23 000 domains. Each domain belonged to a subunit and each subunit belonged to a class as a result of clustering. All domains were aligned with each other again and further superimposed using three-dimensional fitting algorithms (Kabsch, 1976). Quality factors (Q-factors) were calculated using the procedure described by Krissinel & Henrick (2004). The Q-factors were used in hierarchical clustering of the domains. Once clusterization of the domains was finished, they were used to check and correct the clustering of each entry (subunits). This procedure ensured that subunits and domains belonging to the same class were similar in three-dimensional structure and not merely in sequence. It should be noted that domains were kept in the database as a set of operations which was necessary to generate them from the basic entries (subunits).

3.3. Multimers

Multimers for each entry were taken from the EBI's PISA service (http://www.ebi.ac.uk/msd-srv/prot_int/pistart.html) for multimer generation from crystal structures (Krissinel & Henrick, 2004). Multimers are stored as operations to generate them from basic entries (subunits). This substantially reduces the amount of information stored in the database.

The database also contains a full list of PDB entries with their unit-cell parameters and space groups. This list helps to search the PDB using cell and symmetry only.

3.4. Update

Every 15 d, the database is updated using newly deposited structures. If the sequence and three-dimensional structure of the newly deposited structures are similar to an entry in the existing database, then their domain definitions are also transferred. For the remaining structures, manual analysis is carried out. Currently, even automatically generated domains are checked manually to make sure that automatic domain-definition transfer does not introduce errors.

3.5. Search using a single sequence

When a sequence is given, a search is carried out in the database at the appropriate level. For one member of the database belonging to a branch of the tree, sequence alignment is carried out and the score, relative aligned length and number of gaps are calculated. A new quality factor is then calculated,

$$\mathrm{CQ} = \mathrm{score}\ \exp\left\{-\left[1-\frac{N_{\mathrm{align}}}{\max(N_1, N_2)}\right]^2\right\}\left(1-\frac{N_{\mathrm{gap}}}{N_{\mathrm{align}}}\right)^4, \quad (1)$$

where 'score' is based on the normalized BLOSUM62 substitution matrix (Henikoff & Henikoff, 1992), N_1 and N_2 are the number of residues in the first and the second sequence, N_{align} is the number of aligned residues and N_{gap} is the number of gaps. This function seemed to work consistently better than many other functions that were tried.

Afterwards, the branch corresponding to the maximum of CQ (maxCQ) is taken and this branch is considered to be similar. If maxCQ < 0.22, then it is considered that there is no similar structure. If a branch is similar to a given sequence, then at most 20 of the best aligned structures with their domain and multimeric organizations are taken from this branch as templates.

If no similar structure is found among the basic entries, if the maximum of CQ is less than 0.60 or if the number of residues aligned with gaps is more than 40 then the system carries out a domain search. Firstly, it uses the full-length sequence and tries to find a similar domain. When stretches of the sequence corresponding to this domain are found, they are removed and the remaining sequence is submitted to a further domain search. At this stage, the remaining sequence is considered as a fragment of sequences. If another domain is found, the search continues until all domains have been found or the remaining sequence stretches are too fragmented (*i.e.* the longest length of a fragment in the remaining sequence is less than 40 residues). This procedure ensures that all domains are found that may be present in the different entries. An example of such a case is shown in Fig. 3. PDB entry 1z45 has two major domains, one of which can also be split into two smaller domains. Domain 1 is similar to 1ek6 (with sequence identity 55%) and domain 2 is similar to 1yga. The domain search considers domain 2 as two separate domains and finds a similar domain for domain 2-1 from 1yga (with sequence identity 51%) and for domain 2-2 from 1udc (49%).

3.6. Search for assemblies

If an input file contains more than one sequence then the system assumes that it is a complex of proteins. In this case, it searches for assemblies consisting of these or a subset of these sequences. If they are found then they are used as template models for molecular replacement and refinement. If no such assembles are found then each sequence is searched in turn and a set of template models is generated for each sequence (with their multimeric as well as their domain structures).

4. Design of the system manager

4.1. Scripting language for the system manager and method of passing parameters

A system manager is needed to integrate the database of macromolecular structures with the scientific software. It should make decisions according to the information that it has

available and should provide a user-friendly interface for non-expert users as well as other programs (*e.g.* a graphical user interface or other pipelines that may incorporate this system). This places several requirements on the computing language of the system manager.

(i) Flexibility: it needs to seamlessly integrate the existing crystallographic software, which may have been developed using very different computing languages (such as Fortran, C and C++).

(ii) Modularity: each protocol or algorithm implemented in the system should work as a module. The modules can be assembled to form new modules and communicate with each other by passing parameters, *e.g.* in the form of Extensible Markup Language (XML). This feature is very important for the future development and update of the system. The manager should also allow the addition of more complicated protocols, which we probably do not know yet. These new modules should be easily plugged into the system without affecting the pre-existing modules.

(iii) Reusability: the reusability of elements is another important feature for rapid and efficient development of the system. It appeared that a scripting language allowing object orientation, *i.e.* Python, was the most appropriate for designing this system.

Communication between different modules of the system (database, programs) was carried out using an XML file format. *BALBES* uses a Python extension, PyXml (http://pyxml.sourceforge.net/), to process XML files.

4.2. Implementation of the system manager

In the *BALBES* system manager, all of the scientific programs are wrapped into Python classes that are descendants of an abstract class: this abstract class contains those procedures which are common in running a scientific program, such as calling the program, tracing the running process ID, killing the job *etc*. Different data are also wrapped as various Python classes to accommodate the needs of parameter passing; for example, the class CModel is designed to record and manipulate all the information required for a template model at different stages of finding a solution, such as its chain ID, sequence identity, the multimers and domains it may contain, the parameters needed and the resultant outputs when working on it by MR and refinement. Different combinations of the objects of these classes form independent modules that perform different functionalities.

The overall workflow in *BALBES* is shown in Fig. 2. After the user's input structure-factor file has been provided, it is analysed using *SFCHECK* and all necessary information is extracted (such as the unit-cell parameters, space group, data completeness, optimal resolution, the pseudo-translation vector if it exists, twin operators and estimates of the twin fractions). Next, *BALBES* begins to analyse the sequence, unit-cell parameters and space group. If the space group is the same as one of the entries and the unit-cell parameters are very similar (the maximum difference in unit-cell lengths and angles between the target and search crystals is less than 0.5%), then the system tries to use this PDB entry for refinement. This is performed to account for potential mistakes that may arise during expression and crystallization. If the differences in the unit-cell parameters are within 5% (the corresponding maximum difference is less than 5%) and the sequence identity is greater than 90%, then the system again tries to use this PDB entry for refinement. If refinement does not produce a desirable $R/R_{\rm free}$, the system then starts the automated molecular-replacement runs. A desirable $R/R_{\rm free}$ in the current version is determined according to the following procedure.

Let $\Delta R_{\rm free} = (R_{\rm free} - R_{\rm free\_init})/R_{\rm free}$.

(i) If $R_{\rm free} \leq 0.35$ then the structure is considered 'solved' regardless of the value of $\Delta R_{\rm free}$.

(ii) If $0.35 < R_{\rm free} \leq 0.45$ then the structure is marked as 'solved' if $\Delta R_{\rm free} < 0.0001$, which means that $R_{\rm free}$ could slightly increase.

(iii) If $0.45 < R_{\rm free} \leq 0.50$ then $\Delta R_{\rm free}$ must be less than -0.05 for the structure to be considered as 'solved', which means that $R_{\rm free}$ should decrease.

(iv) If $R_{\rm free} > 0.50$ and $\Delta R_{\rm free} > 0.03$, then the structure is considered to be 'not solved'.

(v) All other cases are considered as potential solutions.

The first job in automated molecular replacement is to find the template structures by searching the internal database. The algorithms and criteria

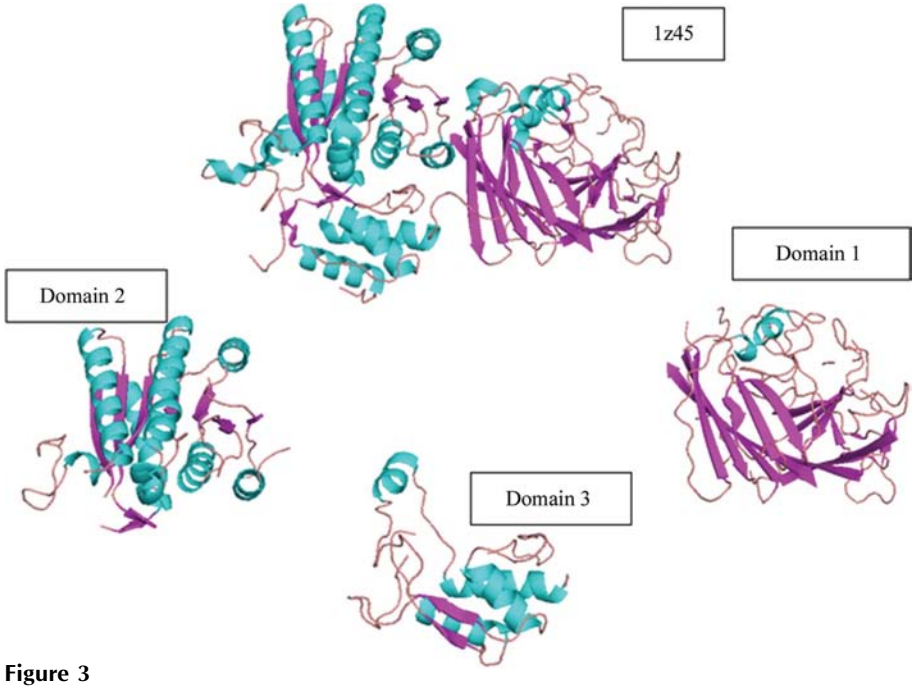


Figure 3
An example of a search for several domains from the domain database. The target structure (1z45) has three domains. The system finds all domains step by step. These domains belong to 1udc (domain 1), 1yga (domain 2) and 1ek6 (domain 3).

for this are detailed in the previous section. Currently, we select those with CQ > 0.22 as the template structures. When this process has finished, users are provided with a group of template structures as detailed in the previous section. *BALBES* works on these structures in turn according to their priorities. That is, if assemblies are found *BALBES* will use the structures in these assemblies as search models, then the structures associated with different single sequences and finally the structure formed by domains from different PDB entries. Usually, several template structures are found in an assembly or associated with a sequence. The system manager starts with the template structure with the highest sequence identity, then the second structure and then the third structure. For each structure, multimer models, if they exist, are tried first and then the monomer models. There are different protocols used to carry out MR. The most widely used protocol is a combination of MR and refinement on a whole template structure. As a simple example, Table 1 presents a template structure found by *BALBES* that is associated with one sequence in which there are four search models. MR is performed on the trimer model first, followed by refinement. If it is not considered to be a solution (currently using the behaviour of $R_{\rm free}$ as defined above) the dimers and then the monomers are tried. If no solution is found for the whole multimers or monomers and domains exist, a more complicated set of protocols is employed.

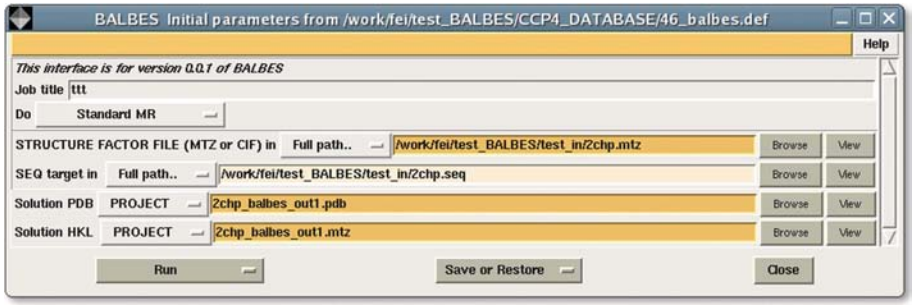


Figure 4
BALBES ccp4i interface.

Table 1
Search models in a template structure.

PDB code 1ji5; No. of models = 4.

Model	Chain ID	Similarity	Residues	Multimer?	Domain?	Monomers
1	*A*	0.5	142	Monomer	No	5
2	*A*	0.5	119	Monomer	Yes	5
3	*AB*	0.5	284	Dimer	No	2
4	*ABC*	0.5	426	Trimer	No	1

5. Programs

The system uses currently available programs including *MOLREP* (Lebedev *et al.*, 2008), *REFMAC* (Murshudov *et al.*, 1997) and *SFCHECK* (Vaguine *et al.*, 1999). The system makes use of these programs and at the same time tests them. This means that these programs are constantly tested using thousands of test cases. Improvements based on these tests increases the robustness of these programs, while increasing the power of the system in the next release.

The most interesting aspect of these tests is the analysis of failed cases. Having a huge amount of test cases helps to prioritize future developments and their analysis helps to generate new ideas for phasing, molecular replacement, model building and refinement.

6. Interfaces

Three types of user interface have been developed for *BALBES*. First and foremost is the command-line interface. This interface also forms the basis for the other two interfaces, the *ccp*4*i* (Potterton *et al.*, 2003) interface, which allows the use of the tools available within *ccp*4*i*, and the web interface, which allows the use of tools developed for web browsers.

6.1. Command-line interface

The command-line interface takes inputs of sequence and data,

```
balbes -f <data> -s <sequence> -o <output>,
```

where `data` is a file containing experimental data from the crystal under study, `sequence` is the file containing the sequence(s) of the unknown structure and `output` is a subdirectory where information about the template structures, results and details of the working system are written. The currently accepted file formats for experimental data are MTZ (Collaborative Computational Project, Number 4, 1994) and CIF (Hall *et al.*, 1991). The sequence format is FASTA.

If a user wants to use his own library of structures then this can be performed using

```
balbes -f <data> -l <LibraryOfModels> -s <sequence>
```

where `data` and `sequence` are defined as above and `LibraryOfModels` is a subdirectory containing PDB files.

If a user wants to use his particular model then this can be performed using

```
balbes -f <data> -m <model>
```

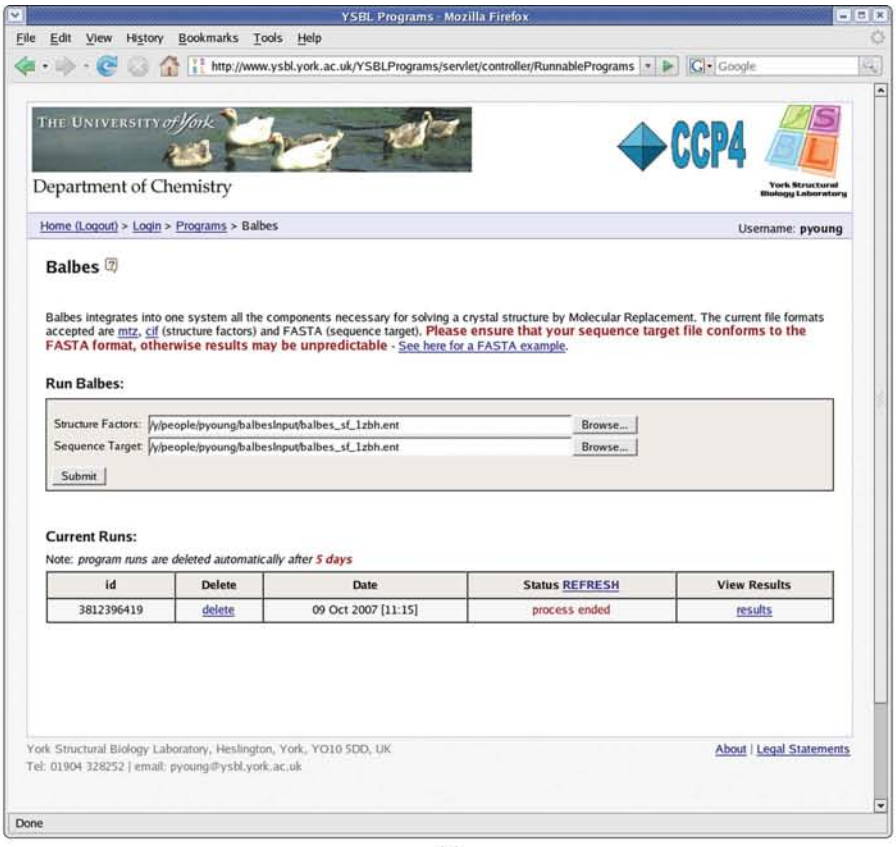
or

```
balbes -f <data> -m <model> -s <sequence>
```

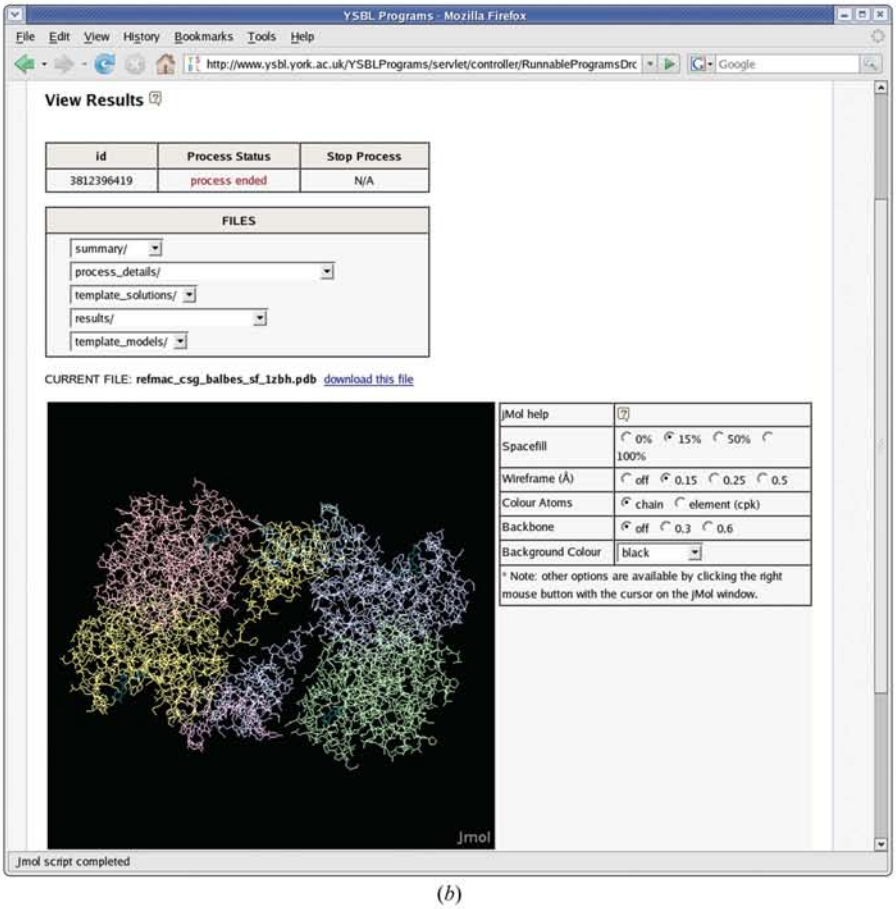
where `model` is now an input PDB file.

6.2. *ccp*4*i* interface

Fig. 4 shows an example of the *ccp*4*i*-style interface. The user only needs to provide a sequence and an experimental data file. Although the input is sufficiently simple, the output files contain all the process information, including the results of the analysis of the data by *SFCHECK*, *REFMAC* and *MOLREP*. If a solution is found, then a PDB file and an MTZ file containing the weighted coefficients corresponding to the refined models are also given.



(*a*)



(*b*)

6.3. Web interface

Figs. 5(*a*) and 5(*b*) show the *BALBES* web interface. The user is required to upload data and sequence information and the process is then run. Output files are displayed according to their type; for example, if the output is a PDB file either it can be downloaded to the local computer or displayed using *Jmol* (http://jmol.sourceforge.net/).

7. Calibrating the system

We are testing *BALBES* systematically during its development, which has proven to be beneficial to both the development of the whole system and of its individual components, including the incorporated scientific programs. While updating the database, the structure factors (if available) are also taken from the PDB. For these structures, *BALBES* runs automatically using the previous database and the results are compared with those of the final structures. The program developed for this purpose, *solution_check*, performs the comparison of these structures. This program compares two sets of PDB coordinates using all possible origins specific for this space group. Table 2 shows tests carried out during 2006. After each session of tests, a detailed analysis of failed cases is carried out. If the reason for failure is clear and the program responsible for the failure can be identified, then that particular program is updated. If necessary, new algorithms are then designed and implemented to fix the problem. This has already enhanced the efficiency of *BALBES* and we have developed and implemented several new protocols (or algorithms) for both

Figure 5
BALBES web server. *BALBES* can be run from the YSBL programs website by uploading a structure-factor file and sequence-target file to the web server (*a*), which interacts with *BALBES via* a program poller. For example, the poller looks for a `startingProcess` file on the web server; when this is found, *BALBES* is run (on a separate host) and output files are copied across to the web server. The user can then view the output files by selecting an option from one of the drop-down menus (*b*). At present, viewable file types are text files, MTZ, PDB (using *Jmol*) and PDF.

Table 2
Test statistics for structure-factor files released between 1 January 2006 and 9 October 2006 (files released between 5 August and 21 September 2006 are excluded).

Method	No. of cases	Cases solved	Success rate (%)
All	3136	2323	74.1
MR	2090	1759	84.1
SIR	21	5	23.8
MIR	57	13	22.8
SAD	263	78	29.7
MAD	305	104	34.1
Other†	400	364	91.0

† The techniques used for structure determination are not properly specified in the PDB file. These are most probably specified in the structures of isomorphous crystals.

the individual scientific programs and *BALBES* itself. One of these protocols is shown in Fig. 6. This protocol combines refinement and several options of molecular replacement.

The current version of the system does not include nucleic acid structures and structures solved by NMR. We are currently developing techniques and protocols for the efficient use of these structures. Both these type of entries have their peculiarities that need to be taken into account before including them in the system.

The current success rate is around 75%, as shown in Table 2. It should be noted that structures are usually deposited in packs, *i.e.* one structure is solved using experimental phasing and then several related structures are solved using this method before all structures are deposited to the PDB simultaneously. If all search structures become available, then one can expect that this percentage will be higher. However, as was mentioned above, the PDB contains solved structures and thus all statistics based on this data bank are necessarily biased towards them. Therefore, the real success rate of the system is difficult to judge.

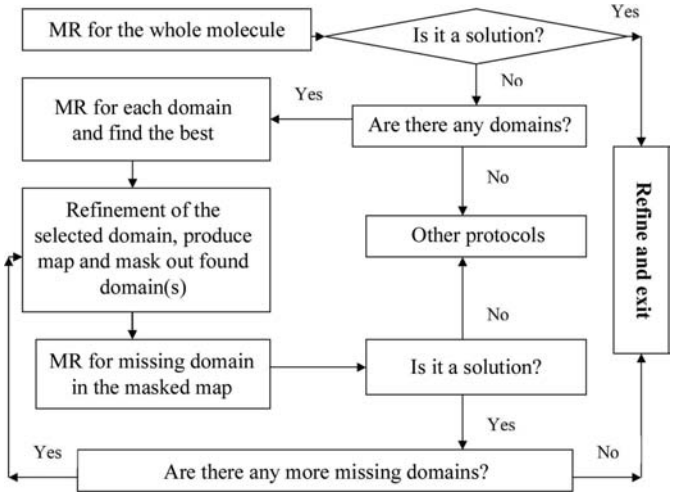


Figure 6
A protocol for the combination of refinement and molecular replacement, with and without phases, when domains exist in the search model.

8. An example of the application of *BALBES*: multidomain protein 1z45

In this example, we use a multidomain protein in which the domains are from different molecules (see Fig. 3). Once the domains have been found, a simple molecular replacement is carried out using the largest domain and a very good contrast solution is found, which is then refined. R and $R_{\rm free}$ after refinement of only one domain are 33% and 41%, respectively. Next, the refined model is used and weighted structure map coefficients are calculated in *REFMAC* to search for smaller domains in the electron density. The system finds the second domain and refines the first two domains. The system then tries to find the third domain but fails to do so. The reason for this is that it is too small and the packing function may prevent it solving this. It is a small fragment and the problem is a model-completion problem that can be solved using, for example, *ARP*/*wARP*.

9. Conclusions and future perspectives

The organization of the database for macromolecular crystal structure solution is an important ingredient in designing automatic pipelines. We have designed such a database and as a proof of principle it has been successfully integrated into the *BALBES* molecular-replacement pipeline. Further development of this database is currently is under way. Future versions of the database will include several important features including molecule formation, operation from domains and analysis of these formations for compactness and variability, design and the regular update of sequence profiles for each domain class.

Tests using the *BALBES* system have shown that with relatively simple protocols around 75% of all structures available in the PDB can be solved by MR automatically. We are currently analysing successful and unsuccessful cases. Successful cases are provided to developers of *ARP*/*wARP* for testing of automation. Unsuccessful cases are analysed by us to improve the molecular-replacement and refinement programs and procedures. These cases are available from the authors on request.

The system is currently under intensive development. For example, the procedures described by Isupov & Lebedev (2008) and Lebedev *et al.* (2008) will be implemented in future versions of the system.

A future version of the system will also include decisions on such important aspects of crystallography as the correction of false origins when these are encountered (Lebedev, private communication) and automatic recognition and use of twinning during structure solution and refinement (Zhou, 2005). One of the advantages of an automatic pipeline is that information can easily be extracted during structure solution and used when it is necessary. If a structure is solved by molecular replacement, then information about the model used can be utilized in refinement. For example, information about domains and/or secondary structures could be used during model building as well as refinement. It might be important

when a search model is refined against high-resolution data and the target is at low resolution.

In future, it is expected that this system will be linked with *ARP*/*wARP* and/or other automatic model-building procedures, thus completing the automation of molecular replacement. Combining this procedure with existing automatic experimental phasing procedures such as *CRANK* (Ness *et al.*, 2004) and *Auto-Rickshaw* (Panjikar *et al.*, 2005) would truly complete the automation of structure solution.

The system is currently available from http://www.ysbl.york.ac.uk/~fei/balbes/download. When it is ready, it will be made available to the user community *via* the CCP4 download site http://www.ccp4.ac.uk.

We thank Andrey Lebedev for discussions and useful suggestions and Misha Isupov, Gleb Bourunkev and Victor Lamzin for testing and useful feedback. This work was supported by the Wellcome Trust (FL and GNM; grant No. 064405/Z/01/A), BBSRC (AAV; grant No. 1 RO1 GM069758-03) and BIOXHIT (FL and PY; grant No. LSHG-CT-2003-503420). The computers used for testing the system were acquired using funds from NIH (grant No. 1 RO1 GM069758-03) and Wellcome Trust grants.

References

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.
Collaborative Computational Project, Number 4 (1994). *Acta Cryst.* D**50**, 760–763.
Delarue, M. (2008). *Acta Cryst.* D**64**, 40–48.
Emsley, P. & Cowtan, K. (2004). *Acta Cryst.* D**60**, 2126–2132.
Germain, G., Main, P. & Woolfson, M. M. (1970). *Acta Cryst.* B**26**, 274–285.
Hall, S. R., Allen, F. H. & Brown, I. D. (1991). *Acta Cryst.* A**47**, 655–685.
Henikoff, S. & Henikoff, J. G. (1992). *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.
Isupov, M. N. & Lebedev, A. A. (2008). *Acta Cryst.* D**64**, 90–98.
Jones, T. A., Zou, J.-Y., Cowan, S. W. & Kjeldgaard, M. (1991). *Acta Cryst.* A**47**, 110–119.
Keegan, R. & Winn, M. (2008). *Acta Cryst.* D**64**, 119–124.
Kabsch, W. (1976). *Acta Cryst.* A**32**, 922–923.
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.
Krissinel, E. & Henrick, K. (2004). *Acta Cryst.* D**60**, 2256–2268.
Lebedev, A., Vagin, A. A. & Murshudov, G. N. (2008). *Acta Cryst.* D**64**, 33–39.
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.* **147**, 536–540.
Navaza, J. (1987). *Acta Cryst.* A**43**, 645–653.
Needleman, S. B. & Wunsch, C. D. (1970). *J. Mol. Biol.* **48**, 443–453.
Ness, S. R., de Graff, R. A. G., Abrahams, J. P. & Pannu, N. S. (2004). *Structure*, **12**, 1753–1761.
Panjikar, S., Parthasarathy, V., Lamzin, V. S., Weiss, M. S. & Tucker, P. A. (2005). *Acta Cryst.* D**61**, 449–457.
Pearl, F. *et al.* (2005). *Nucleic Acids Res.* **33**, D247–D251.
Perrakis, A., Morris, R. & Lamzin, V. S. (1999). *Nature Struct. Biol.* **6**, 458–463.
Potterton, E., Briggs, P., Turkenburg, M. & Dodson, E. (2003). *Acta Cryst.* D**59**, 1131–1137.
Schwarzenbacher, R., Godzik, A. & Jaroszewski, L. (2008). *Acta Cryst.* D**64**, 133–140.
Storoni, L. C., McCoy, A. J. & Read, R. J. (2004). *Acta Cryst.* D**60**, 432–438.
Terwilliger, T. C. & Berendzen, J. (1999). *Acta Cryst.* D**55**, 849–861.
Vagin, A. & Teplyakov, A. (1997). *J. Appl. Cryst.* **30**, 1022–1025.
Vaguine, A. A., Richelle, J. & Wodak, S. J. (1999). *Acta Cryst.* D**55**, 191–205.
Zhou, D. (2005). PhD thesis. University of York, York, England.