Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

Sequence alignment for molecular replacement

Geoffrey J. Barton

School of Life Sciences, University of Dundee, Dundee DD1 5EH, Scotland

Correspondence e-mail: geoff@compbio.dundee.ac.uk

Received 8 May 2007
Accepted 20 September 2007

This article focuses on the key step of obtaining the best possible sequence alignment of the Query (the protein you are interested in) to the Target (a protein of known three-dimensional structure) in order to build a molecular model for molecular replacement. Common sequence-alignment methods are discussed, starting from structural alignment and then moving to pairwise, multiple and profile–profile methods. The limitations of sequence-alignment methods and guidelines on how to judge the likely accuracy of alignment are considered. This is not a detailed tutorial on how to use specific programs; rather, the reader is directed to current tools and techniques that are likely to yield good results.

1. Introduction

A molecular-replacement search requires a model of the protein of interest (the Query). The first step in building the model is to identify one or more proteins of known three-dimensional structure (the Targets) that are similar to the Query. The subject of this article is the second step, which is the generation of an accurate sequence alignment of the query to the identified target protein(s). There is a bewildering array of sequence-alignment tools and techniques and in this short article I will not attempt to make a comprehensive review. I first consider what the problem of sequence alignment is and then discuss the basic ways in which alignments may be constructed and evaluated. Where possible, I give suggestions of recently developed or enhanced software to try with some comment on the limitations of all methods.

2. Sequence alignment, structure, evolution and function

What does a sequence alignment actually represent? There are two views on this. The first, and most important for molecular modelling, is to consider only the present-day sequences that are being aligned and the structural and functional importance of each amino acid in each protein. In this context, if two amino acids are aligned the implication is that they are performing similar structural and functional roles in the two proteins. If the three-dimensional structures of both proteins are known then it follows that the most accurate alignment of present-day sequences can be obtained by simultaneous consideration of the structures and, ideally, knowledge of functional sites. The second view of an alignment is based on the principle that present-day sequences have evolved from a common ancestor. An alignment in this

context should reflect the process of mutation, insertion and deletion that has occurred over the course of evolution since the last common ancestor sequence. While this may produce a convincing alignment, it may not reproduce the best structural alignment as judged by present-day proteins. The vast majority of sequence-alignment programs only have the sequence to work from and so attempt to reproduce evolutionarily reasonable alignments. Without a reliable method to predict the three-dimensional structure of the protein from sequence alone, this is the best that they can do. Since for modelling one is usually interested in transferring structural information from one protein to the other, it is important to understand how well an automatically generated sequence alignment can reproduce the structural alignment. However, before looking at this, we must first understand the challenges of protein structural alignment.

2.1. Structural alignment: a gold standard for sequence alignment

Many techniques have been developed for the alignment of protein three-dimensional structures. The majority focus on the problem of searching databases for similarities to a newly determined structure (see Novotny *et al.*, 2004, for a recent evaluation of structure-searching servers). In contrast, there are comparatively few methods optimized for alignment and even fewer that seek to generate multiple structure alignments (superpositions and sequence alignments from more than two structures; see, for example, Russell & Barton, 1992; Taylor *et al.*, 1994; Shatsky *et al.*, 2004.)

Although structure comparisons provide the most structurally and functionally reliable alignment of protein sequences, one has to take care in interpreting such alignments. This is illustrated for the 27 SH2 domains structurally superimposed using *STAMP* (Russell & Barton, 1992) in Fig. 1 and the corresponding sequence alignment in Fig. 2.

The SH2 domain consists of a core β-sheet with helices on either side. As can be seen from Fig. 1, this core structure is relatively well conserved across the 27 domains, so in this region of the structure the alignment of the sequences has some meaning. The amino acids are in a similar location in each structure and so are likely to perform similar structural roles in the different proteins. However, in the loops that connect the core secondary structures, the story is different. Even when the loops are of similar length there can be considerable variation in structure. Thus, a sequence alignment of the residues in the loop may make no structural sense. When the loops differ in length the situation is even worse, with even more radical structural differences apparent.

The message is that although proteins of similar sequence have similar structures, the similarity in structure will not always extend to every residue in the protein. As a consequence, when looking at sequence alignments or, as in molecular replacement, attempting to align a sequence to a structure, one has to take care not to be overenthusiastic about aligning every residue. Fortunately, for structural alignment at least, in addition to a multiple structure superposition and alignment, the *STAMP* algorithm (Russell & Barton, 1992) provides a numerical indication of the likely reliability or 'alignability' of each column in the sequence alignment. This is illustrated in Fig. 2 for the SH2 domains by the red bars above the alignment. Half of the positions in the alignment are shown as reliable (90/180). The unreliable positions are mostly those where there are significant insertions in one or more structures, but also some positions that do not correspond to insertions (*e.g.* 88–89). Clearly, the proportion of reliably aligned positions relates to the overall similarity between the structures. If smaller less divergent subsets of proteins are aligned, then a higher proportion of the structure will be alignable. Essentially, the more similar the structures are to each other, the higher proportion of the structure will be aligned reliably. This simple principle must be taken into account when aligning sequences in the absence of structural information.

3. How similar do sequences need to be for reliable alignment?

As discussed in the previous section, examination of structural alignments shows that the more similar the proteins, the larger the proportion of the structure that can be aligned. It follows rather obviously that if one only has sequences, then the more similar the sequences, the more reliable any alignment of those sequences is likely to be. This relationship between alignment reliability and sequence similarity has been quan-

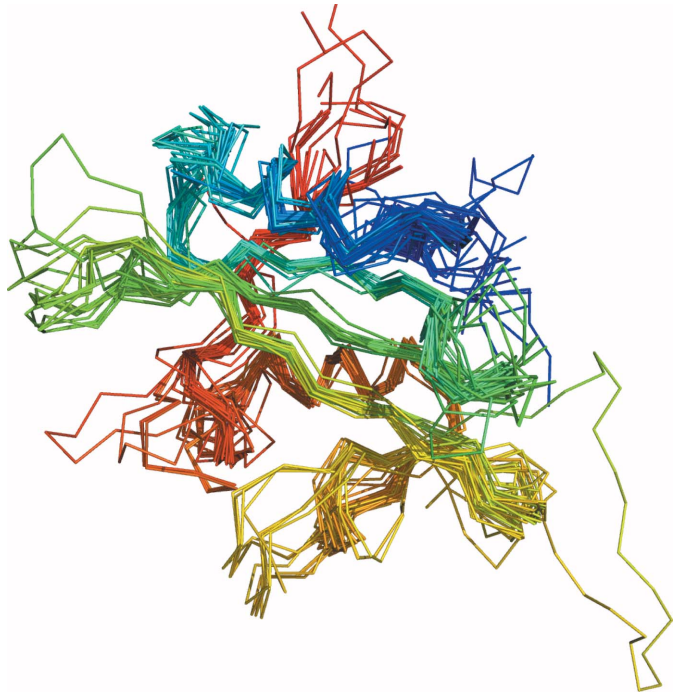

Figure 1
C_α representation of 27 SH2 domain structures aligned using the program *STAMP* (Russell & Barton, 1992). The image was prepared in *PyMOL* and coloured red to blue from the N- to C-termini. The corresponding structure-based sequence alignment is shown in Fig. 2.

tified in a number of papers (*e.g.* Barton & Sternberg, 1987*a*; Boscott *et al.*, 1993; Raghava *et al.*, 2003).

3.1. Percentage identity: pitfalls

'Percentage identity' (PID) is often quoted for the alignment of two protein sequences. It is an apparently simple measure of similarity and scales of confidence in alignment or structural similarity have been developed based on this measure (*e.g.* Sander & Schneider, 1991; Rost, 1999). The simplicity of percentage identity is a strength of the measure, but care has to be taken in comparing PID values generated by different software since the value of PID for a given alignment is dependent on how the PID is calculated (Raghava & Barton, 2006). The numerator is always the number of identical amino acids aligned, but there are at least five different denominators described in the literature. Variations include dividing by the length of the shortest sequence, by the number of aligned positions and by the average of the two sequence lengths. Percentage identity is further complicated by the fact

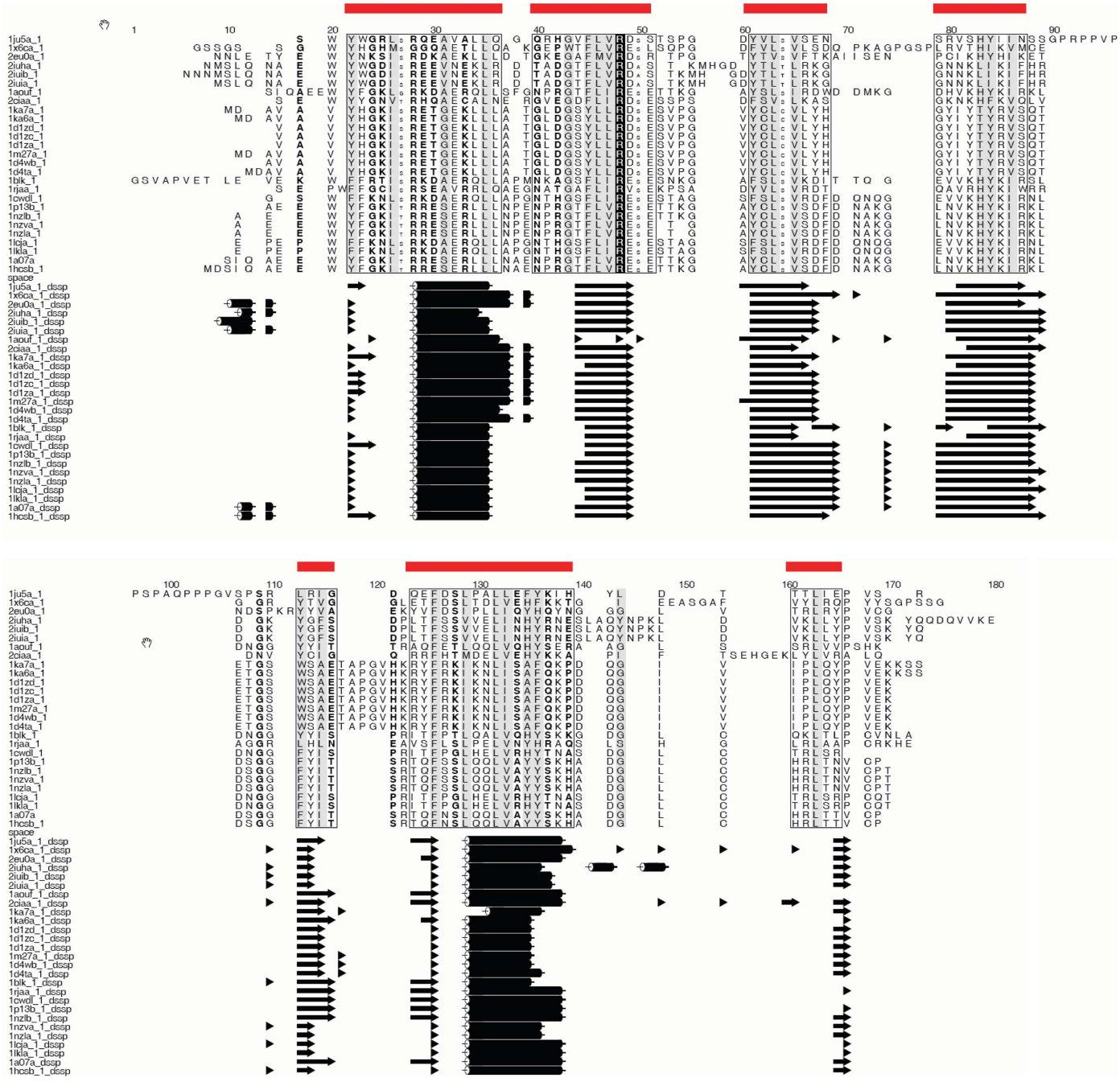

Figure 2
Multiple sequence alignment of 27 SH2 domains produced by *STAMP* (Russell & Barton, 1992) with the superposition shown in Fig. 1. Regions considered by *STAMP* to be aligned reliably are indicated by red bars over the sequence. The alignment was displayed using *ALSCRIPT* (Barton, 1993).

that it is strongly length-dependent, as illustrated in Fig. 3. Furthermore, if parameters in one alignment program are altered or different alignment programs are applied to the same pair of sequences then quite different PID values may result. In a recent study, a variation of up to 11.5% was observed for different PID calculations which increased to 22% when combined with different algorithms (Raghava & Barton, 2006).

3.2. *Z* scores as a measure of alignment reliability

Percentage identity is useful, but it is not usually the best measure to use to determine if two sequences will align well. The statistical Z score is more complex to calculate, but corrects to some extent for length and compositional biases in the sequences. Z scores are calculated by aligning the two sequences, recording the score S for the alignment and then shuffling the amino-acid order in one or both sequences and re-aligning. The shuffling and re-alignment is typically repeated 100 times and the Z score is then expressed as $(\overline{x} - S)/\sigma$, where $\overline{x}$ and σ are the mean and standard deviation of the shuffled sequence-alignment scores. The Z scores that result cannot be translated into probabilities through standard probability tables since the alignment-score distributions are not normally distributed. Despite this, a mapping to probabilities has been performed by modelling the Z-score distribution (Webber & Barton, 2001). Evaluation of alignments against reference structural alignments (Barton & Sternberg, 1987*a*) and correspondence in secondary structure (Boscott *et al.*, 1993) suggest that alignments that have a Z score of 6 or more will be accurate over most of their core secondary structures. It is also clear that as the Z score increases the alignment accuracy also increases, but below 6σ the accuracy is unpredictable (Boscott *et al.*, 1993).

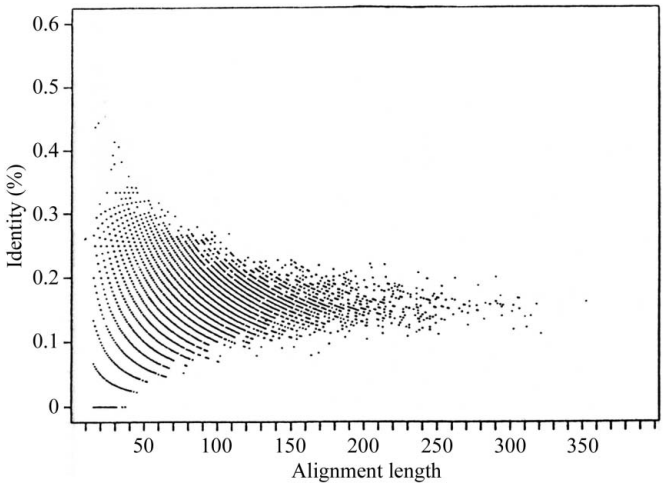

Figure 3
The relationship between the alignment length and percentage sequence identity (calculated as number of identities divided by the length of the shortest sequence) for alignments between protein pairs that are known not to have similar folds.

4. The basics of sequence alignment

The basics of how protein sequences are aligned pairwise or multiply have been discussed in detail in a previous CCP4 Study Weekend article (Barton, 1998); please see that article for details and appropriate references. Here, I give a brief summary of alignment methods before concentrating on the specific benefits of exploiting the evolutionary information present in multiple sequences.

4.1. Scoring amino-acid alignment

To align two sequences one first needs some scoring scheme that rewards the alignment of amino acids with similar properties. The simplest scheme just scores +1 for identity and 0 for mismatches; however, virtually all practical software exploits a symmetrical 20×20 table that gives a score for every possible amino-acid pairing. Such tables are normally expressed as log-likelihood ratios and so range either side of zero, with negative numbers representing amino-acid substitutions that are less likely to occur than by chance alone (*e.g.* Asp/Leu) and positive numbers representing those substitutions that are more common (*e.g.* Arg/Lys).

4.2. Scoring insertions and deletions

Since we know that amino acids may be inserted or deleted during the course of evolution, a score is also needed for positions in the alignment where amino acids in one sequence are not aligned with an amino acid in the other sequence. The absence of an amino acid at an alignment position is called a 'gap' and the score for gaps is known as a 'gap-penalty'. Gap-penalties typically have the form $ul + v$, where u and v are constants and l is the length of the gap. Thus, there is a cost associated with creating the gap and one for extending it.

4.3. Finding the best alignment

Given the scoring matrix and gap-penalty, the problem is to find the alignment of the sequences that gives the maximum score. This is normally performed by a dynamic programming algorithm as explained in Barton (1998). Dynamic programming requires MN steps, where M and N are the lengths of the two sequences, to find the best score for the comparison of the two sequences and also an alignment. Most alignment programs that implement dynamic programming only return a single alignment. However, there may be more than one valid alignment with the same score. It is important to bear in mind that even if there are no alignments with the same score, there will usually be many different alignments that have scores that are very close to the best score. These alignments may all be equally valid or at least no less incorrect than the alignment with the best score. Typically, as sequences that are less similar are aligned there will be more possible alternative alignments with scores similar to the best. Differences in alignment are concentrated around regions of the sequences where there is lowest sequence similarity, in particular around insertions/deletions.

4.4. Finding trustworthy parts of a pairwise sequence alignment

Although the 6σ cutoff provides a guide to the likely overall accuracy of a sequence alignment, it tells you nothing about which parts of the alignment are correct. One way to obtain a guide to which positions are trustworthy is to use an alignment program that highlights regions where alternative alignments are possible. Since the alignment in these regions is hard to define, the reliability of the alignment in these regions is likely to be lower than in other parts of the structure. Unfortunately, few, if any, commonly used alignment programs report such regions. An alternative approach is to to perturb any adjustable parameters that the program has, normally the gap-penalty and type of pair-score matrix, and record those regions of the alignment that change. Again, these are likely to be the less reliable parts of the alignment. A further method is to align with more than one program and then again examine the regions of the alignment that are least stable.

Although all these approaches to identifying reliable regions of an alignment of two sequences are effective, they are not straightforward to perform quickly. Fortunately, there is an easier way to assess confidence in an alignment by exploiting multiple sequences as explained in the following sections.

5. The benefits of multiple sequence alignment

For molecular replacement, one is typically only interested in the alignment of two sequences, the Query and a Target of known three-dimensional structure. However, even in this situation, if more than two sequences exist for the protein family it is better to perform a multiple sequence alignment. On average, multiple alignments give higher accuracy alignments than pairwise alignments of the same sequences. This is because when multiple sequences are aligned residues that are conserved through evolution for structural or functional reasons are highlighted within the alignment profile, as explained in the next section. For example, the hydrophobic patterns characteristic of core secondary-structure elements are aligned more accurately. Fig. 4 illustrates the difference between multiple and pairwise alignment accuracy for a large number of protein pairs. The mean difference is shifted to the right of zero, with the majority of sequence pairs more accurately aligned when part of a multiple alignment compared with when they are in a pairwise alignment. However, not all sequence pairs improve in accuracy on multiple alignment. The inclusion of divergent sequences in the multiple alignment is one reason for this. Accordingly, it is best first to check that all the sequences to be multiply aligned cluster with Z scores above 5–6σ.

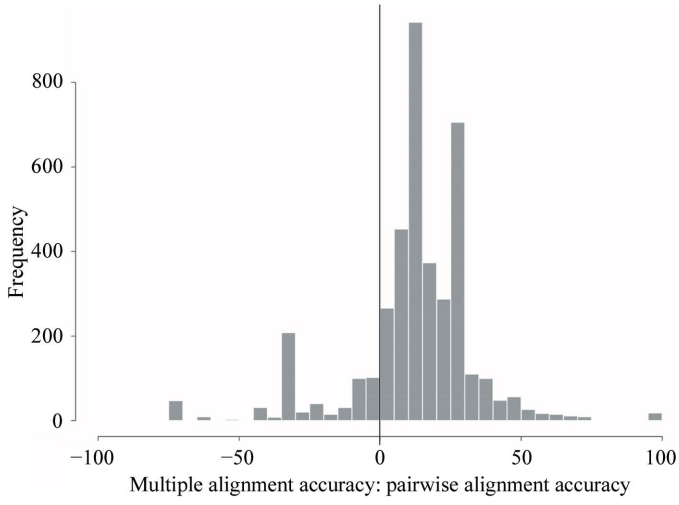

Figure 4
The difference between pairwise and multiple alignment accuracy for sequence alignments of protein families from the Oxbench benchmark suite (Raghava *et al.*, 2003).

Current benchmarks (Raghava *et al.*, 2003) suggest that the best methods for multiple sequence alignment include the comparatively recent programs *T-coffee* (Notredame *et al.*, 2000), *PROBCONS* (Do *et al.*, 2005) and *MAFFT* (Katoh *et al.*, 2005). However, for proteins that show sufficient sequence similarity to align reliably over most of their length, there is little difference between these methods and older multiple alignment methods such as *ClustalW* (Thompson *et al.*, 1994) and *AMPS* (Barton & Sternberg, 1987*b*; Barton, 1990) as all perform very well (Raghava *et al.*, 2003). In practice, it is best to try multiple methods on your sequences and, as for pairwise methods, use the differences between the resulting alignments to judge which regions of the alignment are likely to be unreliable.

5.1. Profile alignment

The majority of multiple sequence-alignment methods work hierarchically by first organizing the sequences to be aligned on a tree (calculated by pairwise alignment and hierarchical clustering or similar methods) and then working up the tree, aligning either two single sequences, a sequence to an alignment or two alignments. This process is explained in more detail in Barton (1998). The essence is that once an alignment has been generated it remains fixed. The alignment is represented by a profile of numbers that code the amino-acid frequencies at each position in the alignment and the frequency of occurrence of gaps. The exact way in which the profile is generated depends on the alignment algorithm and can vary from simple averages (Barton & Sternberg, 1987*b*) to an explicit probabilistic representation (Do *et al.*, 2005). Most methods of generating a profile do not use raw amino-acid frequencies, but normalize frequencies by local or global amino-acid composition. General pair-score matrices such as *BLOSUM* (Henikoff & Henikoff, 1992) are used to compensate where there are few observed substitutions from the alignment that the profile is derived from.

Although profile alignment and profile–profile alignment techniques are central to multiple sequence alignment, they are also often used to align an alignment to one or more single sequences or to a library of alignments. The most common technique currently used for this is the profile HMM (hidden Markov model). HMMs are one of the most sophisticated

methods of capturing the information present in a multiple sequence alignment. They encode the probability of observing each amino-acid type at a position as well as the transition between amino-acid types at successive positions and between amino acids and gaps. As a consequence of this sophistication and a clean statistical framework, many libraries of Profile HMMs for common protein families and domain families have been constructed. The best known of these is Pfam (Bateman *et al.*, 2004), but others exist for more detailed coverage of specific superfamilies (*e.g.* protein kinases; Miranda-Saavedra & Barton, 2007). A commonly used profile HMM program suite is *HMMER*. Techniques for profile–profile HMM alignment are less well developed, but include the program *PRC* that underpins the Superfamily database (Wilson *et al.*, 2007).

5.2. Incorporating structural information into alignment

When aligning the query to a protein of known structure, the additional information present in the structure can be used to help improve alignment. While this can be performed by hand as explained in the next section, a number of techniques aim to incorporate structural information into alignment. Since insertions and deletions tend to occur in surface-loop regions, one of the earliest ideas was to bias the gap-penalty to penalize gaps in core secondary structures more heavily than those in loops (Barton & Sternberg, 1987*a*). Recently, the *T-coffee* algorithm has been extended to incorporate information from structural and structure-sequence alignments. *3D-coffee* (O'Sullivan *et al.*, 2004) provides a convenient way to combine reliable structural alignments with sequence alignments in a single multiple alignment and claims significant accuracy improvements (O'Sullivan *et al.*, 2004).

During the 1990s there was considerable effort in developing 'threading' 'fold-recognition' methods that in their most sophisticated forms (*e.g.* Jones *et al.*, 1992) used a statistical pair-potential to judge how well a sequence would fit to a particular fold. Alternative fold-recognition strategies extended sequence-alignment algorithms to include additional information from the structure through extended structure-based substitution matrices (Overington *et al.*, 1992; Kelley *et al.*, 2000). The majority of these methods are aimed at detecting similarity in structure when the signal in the sequence is very weak, rather than optimizing alignment accuracy. For most sequences that are not highly divergent

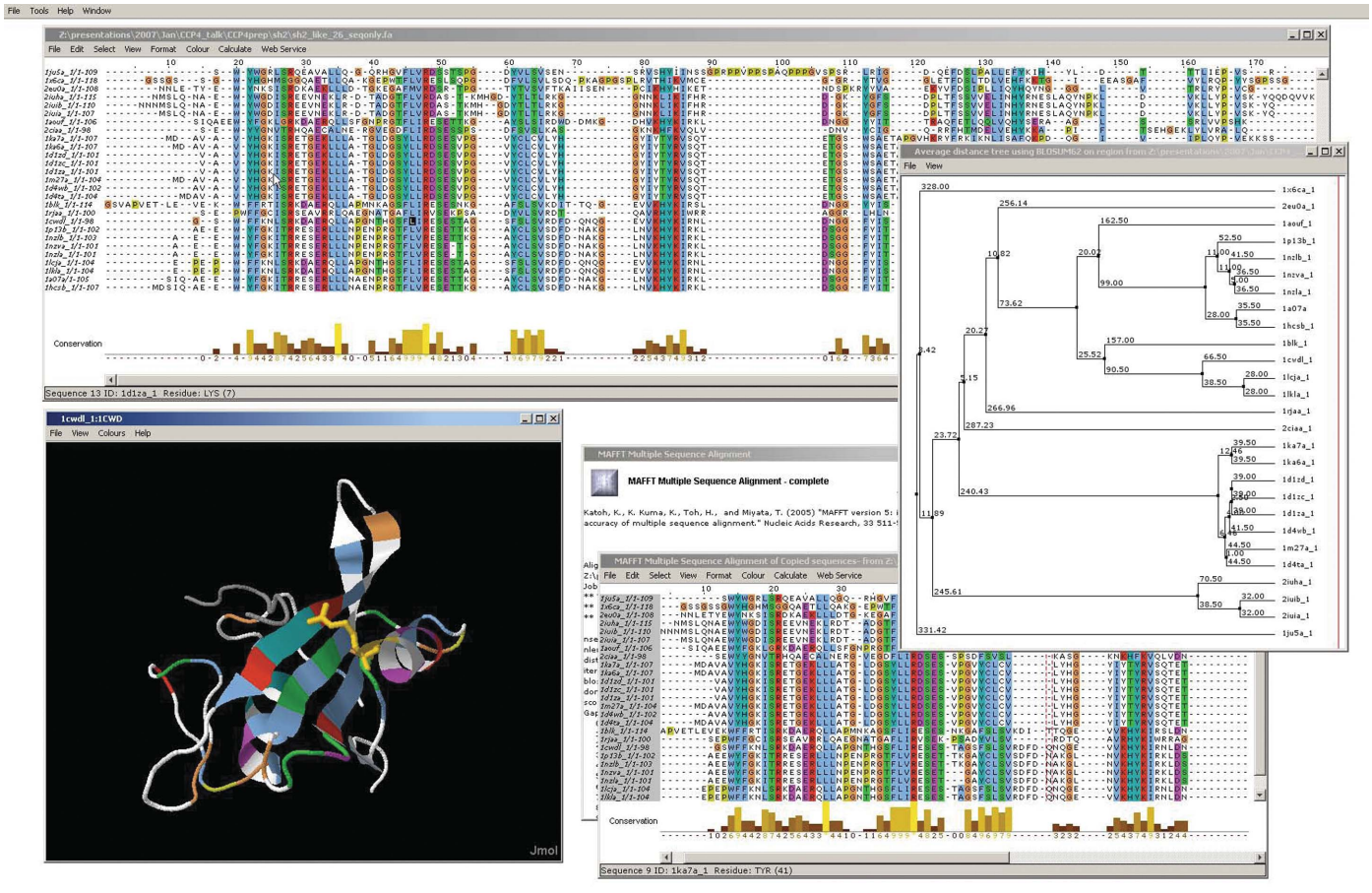

Figure 5
Screenshot of the *Jalview* (http://www.jalview.org; Clamp *et al.*, 2004) multiple alignment editor and workbench. The figure illustrates a *Jalview* session with the structural multiple alignment from Fig. 2, a simple tree derived from the alignment and a structure for one of the domains displayed with the in-built *Jmol* application and coloured in the same way as the alignment. Mousing over a residue in the alignment view has highlighted the Arg residue in the *Jmol* view. Sequence alignments may be generated directly from *Jalview* as illustrated in the lower right of the screen for an alignment of the same sequences generated by *MAFFT* (Katoh *et al.*, 2005).

from the structural target and so are most likely to be useful in molecular replacement, there are unlikely to be significant benefits in employing threading methods. Indeed, for threading methods optimized to detect remote structural similarity, the quality of alignment may be worse than that possible with a well optimized multiple sequence alignment.

5.3. Inspection and optimization of multiple alignment by hand

Since the vast majority of multiple alignment methods are hierarchical, errors in alignment can be locked in early in the hierarchy and not corrected to take account of the sequences added later. Some methods exploit iteration (*e.g.* Barton & Sternberg, 1987*b*; Gotoh, 1993) in an attempt to mitigate such errors, but despite this mistakes will remain that can be apparent when the final alignment is inspected by eye.

The alignment generated by software should always be interpreted and assessed by reference to a structure if one is available. This process is greatly aided by a multiple alignment editor and workbench such as *Jalview* (http://www.jalview.org; Clamp *et al.*, 2004). *Jalview* incorporates a range of sophisticated alignment-editing and visualization tools as well as direct links to a number of multiple alignment methods [*e.g.* *MAFFT* (Katoh *et al.*, 2005) and *MUSCLE* (Edgar, 2004)] and annotation services such as secondary-structure prediction by *JPred*/*JNet* (Cuff *et al.*, 1998; Cuff & Barton, 2000). *Jalview* can also display protein three-dimensional structures when co-ordinates are available and link these to the alignment. Further functions include the automatic look-up of pre-calculated sequence features from a number of web-accessible resources such as Uniprot (Apweiler *et al.*, 2004) as well as the calculation and display of single-linkage and neighbour-joining trees from the alignment. Fig. 5 illustrates a typical screenshot of the current version of *Jalview*, which displays structures using the *Jmol* molecular-structure viewer.

When evaluating and optimizing the alignment, it is best to start by examining the alignment as a whole. Does it have gaps scattered all over it or are there clear blocks of aligned residues separated by regions that are more 'gappy'? If sequences that cluster above 5–6σ have been selected then the alignment should contain regions that are relatively gap-free. In *Jalview*, the 'conservation' line under the alignment indicates how well the physico-chemical properties of the amino acids are conserved in each column of the alignment. Clearly, regions where the properties are not well conserved are likely to be more variable in structure and/or incorrectly aligned.

Since at least one of the sequences under examination will have a known three-dimensional structure, this should be displayed alongside the alignment. The blocks of conserved regions should correspond to the core secondary structures and any other structural feature that is important to all the sequences in the alignment. The main problems involve the location of gaps. Look carefully at the alignment either side of a gap. Check if a shift of the sequence might bring more hydrophobic residues into register in the conserved blocks either side of a gap. However, since regions of an alignment where there are gaps correspond to parts of the protein that are structurally variable, it is not worth spending a lot of time agonizing over the precise alignment of amino acids within gappy regions. Rather, it is better to preserve any alternative alignments that seem equally reasonable in gappy regions and so build multiple alternative models to take forward as MR search objects.

6. Summary guidelines for sequence alignment

There is no single 'right way' to align a sequence to a structure and be assured of the most accurate alignment possible. The best approach will depend on the diversity in the protein sequence family under study, the availability of structures of members of the family and where the target protein lies in similarity compared with all members of the family. Some sequence families have relatively small variations in loop length and conformational changes in the conserved core. Others are much more variable. Some proteins undergo major conformational changes during function, while others do not. Each of these family and protein-specific differences make general rules difficult for alignment. Despite these problems, the following guidelines should help as a starting point.

(i) Consult Pfam, Superfamily and similar alignment libraries for similarities to the query protein.

(ii) Build a structural multiple alignment for available structures.

(iii) Build an HMM from the alignment (*e.g.* using *HMMER*) and search for further sequences similar to the family.

(iv) Build an HMM from the combined sequence and structure alignment. Also consider *3D-Coffee* (O'Sullivan *et al.*, 2004) for this step.

(v) Align the query sequence to the resulting HMM.

(vi) Inspect alignment(s) in *Jalview*, paying attention to regions of the alignment that are likely to be less reliable.

An alternative to the use of HMMs would be to gather sequences similar to the query by any search method (*e.g.* PSI-BLAST) and then perform a multiple sequence alignment by a standard MSA program as discussed above. This strategy will also work well if you have only one structure that could serve as a template for modelling.

I thank all colleagues who have contributed to the work from my group that is mentioned in this review as well as the BBSRC, MRC and Royal Society for their support.

References

Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M. J., Natale, D. A., O'Donovan, C., Redaschi, N. & Yeh, L. S. (2004). *Nucleic Acids Res.* **32**, D115–D119.
Barton, G. J. (1990). *Methods Enzymol.* **183**, 403–428.
Barton, G. J. (1993). *Protein Eng.* **6**, 37–40.
Barton, G. J. (1998). *Acta Cryst.* D**54**, 1139–1146.
Barton, G. J. & Sternberg, M. J. (1987*a*). *Protein Eng.* **1**, 89–94.
Barton, G. J. & Sternberg, M. J. (1987*b*). *J. Mol. Biol.* **198**, 327–337.

Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L., Studholme, D. J., Yeats, C. & Eddy, S. R. (2004). *Nucleic Acids Res.* **32**, D138–D141.
Boscott, P. E., Barton, G. J. & Richards, W. G. (1993). *Protein Eng.* **6**, 261–266.
Clamp, M., Cuff, J., Searle, S. M. & Barton, G. J. (2004). *Bioinformatics*, **20**, 426–427.
Cuff, J. A. & Barton, G. J. (2000). *Proteins*, **40**, 502–511.
Cuff, J. A., Clamp, M. E., Siddiqui, A. S., Finlay, M. & Barton, G. J. (1998). *Bioinformatics*, **14**, 892–893.
Do, C. B., Mahabhashyam, M. S., Brudno, M. & Batzoglou, S. (2005). *Genome Res*, **15**, 330–340.
Edgar, R. C. (2004). *Nucleic Acids Res.* **32**, 1792–1797.
Gotoh, O. (1993). *Comput. Appl. Biosci.* **9**, 361–370.
Henikoff, S. & Henikoff, J. G. (1992). *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.
Jones, D. T., Taylor, W. R. & Thornton, J. M. (1992). *Nature (London)*, **358**, 86–89.
Katoh, K., Kuma, K., Toh, H. & Miyata, T. (2005). *Nucleic Acids Res.* **33**, 511–518.
Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. (2000). *J. Mol. Biol.* **299**, 499–520.
Miranda-Saavedra, D. & Barton, G. J. (2007). *Proteins*, **68**, 893–914.
Notredame, C., Higgins, D. G. & Heringa, J. (2000). *J. Mol. Biol.* **302**, 205–217.
Novotny, M., Madsen, D. & Kleywegt, G. J. (2004). *Proteins*, **54**, 260–270.
O'Sullivan, O., Suhre, K., Abergel, C., Higgins, D. G. & Notredame, C. (2004). *J. Mol. Biol.* **340**, 385–395.
Overington, J., Donnelly, D., Johnson, M. S., Sali, A. & Blundell, T. L. (1992). *Protein Sci.* **1**, 216–226.
Raghava, G. P. & Barton, G. J. (2006). *BMC Bioinformatics*, **7**, 415.
Raghava, G. P., Searle, S. M., Audley, P. C., Barber, J. D. & Barton, G. J. (2003). *BMC Bioinformatics*, **4**, 47.
Rost, B. (1999). *Protein Eng.* **12**, 85–94.
Russell, R. B. & Barton, G. J. (1992). *Proteins*, **14**, 309–323.
Sander, C. & Schneider, R. (1991). *Proteins*, **9**, 56–68.
Shatsky, M., Nussinov, R. & Wolfson, H. J. (2004). *Proteins*, **56**, 143–156.
Taylor, W. R., Flores, T. P. & Orengo, C. A. (1994). *Protein Sci.* **3**, 1858–1870.
Thompson, J. D., Higgins, D. G. & Gibson, T. J. (1994). *Nucleic Acids Res.* **22**, 4673–4680.
Webber, C. & Barton, G. J. (2001). *Bioinformatics*, **17**, 1158–1167.
Wilson, D., Madera, M., Vogel, C., Chothia, C. & Gough, J. (2007). *Nucleic Acids Res.* **35**, D308–D313.