Acta Crystallographica Section D
Biological Crystallography
ISSN 0907-4449

CAALIGN: a program for pairwise and multiple protein-structure alignment

T. J. Oldfield‡

Accelrys Inc., 10188 Telesis Court, Suite 100, San Diego, CA 92121, USA

‡ Current affiliation: European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, England.

Correspondence e-mail: oldfield@ebi.ac.uk

Received 5 July 2006
Accepted 8 January 2007

Coordinate superposition of proteins provides a structural basis to protein similarity and therefore complements the technique of sequence alignment. Methods that carry out structure alignment are faced with the problem of the large number of trials necessary to determine the optimal alignment solution. This article presents a method of carrying out rapid (subsecond) protein-structure alignment between pairs of proteins based on a maximal C^{α}-atom superposition. The algorithm can return alignments of 12 or more residues in length as multiple non-overlapping solutions of alignment between a pair of proteins which are independent of the fold connectivity and secondary-structure content. The algorithm is equally effective for all protein fold types and can align proteins containing no secondary-structure elements such as is the case when searching for common turn structures in proteins. It has high sensitivity and returns the set of true positive results before any false positives as judged by SCOP classification. It can find alignments between topologically different folds and returns information about sequence alignment based on structure alignment. Additionally, this algorithm has been extended to carry out multiple structure alignment to determine common structures within groups of proteins, including the nondegenerate set of proteins in the PDB. The algorithm has been implemented within the program *CAALIGN* and this article presents results from pairwise structure alignment, multiple structure alignment and the generation of common structure fragments found within the PDB using multiple structure alignment.

1. Introduction

The three-dimensional structure of proteins is encoded by a one-dimensional genome and the function of proteins is determined by the three-dimensional interaction of residues localized within a small volume. The classification of the sequence and structural information is based on the ability to recognize similarity between the different sequences and different coordinates. It is apparent that the evolution of sequence is much more rapid than the evolution of structure for a particular protein function. This means that the search for protein similarity to recognize function and homology/analogy within sequence space requires the analysis of information that is often at the limit of the signal to noise. In general, it is difficult to observe evolutionary context when the sequence similarity is below 30%. The use of structural similarity to determine function and classification is therefore desirable.

The rapid comparison of protein structures is useful in many areas of research. The classification of structures by fold is essentially a problem of determining the structural similarity between pairs of proteins. Multiple structure alignment (MSA) takes this further by identifying common fold and packing features within a family of proteins; when used on a nondegenerate list of proteins, it can identify common motifs recurrent throughout protein-fold space. Structure alignment (SA) also allows comparative analysis of function within homologous and analogous proteins and hence of the evolutionary relationships within protein families. Additionally, the characterization of equivalent residues, particularly around an active site, can often suggest the reaction mechanism and function.

SA is not theoretically difficult, but presents a technical challenge because the search space is so large. In general, it is necessary to determine both the position and length of alignment within a pair of structures that have equivalent residues. The alignment solution may be independent of the sequential order of the residues and can have multiple insertions and deletions between aligned regions that are not necessarily topologically related within the sequence. Previously, the problem has been approached in a number of ways. Methods have been implemented that are based on the use of secondary-structure elements (SSEs) to reduce the size of search space, such as vector alignment (Gibrat *et al.*, 1996) and the use of clustering (Vriend & Sander, 1991; Mizuguchi & Go, 1995; Oldfield, 1992), depth-first recursive search (Kleywegt & Jones, 1997) and graph theory (Mitchell *et al.*, 1990; Alexandrov, 1996; Grindley *et al.*, 1993; Krissinel & Henrick, 2004). Methods also exist based on residue alignment (using coordinates) that use techniques such as dynamic programming of distance matrices (Orengo & Taylor, 1996; Subbiah *et al.*, 1993), Monte Carlo simulations (Holm & Sander, 1993) and combinatorial extension (Shindyalov & Bourne, 1998). Additionally, the use of human recognition of fold forms the basis of the SCOP classification (Murzin *et al.*, 1995). An important issue with regard to structure alignment is the ever-increasing number of proteins within the PDB (Berman *et al.*, 2000). Even without the potential products of structure genomics projects, there has been an exponential growth in protein structure submissions to the PDB. Therefore, any method of structure alignment must be fast and scalable with regard to the number of protein structures and the size of any one single entry.

The algorithm presented here is independent of the topology of the protein, allowing the identification of snippets of similar protein structure that are independent of the framework and connectivity of the surrounding protein. The program returns a number of possible different outputs: the superposed coordinates, the transformation matrix, the residue-mapping arrays, the sequence alignment determined from the structure superposition, the percentage of residue identity, the length of alignment, score, r.m.s.d. and whether the proteins have the same sequence connectivity. The information output by the program is repeated for multiple occurrences of non-overlapping solutions between a pair of proteins. The pairwise alignment program is designed to carry out one-against-many alignments or many-against-many to form a square symmetric matrix of relationships.

An extension to the basic pairwise alignment from *CAALIGN* is MSA. Analogous to multiple sequence alignment, the exact solution to MSA is nontrivial as it requires an additional mathematical dimension of analysis for each additional structure to be aligned. A general method to determine an exact solution is possible, but would be impractical based on time and memory constraints. Unlike SA, there are few methods implemented for MSA and these extend existing SA algorithms (Gerstein & Levitt, 1996; Orengo & Taylor, 1996; Shindyalov & Bourne, 1998; Sali & Blundell, 1990; Guda *et al.*, 2001; Leibowitz *et al.*, 2001; Krissinel & Henrick, 2005). The method of Leibowitz and coworkers looks for common geometric substructures and extends from these cores; the other methods are extensions to all-to-all pairwise alignment. The approach taken here is based on common cores and a linkage combination of the pairwise results. It does not assume that any one structure is an ideal solution to the MSA, as it returns a weighted mean set of coordinates as the best family alignment between proteins. The program not only returns detail on the MSA, but also returns the multiple sequence alignment based on structure alignment for each cluster of aligned structures where the structure alignment is topologically equivalent to the sequence.

2. Methods

2.1. Pairwise alignment

2.1.1. Seed-point analysis. For this alignment method, the global alignment solution can only be guaranteed to be found if we try all combinations of C^{α}-atom mappings for two structures; however, this is not practical owing to the large number of combinations possible for the atomic coordinates. The analysis can be simplified using the small number of

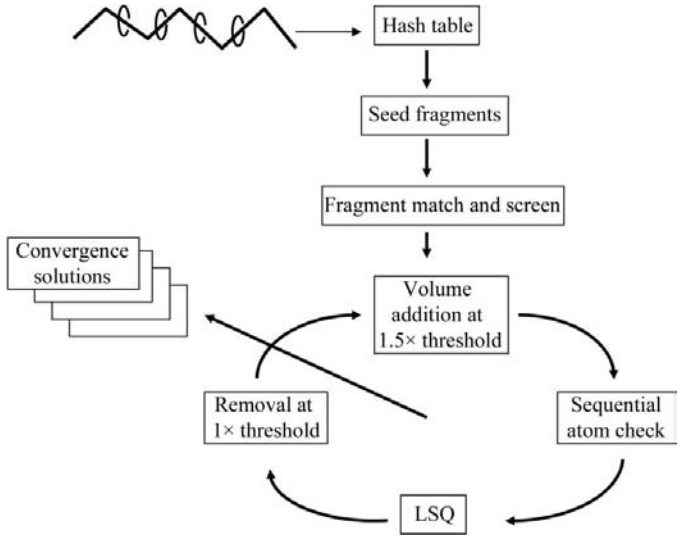

Figure 1
Overall design of superposition method.

secondary-structure vectors, but this only works with proteins that have definite and similar secondary structures. The *CAALIGN* program solves the combinatorial problem of C^{α}-atom alignment by searching for a filtered set of seed points before proceeding to a full alignment analysis. It is necessary that the generation of the filtered set has a low false-negative rate in order to avoid missing solutions and a low false-positive rate to reduce calculation times. This provides the speed of alignment associated with SSE alignment, but allows superposition independent of any local fold type. The design of the structure-alignment algorithm is shown in Fig. 1.

2.1.2. Hashing. Seed-point analysis involves finding reasonable starting points for further analysis. It should be fast and not be swamped by special properties associated with protein structure. The algorithm used is based on C^{α} pseudo-torsion angles (Oldfield & Hubbard, 1994) combined into 'words' and rearranged as a hash table. Each pseudo-torsion angle defines a local section of fold based on the relative orientation of four residues and summarizes the combination of multiple φ, ψ and ω angles into a single parameter (Oldfield & Hubbard, 1994; Oldfield, in preparation). Combining multiple values using a power series expands this so that a single unique data value represents the conformation of a 'word' of multiple residues. Finally, if the data are rearranged into a hash table, then we place references to all sections of protein polypeptide chain with the same conformation into a single data object, allowing efficient comparison and searching. Word hashing methods using multiple letters of a protein/nucleic acid sequence have been used in sequence alignment and these are termed k-tuple methods; for example, *FASTA* (Pearson & Lipman, 1988) and *BLAST* (Altschul *et al.*, 1990). For a word size of N C^{α} atoms, it is possible to define $(N-3)$ C^{α} torsion values, which are combined to give a single hash value defined in (1), which is the sum of a power series.

$$\text{hash} = \sum_{i=1}^{N-3} \left(\frac{T_i}{P}\right) \times \left(\frac{360}{P}\right)^{i}, \qquad (1)$$

where T_i is torsion angle (i) in the analysis word in degrees, P is the torsion bin precision and N is the number of C^{α} atoms in a word.

All values are user-defined. For all the results presented in this paper, the analyses used $N = 6$ and $P = 20^{\circ}$. A single hash table was calculated, to which all hash values from all proteins were added with pointer references back to the protein and the sequence position of origin. A seed point is simply any two references with the same hash value that are from different proteins. Note that a range value, the number of neighbouring hash bins included in a calculation, can be set to larger than zero in order to handle edge effects when using hashing algorithms. A range value of zero was used for this analysis.

As with all measures of relative conformation, hashing produces a high concentration of one value, which represents the α-helix. This is because the geometry of a helix has low variance and most of these are false-positive seed points. To solve this problem, a statistical observation was put into effect computationally. Alignment of helical proteins invariably results in superposition of at least one end of the SSE. Because of this, it is not necessary to use the helical hash-bin data for seed points. The algorithm therefore contains an optional pass-through for all hash values and word sizes that define the helix conformation. No difference was found when comparing results both using and not using this pass-through, except for analysis time. It should be noted that β-strands have variable structure so there is no issue with hash-bin saturation.

2.1.3. Volume expansion. The seed-point analysis provides a continuous trace with at least six C^{α} atoms, although this minimum length can be adjusted using the hash word size. Alignment optimization is required to add further C^{α} atoms to this collection through space; that is, to add equivalent pairs of C^{α} atoms within a volume about the seed-point trace in order to increase the total number of C^{α} atoms within the alignment without exceeding the r.m.s.d. limit. The two proteins to be aligned are superposed based on the seed-point atoms by a least-squares algorithm (Kabsch, 1976) with modifications (Oldfield, 2002); atom pairs are rejected if they become separated by this process and new pairs are added if they become close. The cycle is repeated up to five times using a user-defined r.m.s.d. threshold, where each cycle consists of a least-squares alignment, an atom-addition/removal step and a continuity check (Fig. 1).

2.1.4. Atom rejection and addition. The atom-rejection routine checks all current atom-pair mappings after a superposition cycle and removes those where the separation is more than the r.m.s.d. limit. New atom pairs are added to the alignment if their separation is less than the 1.5 times the r.m.s.d. limit. This process stops the alignment process when no additional atoms are added with respect to the previous cycle or continues for a maximum of five cycles.

2.1.5. Continuity check. The continuity check is an important aspect of the alignment expansion. This is because a pair of proteins yet to be aligned properly can have pairs of atoms throughout the protein pair that are incorrectly marked as equivalent in a seemingly random fashion. This is particularly the case for atoms that are some distance from the seed point, because the separation between a pair of correctly mapped atoms is directly proportional to their distance from the seed point. Incorrectly mapped atom pairs will tie down the alignment optimization within a false minimum. The C^{α} equivalence between a pair of structures also becomes ambiguous when the r.m.s.d. limit on a particular atom position exceeds half the separation distance between two C^{α} atoms. That is, the algorithm can miss equivalent C^{α} atoms within a polypeptide chain in one protein sequence owing to variance in the position of the atoms determined by experiment. The continuity check looks for sequential atoms and reinforces the rule that consecutive atoms should be mapped as equivalent between the two proteins. Any fragment of superposed atoms of length five atoms or less is removed from the superposition list. The chain direction of the mapped atoms is not prescribed within the algorithm, allowing reversed fragment alignment. Any divergent structure at the end of aligned structure is automatically trimmed off by this criterion. Therefore, the returned r.m.s.d. is always less than or

equal to the user-defined r.m.s.d. limit and is generally observed to be much smaller.

2.2. Identical solution analysis

The alignment program is designed to return either (i) the best solution, (ii) all possible non-overlapping solutions or (iii) all solutions between a pair of proteins. The best solution is defined here as the result with the best target score. The algorithm presented returns multiple solutions starting at each seed point and will therefore generate multiple degenerate solutions. If the best solution is required, then the current target is tested to detect whether a new solution is better than any known current solutions and if so it replaces the known solution. Where the set of nondegenerate solutions is required, the residue equivalence between the new and current result list is determined and if more than ten residues, the shorter is rejected. Some overlap is tolerated, as solutions are observed where two domains align separately but not together (for example, owing to a hinge-region change) but a common region appears in both alignments. The analysis is carried out by checking the alignment mapping array of the new solution with the list of previous solutions; if there are less than ten common residues then the new solution is appended to the hit list, otherwise the shorter solution is deleted. If all the solutions are required, then a new solution is just added to the current list of known solutions.

2.3. Target functions

To optimize the superposition of a protein, it is necessary to define a target to minimize. Protein SA requires the optimization of similarity (such as r.m.s.d.) and superposition volume (such as the number of C^{α} atoms). There are four target functions implemented within the program. The first uses alignment length as a threshold, the second is based on alignment length as a percentage of the number of residues within the target structure and the third is the logical AND of the two targets. The fourth method uses implementation of the CE score (Jai *et al.*, 2004) defined as

$$\text{Score} = (\text{r.m.s.d.}/\text{length}) \times (1 + \text{gaps}/\text{length}). \qquad (2)$$

In all the results presented below, the CE score was the chosen target function with a threshold set to 0.05. The CE score is zero for an exact alignment and is proportional to the r.m.s.d. and the number of gaps in the alignment and inversely proportional to the alignment length.

2.4. Program output

For each pairwise alignment, the program can write the following results.

(i) An r.m.s.d. determined from the matched C^{α} atoms.

(ii) The number of matched C^{α} atoms.

(iii) CE score value.

(iv) Topological analysis of the chain mapping.

(v) Sequence identity as a percentage within the structurally aligned region.

(vi) A sequence alignment based on the structure alignment (only if the mapping is sequential).

(vii) Two mapping arrays that define the C^{α} mapping between two structures.

(viii) The C^{α} coordinates of a working molecule transformed to the reference molecule for either all C^{α} atoms or just the aligned section.

(ix) A 4×4 matrix that will align the original PDB entry with the reference structure.

2.5. Multiple structure alignment

The basic alignment algorithm has been extended to allow MSA. Given a set of proteins, it determines subsets that form structural clusters within the user-defined target function. This is performed in cycles that progressively merge pairs of structures to create structure groups, finally creating clusters which are the solution to MSA. An outline of the algorithm design is shown in Fig. 2.

The first cycle of the MSA consists of calculating the pairwise alignment matrix for all against all from a user-defined list of proteins. This matrix is analysed to determine the list of closest pairs of structures that are better than the alignment threshold and, for each close pair of structures, to determine the weighted average coordinates of the matched C^{α} atoms. These averaged coordinates are propagated to the next cycle of analysis as a single object, here called a structure group. For the remaining structures that are not part of one of the initial aligned pairs, two options exist. The first option is that the single structure aligns with one of the structure groups below the threshold, but the closest partner is already taken; this means it is part of the structure group but not the closest member of that group. This structure is carried over to the next level of the search with half weight. The second option occurs if a structure does not align with any other structure group within the defined threshold and so does not have any partner at this level of analysis. This structure is eliminated from further analysis. The analysis is repeated using the averaged coordinates generated from a previous cycle and, in practice, the number of remaining structures is approximately halved by each cycle of the pairing analysis. The analysis is complete when no structure groups can propagate to another cycle of analysis. Each distinct structure group that is elimi-

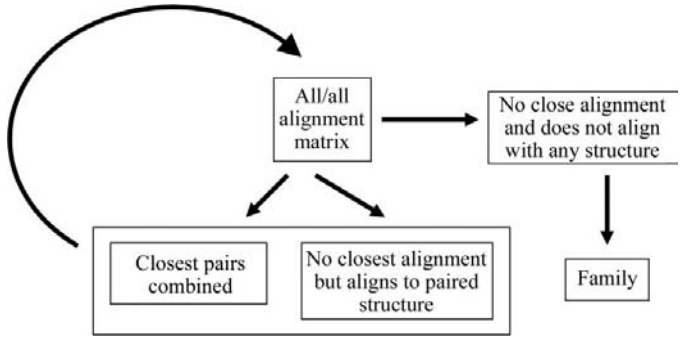

Figure 2
Overall design of multiple structure-alignment algorithm.

Table 1
Run-time parameters and the default values used for this analysis.

Parameter	Optimal value
Hash word size	6
Torsion bin size (°)	20
Maximum cycles of superposition	5
Minimum seed-chain length	Hash word size
Consecutive atom check	5
Search coverage	Global
Check for previous solution	Full mapping check

nated from a search cycle is termed a cluster and each cluster can be composed of a single structure or many structures.

Coordinate combination is performed with weighting based on the number of structures in the structure group. In this way, structure groups generated from many structures are highly weighted and have a known membership. Each cycle represents merging, using weights, of the inverted binary tree until the MSA is complete. The program will also return the structure-based sequence alignment for each cluster, as well as the average linkage relationship, with distances between members being based on the target function. The output coordinates will not be equal to any one original coordinate structure because none are likely to be an ideal solution to a cluster.

If MSA is performed on a family of proteins, there will be a single output cluster consisting of the average coordinates of all the members of the family. If the list of proteins contains a number of fold families, then the average coordinates for each family are split off when they become a separate cluster. If a list of nondegenerate proteins is aligned, then the output is a set of protein fragments, independent of sequence and connectivity, that are common within the list of proteins; for example, a β-sheet or helix bundle.

2.6. Assessment of selectivity and sensitivity

To determine the quality of the results from the program *CAALIGN*, an analysis was carried out analogous to the published assessment of Novotny *et al.* (2004), which describes a number of different tests performed by SA servers and ranks the results by selectivity and sensitivity; the full results for a number of alignment servers are presented at http://xray.bmc.uu.se/~marian/servers. The analysis is split into assessment by fold type (α, β, α–β, little secondary structure), difficult cases, multi-domain and NMR data. Each analysis in Novotny *et al.* (2004) was repeated with the program *CAALIGN* with the default set of parameters. Novotny *et al.* (2004) additionally describe service quality based on true positives and false positives, where a positive result is based on the classification by CATH topology (Orengo *et al.*, 1997) generated by v.2.0 and v.2.4. The *CAALIGN* program was run with a target CE score of 0.05 and an upper limit on C^{α} r.m.s.d. of 2.0 Å. [It should be noted that there is no current structure 1awg as quoted by Novotny *et al.* (2004) and no archive has a reference to this ID code. The cyclophilin structure 1awq (Vajdos *et al.*, 1997) was used in its stead. In addition, NMR model structure 20 is noted as the most divergent, although the website replaces model 20 with model 19. Analysis here was performed with both models 19 and 20.] All procedures described by Novotny and coworkers were repeated using the program *CAALIGN* and the results in this paper are based on calculations using the program on a 1 Ghz desktop computer against 28 522 structures of the May 2005 PDB archive.

2.7. Implementation

All code was written in the computer language C using hash tables for torsion hashing, structure lists to store fragments of alignments and a binary tree construct for the clustering. Alternate hashing functions using distances between residue centroids and C^{α} positions were not as sensitive as the torsion-angle hashing described here at discriminating between different local folds. These distance-based hashing methods resulted in more false-positive hits during the initial local structure screening, thus significantly slowing the overall alignment process. Results from these distance-based hashing algorithms are not presented. All memory allocation is dynamic and allocated on demand at run time. Run-time parameters are available to control the algorithm, such as hash bin size, hash word size and sequence-continuity check, but these have been optimized and the default values (Table 1) were used to obtain the results presented below. The r.m.s.d. and alignment-length targets are defined using run-time parameters, along with the amount of the output generated by the program. Settings are available to speed up the calculation, to use a global search or probable best solution and to perform simple alignment and different fragment overlap analysis.

All the analysis and times presented here are for the global solution searching and full overlap analysis; the latter is necessary for alignments where the sequence connectivities are not the same. The program is CPU-bound when used for pairwise and one-to-many calculations and generally results in a run-time size of less than 10 Mb. MSA uses significant amounts of memory because all alignment details must be retained during the process of combination.

3. Results

Four different kinds of analyses were carried out as a means of testing and proving the alignment program. The first was the simple pairwise alignment of two proteins, which is a one-to-one alignment. The second analysis was the extension to a one-to-many calculation. The program simply runs through a list of protein files and returns matches for a single structure to any of the files in the list. The third analysis (many-to-many) finds the family relationship within a list of proteins using MSA.

The fourth analysis was MSA of a list of unique protein structures prepared by application of a number of restriction criteria (Oldfield, 2001). In this context, a unique set is defined as a list of protein ID codes selected on the basis of structures with good geometry that differ by more than 20% in their sequence from all other proteins in the set. This last analysis

```
Success for Mols : (1mbd)-(4hhb) : length 136/153 : rms  1.394 : seq-homology 26.5

1mbd  .........VLSEGEWQLVLHVWAKVeaDVAGHGQDILIRLFKSHPETLEKFDRFkhLKTEAEMKASEDL
 420              |    |    | ||   |    |   | ||    | |   |   | | |
4hhb  tvltskyrvHLTPEEKSAVTALWGKV..NVDEVGGEALGRLLVVYPWTQRFFESFgdLSTPDAVMGNPKV

1mbd  KKHGVTVLTALGAILKKKGHheaeLKPLAQSHATKHKIPIKYLEFISEAIIHVLHSRHPgdFGADAQGAM
 490  | ||  || |    |              |   | |                     ||       |    | |
4hhb  KAHGKKVLGAFSDGLAHLDNlkgtFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGkeFTPPVQAAY

1mbd  NKALELFRKDIAAKYkelgyqg
 560   |          | ||
4hhb  QKVVAGVANALAHKYh......

Success for Mols : (1mbd)-(4hhb) : length 126/153 : rms  1.356 : seq-homology 27.0

1mbd  VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFdrfkhlkteaemkaSEDLKKHGVTVLT
   0  |||      | || |      | | | | | |   |              |  | ||  |
4hhb  VLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFphfdlshg......SAQVKGHGKKVAD

1mbd  ALGAILKKKgHHEAELKPLAQSHATKHKIPIKYLEFISEAIIHVLHSRHPGDFGADAQGAMNKALELFRK
  70  ||           |  |   || |          |     |   |  |        | |
4hhb  ALTNAVAHVdDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVST

1mbd  Diaakykelgyqg............................................................
 140      ||
4hhb  Vltskyrvhltpeeksavtalwgkvnvdevggealgrllvvypwtqrffesfgdlstpdavmgnpkvkah

Success for Mols : (1mbd)-(4hhb) : length 126/153 : rms  1.346 : seq-homology 27.0

1mbd  .......VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFdrfkhlkteaemkaSEDLKK
 280         |||      |   | || |     |   | | | | | |   |  |           |   |
4hhb  alahkyhVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFphfdlshg......SAQVKG

1mbd  HGVTVLTALGAILKKKgHHEAELKPLAQSHATKHKIPIKYLEFISEAIIHVLHSRHPGDFGADAQGAMNK
 350  ||   |  ||           |  |   || |          |     |   |  |        |     |
4hhb  HGKKVADALTNAVAHVdDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDK

1mbd  ALELFRKDiaakykelgyqg.........................................................
 420   |      ||
4hhb  FLASVSTVltskyrvhltpeeksavtalwgkvnvdevggealgrllvvypwtqrffesfgdlstpdavmg

Success for Mols : (1mbd)-(4hhb) : length 130/153 : rms  1.391 : seq-homology 25.4

1mbd  ..VLSEGEWQLVLHVWAKVeaDVAGHGQDILIRLFKSHPETLEKFdrfkhlktEAEMKASEDLKKHGVTV
 140     |    |    |   | ||    |    |    | ||     | |    |   |  | |          | ||   |
4hhb  rvHLTPEEKSAVTALWGKV..NVDEVGGEALGRLLVVYPWTQRFFesfgdlstPDAVMGNPKVKAHGKKV

1mbd  LTALGAILKKKGHheaeLKPLAQSHATKHKIPIKYLEFISEAIIHVLHSRHPgdFGADAQGAMNKALELF
 210  | |     |              |   |   |                     ||        |     | |   |
4hhb  LGAFSDGLAHLDNlkgtFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGkeFTPPVQAAYQKVVAGV

1mbd  RKDIAAKYkelgyqg.........................................................
 280      | ||  |
4hhb  ANALAHKYhvlspadktnvkaawgkvgahageygaealermflsfpttktyfphfdlshgsaqvkghgkk
```

Figure 3
Results for the pairwise structure alignment of myoglobin (PDB code 1mbd; Phillips, 1980) and hemoglobin (PDB code 4hhb; Fermi *et al.*, 1984). The numbering shown is the residue ID number for structure 4hhb, which is consecutive from 0 to 515, and the four different solutions are ordered by CE score, resulting in the chains ordered by hit as *D*, *A*, *C* and *B*. The sequences are shown so that upper-case characters indicate the regions that superpose by structure and lower-case characters indicate the regions that are not aligned. No sequence alignment has been performed on the regions not aligned by structure. The '|' character indicates the residues that have identical sequence and the sequence homology is only defined for those residues that align by structure (upper-case characters).

was used to identify commonly occurring protein motifs within protein-fold space independent of the topological sequence connectivity of the proteins.

Finally, an assessment of the quality of the results provided by the program was made by using the published analysis of Novotny *et al.* (2004). Each of the four analysis types used in this study was replicated with the *CAALIGN* algorithm described in this article and the results were compared with the published analysis using a number of alignment servers.

3.1. Pairwise structure alignment

The pairwise alignment between myoglobin (PDB code 1mbd; Phillips, 1980) and hemoglobin (PDB code 4hhb; Fermi *et al.*, 1984) was carried out with a CE score target of 0.05. Four results were obtained for the four different monomer subunits of hemoglobin (shown in Fig. 3) contained within the structure 4hhb. The α-chains of the hemoglobin molecule produced distinctly different alignments, judged by the returned sequence match, compared with the β-chains. A summary for each alignment result shows the length of alignment, the r.m.s.d. of the structure alignment and the number of residues that have an exact sequence match over the structurally aligned region. Since the program ignores chain content and domain structure, multiple alignment results are returned defined only by the sequence order of residues within the alignment and ordered by target quality. The time for alignment is less than 1 s.

3.2. Structure alignment of a protein against the PDB

Two different proteins were aligned against the PDB entries as of 27 September 2004. These structures would result in either poor or no alignment with SSE methods because they do not contain linear secondary structure; the second example only contains C^{α} atoms.

The first example, cartilage matrix protein (PDB code 1aq5; Wiltscheck *et al.*, 1997), has a SCOP classification of coiled coil and is a triple-helix structure; it thus consists of a long curved helical structure (Fig. 4). A total of 221 hits were produced in the search with this protein. The distribution of CE score is shown in Fig. 5 and the alignments with score below 0.016 are from coiled-coil proteins, while those above this include proteins that are not part of this family (as defined by SCOP classification). This protein fold forms the basis of some fibres and a number of solutions consist of multiple hits that lie along the fibre strand, such as 1c1g (Whitby & Phillips, 2000) and 1if3 (Caffrey, 2001).

The structure pectate lyase (PDB code 1pcl; Yoder *et al.*, 1993; Fig. 6) from *Erwinia chrysanthemi* is classified in SCOP as a single-stranded right-handed β-helix and has only C^{α} atoms. The search with this structure produced the 45 unique protein hits shown in Fig. 7, although some of the proteins resulted in multiple alignments (not shown). There are three classes of alignment hits from this analysis. Five hits (1pcl, Yoder *et al.*, 1993; 1ooc, Dehdashti *et al.*, 2003; 1pe9, Dehdashti *et al.*, 2003; 1jrg, Thomas *et al.*, 2002; 1jta, Thomas *et*

al., 2002) have low r.m.s.d.s, more than 330 residues aligned and a sequence similarity of more than 60%. The second set of alignments has an r.m.s.d. around 1.2 Å, an alignment length of about 200 residues for the core β-helix and a sequence similarity of around 30%. The last set consists of structures that align over 90 residues with an r.m.s.d. of about 1.5 Å and have a sequence similarity less than 20%. Many of these structures do not appear in the SCOP database at the current time, although those that do are classified as right-handed β-helix. There is significant variation in the shapes of the aligned structures and thus between the three classes of hits.

3.3. Multiple structure alignment

MSA of a family of proteins is demonstrated on a set of kinase structures that were originally used by Shindyalov & Bourne (1998) as an MSA test set and then used by Novotny *et al.* (2004) as a test set for the MSA servers (Fig. 8). The MSA calculation was repeated using the *CAALIGN* program with a CE target of 0.05. The sequence alignment based on MSA cluster 1 is shown in Fig. 9. The secondary-structure assignment shown was taken from the PDB file 1atp (Zheng *et al.*, 1993) and is included above each section of alignment, where 'H' indicates α-helix, 'S' indicates sheet structure and '.' indicates no assigned secondary structure. The aligned regions within the protein superposition do not correlate well with the secondary-structure content of 1atp, indicating that algorithms based on SSE alone would struggle to maximize the alignment within the set of proteins.

3.4. Fold-fragment analysis of a unique set of proteins

The program *CAALIGN* was used as a method of generating common fragments of super-secondary structure from a unique set of protein structures (Oldfield, 2001) generated from the PDB in January 1999. Data were selected as mathematically determined (Oldfield, 2002) domain fragments using the following criteria: (i) solved by protein crystallography after 1983, (ii) all-atom models of protein with more than ten residues, (iii) no more than 10% bad Ramachandran angles, (iv) a resolution limit of 2.5 Å or better and (v) an exact sequence similarity of less than 80%. 2320 fragments of protein structures were aligned using the clustering algorithm of *CAALIGN* at various minimum alignment lengths. Typical times were of the order of 15 h for each analysis for the 3.6 million alignments. This consisted of 11 cycles of reduction starting from the 2320 × 2320 square alignment matrix, where

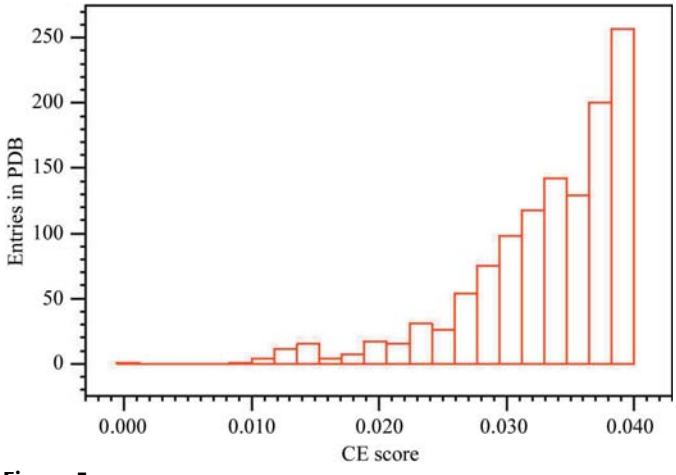

Figure 5
Distribution of CE score between 1aq5 (Wiltscheck *et al.*, 1997) and entries in the PDB (27 September 2004).

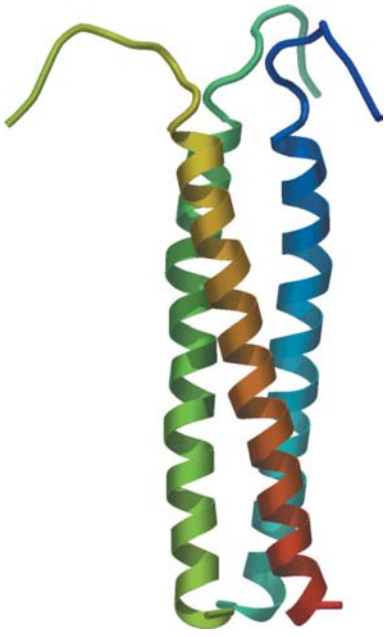

Figure 4
Ribbon diagram of structure 1aq5 (Wiltscheck *et al.*, 1997). The structure is coloured from red at the N-terminus through to blue at the C-terminus.

Figure 6
Ribbon diagram of the structure 1pcl (Yoder *et al.*, 1993). The structure is coloured from red at the N-terminus through to blue at the C-terminus.

```
(1pcl) : length 355/355 : rms    0.000 : seq-similarity  100.0  :  CE-score   0.0000
(1ooc) : length 337/355 : rms    0.703 : seq-similarity   66.8  :  CE-score   0.0021
(1pe9) : length 338/355 : rms    0.730 : seq-similarity   66.6  :  CE-score   0.0022
(1jrg) : length 329/355 : rms    0.726 : seq-similarity   66.0  :  CE-score   0.0023
(1jta) : length 331/355 : rms    0.723 : seq-similarity   66.2  :  CE-score   0.0023
(2bsp) : length 313/355 : rms    1.097 : seq-similarity   37.7  :  CE-score   0.0036
(1bn8) : length 309/355 : rms    1.140 : seq-similarity   38.8  :  CE-score   0.0038
(1qcx) : length 197/355 : rms    1.127 : seq-similarity   32.0  :  CE-score   0.0062
(1plu) : length 213/352 : rms    1.260 : seq-similarity   32.9  :  CE-score   0.0063
(1idj) : length 206/355 : rms    1.180 : seq-similarity   29.1  :  CE-score   0.0062
(1o8j) : length 209/352 : rms    1.225 : seq-similarity   33.5  :  CE-score   0.0062
(1o8i) : length 211/352 : rms    1.228 : seq-similarity   33.2  :  CE-score   0.0062
(1idk) : length 201/355 : rms    1.176 : seq-similarity   30.3  :  CE-score   0.0063
(1o8m) : length 210/352 : rms    1.251 : seq-similarity   33.3  :  CE-score   0.0063
(1o8e) : length 209/352 : rms    1.238 : seq-similarity   33.5  :  CE-score   0.0063
(1o8k) : length 209/352 : rms    1.243 : seq-similarity   33.5  :  CE-score   0.0063
(1air) : length 211/352 : rms    1.247 : seq-similarity   33.6  :  CE-score   0.0063
(1o8d) : length 211/352 : rms    1.242 : seq-similarity   33.2  :  CE-score   0.0063
(1o8f) : length 209/352 : rms    1.231 : seq-similarity   33.5  :  CE-score   0.0063
(1o88) : length 203/352 : rms    1.236 : seq-similarity   33.0  :  CE-score   0.0064
(1o8g) : length 203/352 : rms    1.233 : seq-similarity   33.0  :  CE-score   0.0064
(2pec) : length 203/352 : rms    1.232 : seq-similarity   33.0  :  CE-score   0.0064
(1o8h) : length 205/352 : rms    1.234 : seq-similarity   33.7  :  CE-score   0.0064
(1o8l) : length 206/352 : rms    1.240 : seq-similarity   33.5  :  CE-score   0.0064
(1bhe) : length 142/355 : rms    1.562 : seq-similarity   13.4  :  CE-score   0.0119
(1ru4) : length 104/355 : rms    1.495 : seq-similarity   14.4  :  CE-score   0.0159
(1kcd) : length  98/333 : rms    1.570 : seq-similarity   11.2  :  CE-score   0.0178
(1hg8) : length  99/349 : rms    1.615 : seq-similarity   20.2  :  CE-score   0.0178
(1kcc) : length  98/333 : rms    1.576 : seq-similarity   11.2  :  CE-score   0.0179
(1h3k) : length  98/318 : rms    1.629 : seq-similarity   12.2  :  CE-score   0.0185
(1k5c) : length  91/333 : rms    1.575 : seq-similarity   11.0  :  CE-score   0.0192
(1ib4) : length  89/355 : rms    1.611 : seq-similarity   13.5  :  CE-score   0.0197
(1ktw) : length  91/355 : rms    1.680 : seq-similarity   18.7  :  CE-score   0.0205
(1ia5) : length  80/339 : rms    1.538 : seq-similarity   13.8  :  CE-score   0.0209
(1czf) : length  80/355 : rms    1.635 : seq-similarity   16.2  :  CE-score   0.0225
(1nhc) : length  80/355 : rms    1.698 : seq-similarity   16.2  :  CE-score   0.0236
(1sm1) : length  41/355 : rms    1.441 : seq-similarity   19.5  :  CE-score   0.0386
(1k8f) : length  45/355 : rms    1.728 : seq-similarity   26.7  :  CE-score   0.0418
(1p9a) : length  44/266 : rms    1.817 : seq-similarity   11.4  :  CE-score   0.0460
(1a4y) : length  41/355 : rms    1.721 : seq-similarity    7.3  :  CE-score   0.0471
(1ook) : length  41/355 : rms    1.806 : seq-similarity   12.2  :  CE-score   0.0494
(1qyy) : length  39/355 : rms    1.712 : seq-similarity   12.8  :  CE-score   0.0495
(1gwb) : length  40/355 : rms    1.774 : seq-similarity   12.5  :  CE-score   0.0499
(1ofe) : length  42/355 : rms    1.801 : seq-similarity   11.9  :  CE-score   0.0500
(1upc) : length  42/355 : rms    1.862 : seq-similarity   19.0  :  CE-score   0.0507
```

Figure 7
Hit list created by searching the PDB against the structure of 1pcl (Yoder *et al.*, 1993); multiple hits are not included within this list.

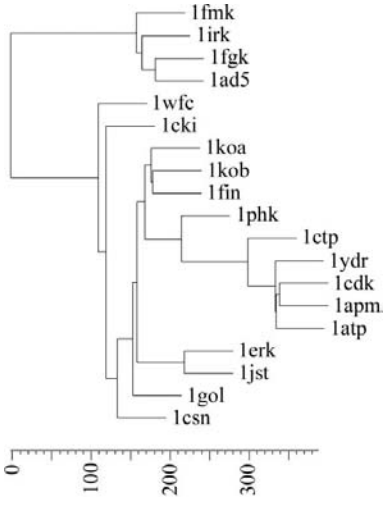

Figure 8
A linkage diagram showing the relationship (by length of alignment) between a set of kinase structures. The *x* ordinate is the alignment length, where zero is no alignment. Clustering was generated from the matrix of structure-alignment lengths for the all-against-all kinase alignment and returned by the MSA algorithm. Those structures close together and with a bifurcation to the right of the diagram are most structurally similar, while progression to the left of the diagram shows family relationship as judged by alignment length.

each cycle approximately halved the size of the matrix. This gave an average alignment time of about 0.02 s for pairwise alignment on a 1 GHz Pentium III PC running Red Hat Linux 7.2; it required 150 Mb of runtime physical memory. Details of the data preparation, an older *ad hoc* clustering method and results, along with their use as molecular-replacement targets, are described in Oldfield (2001). Four examples of structure fragments are shown (Fig. 10) and fragment lists generated from this analysis are available from http://www.ysbl.york.ac.uk/~tom/folds/index.html.

3.5. Assessment of sensitivity

Calculations were performed to allow direct comparison with the results of Novotny *et al.* (2004). Details of the alignment results of cyclophilin are shown in Table 2 and an overall summary for the different fold types is shown in Table 3. These results indicate that the alignment program described here is excellent at determining the hits with a 100% overall rate based on the criteria used in the study of Novotny *et al.* (2004). Domain analysis, variability analysis using NMR data and C^{α}-only tests were all repeated using the criteria set out in the paper by Novotny and coworkers. The results (Table 4) indicate that the *CAALIGN* algorithm produces results that are equivalent to the very best of the structure-alignment servers for all structure types. On the other hand, an analysis using 11 difficult similarities between pairs of proteins taken from Fischer *et al.* (1996) was less successful, resulting in only four matches between protein pairs.

Considering the domain-analysis results as a test, the algorithm replicated the results of Novotny *et al.* (2004), yielding hits for all combinations of one to four domain superpositions. Although the program passes the Novotny test, it should be noted that the algorithm could miss some hits in large multi-domain proteins where differences in packing of the domains could distort the entire structure, perhaps owing to hinge motion, for example. In this case, multiple non-overlapping solutions would be found for each of the domains rather than a single solution; this is considered an advantage as it provides additional information over and above a single overall low-quality alignment. With reference to domain structure, the algorithm has three modes of handling substructure hits. The default mode is to determine the set of 'non-overlapping' solutions, but with an overlap tolerance of

Table 2
Results of an analysis of cyclophilin test structures.

c.f. Table 4 of Novotny *et al.* (2004). The table gives the rank of the hit to the target, the number of other true-positive (TP) hits and the rank of the first false positive (FP), with the number of false positives in brackets, defined as not CATH (Orengo *et al.*, 1997) topology = 2.40.100. Time is the compute time for a 1 Ghz desktop computer to complete the calculation on 28 532 proteins.

Query	Rank (self)	Other TP	Rank of FP	Time (min)
1awq	1	62	None	34
1a33	1	66	None	38
1cyn	1	65	66 (11)	38
1qoi	1	65	67 (1)	39
1lop	1	64	66 (13)	66
1qng	1	64	66 (3)	82
2rmc	1	66	68 (2)	39
1dyw	1	64	66 (4)	77
1ihg	1	65	67 (1)	40

Table 3
Results of structure alignment as a function of the different secondary-structure types.

The numbers in parentheses are the total numbers of test structures. The first row of results indicates the number of search structures found that include other members of the search structures and the second row gives the number of results that included any positive hit, which was the criteria used by Novotny *et al.* (2004)

	α (19)	β (19)	Mixed (15)	Few SSE (8)	Overall (%)
Search list	18	18	14	8	95
Any	19	19	15	8	100

Table 4
Results of structure-alignment sensitivity as a function of variability in target structure using different NMR model structures from 2gda (Baumann *et al.*, 1993).

1gdc (Baumann *et al.*, 1993) is the energy-minimized structure and model 20 is the most dissimilar. The value in parentheses in the first false positive column is the number of false positives.

	1gdc rank	2gda rank	Other true positive	First false positive
1gdc	1	2	19	22 (2)
2gda-2	1	2	15	None
2gda-7	1	2	17	20 (1)
2gda-11	1	2	15	None
2gda-18	1	2	19	22 (1)
2gda-19	1	2	19	21 (2)
2gda-20	1	2	12	14 (10)

ten residues to allow hinge-region overlap. The other options are to return just the best solution or all solutions.

A similar test based on an NMR structure shows that the program is only marginally affected by model variation, as it produces very similar results for all of the seven search structures (Table 4). 1gdc (Baumann *et al.*, 1993) is always returned as the first hit, while 2gda (Baumann *et al.*, 1993) is always returned as the second hit. The number of true positives is slightly different and in only one case (model 20, the most nonrepresentative) did the first false positive appear above the last true positive hit. Unlike the methods based on secondary structure, this algorithm is not sensitive to the

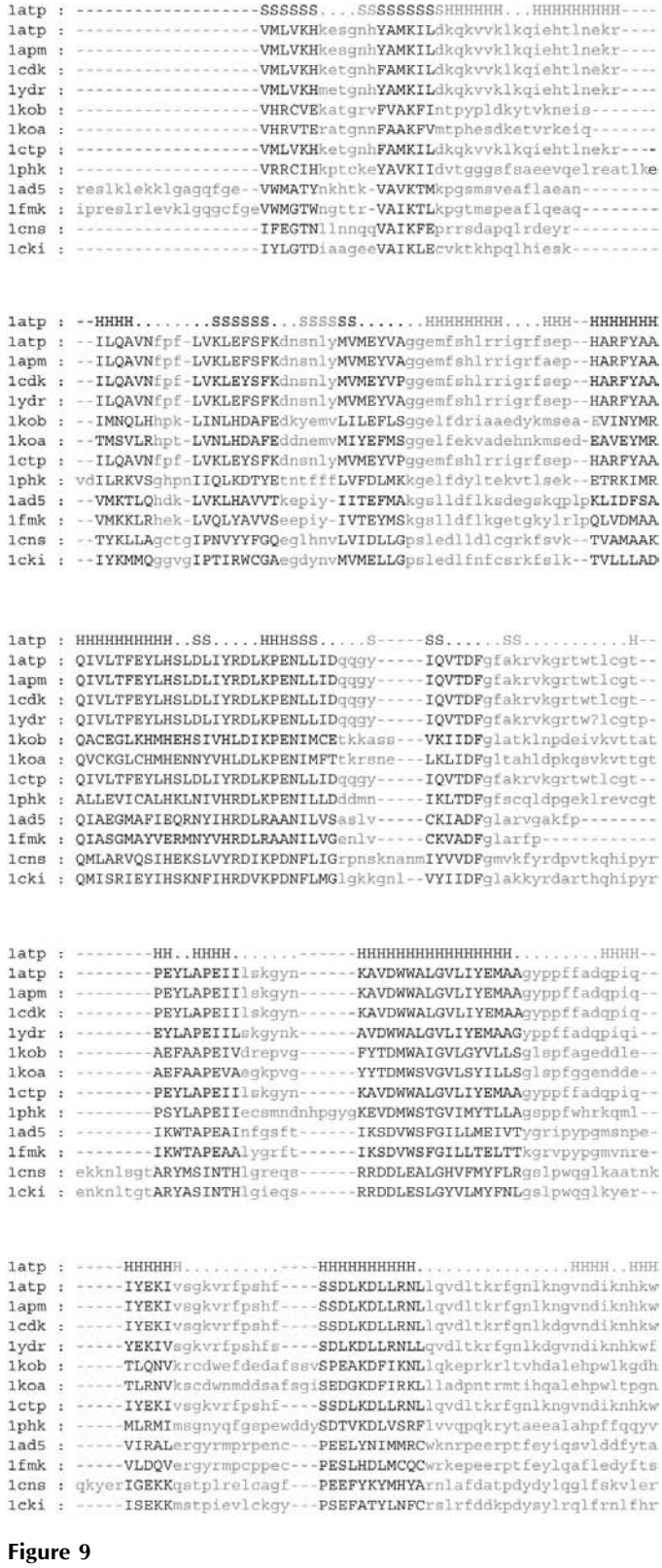

Figure 9
The aligned sequences of 12 kinase protein structures aligned by MSA as cluster 1 from a MSA of 19 protein structures. The secondary structure of 1atp is shown at the top of each alignment, where 'H' indicates helix and 'S' indicates a strand that is part of a sheet. Upper-case characters indicate the alignment by structure; lower-case characters are used for sequence positions that are not aligned. A hyphen marks insertions and regions not structurally aligned are not sequence aligned. The N- and C-terminal unaligned regions were truncated.

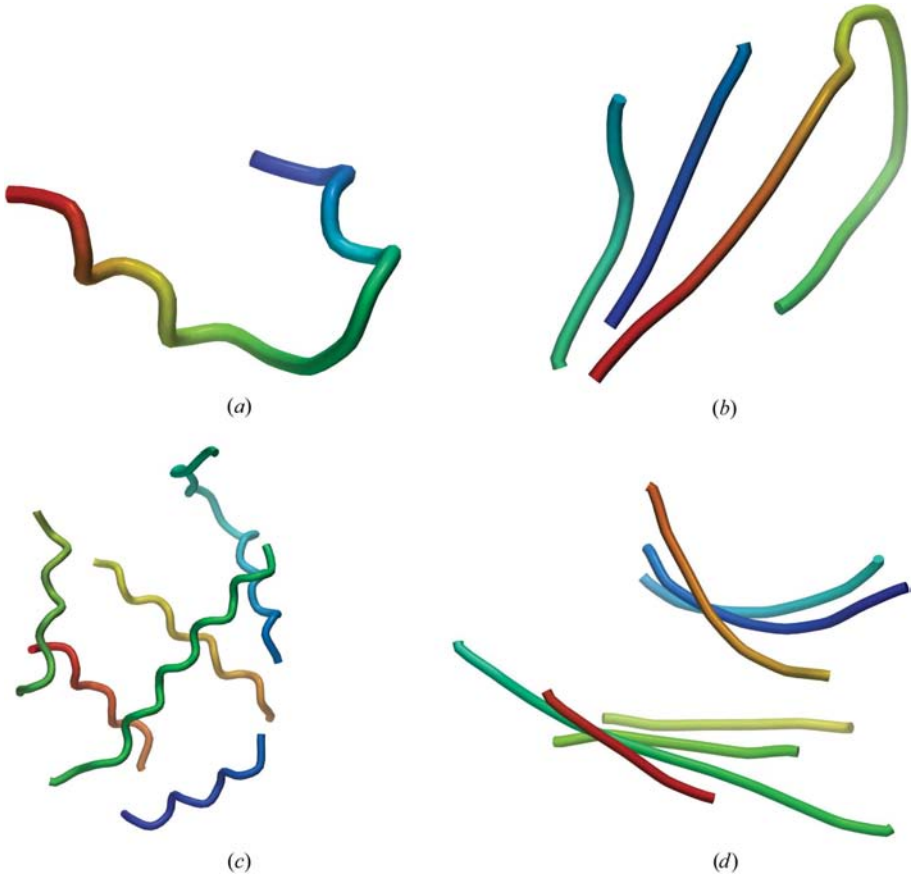

Figure 10
Four common motifs found from MSA of a unique list of proteins selected by sequence difference, date, resolution, X-ray technique and quality.

variation in NMR models that have reduced secondary-structure content as discussed by Novotny *et al.* (2004).

4. Discussion

The program *CAALIGN* returns a protein-structure alignment based on C^{α}-atom positions between a pair of proteins, between one and many proteins and among structure clusters from a list of proteins. The program is not dependent on the secondary-structure content of the protein and only requires the C^{α}-atom positions for alignment. This allows the use of the algorithm to identify nonclassical structures within proteins that cannot be described by a single vector. The sequence order of the aligned regions of proteins is not important for alignment, although this information is returned as part of the result. In this way, the program can find small fragments of recurrent local structure within proteins, such as loops (Fig. 10*a*). This allows study of the sequence dependence of these recurrent features as the algorithm returns a sequence alignment from the structure alignment for both the pairwise and MSA calculations. The program is also designed to return all non-overlapping alignments between a pair of proteins, although an option is provided to return just the best solution.

The calculation speed (Table 2) does not compare favorably with server times reported in the literature, as these are presumably optimized for rapid response with clusters of high-performance computers. Rather, the original papers describing the programs were checked for analysis of performance. The multiple structure alignment of a nondegenerate list of 2320 protein fragments required 150 Mb of run-time memory and took 15 h on a 1 Ghz Pentium III PC running Red Hat Linux 7.2 to complete the approximately 3.6 million alignments. The mean calculation time of 0.02 s is much quicker than quoted for other atomic alignment methods and is not far off that of *SSM* (Krissinel & Henrick, 2004), the benchmark alignment service based on secondary-structure elements. Since the *CAALIGN* program is based on atomic alignment, it has a superior level of sensitivity and is less dependent on secondary-structure type than all other methods.

MSA is unbiased and does not assume that any one set of coordinates is representative of a cluster, since it generates an ideal coordinate solution for each cluster of coordinates. MSA is entirely automated and returns details of the members of the clusters, clustered coordinates, matrices of alignment length/r.m.s.d. values and the sequence-alignment detail based on structure alignment for each cluster. When used on a unique set of proteins, the analysis represents true data mining, where a set of common fragments of folded structure are determined without fold targets defining the search criteria. If the full PDB were used in such an analysis, then the program would determine clusters of identical and near-identical structures as the primary features of interest. To successfully drill down into any data, we must remove known relationships or they will dominate and obscure new information. This type of algorithm, where the target of analysis is frequency of occurrence, does not require knowledge of the meaning of the data (Hand *et al.*, 2001).

The evaluation criteria for sensitivity of alignment used by Novotny *et al.* (2004) show that *CAALIGN* compares well with an array of 11 structure-alignment servers, giving an overall success rate of 100% (Table 3). The structures 1vmo (Shimizu *et al.*, 1994), 1fok (Wah *et al.*, 1997) and 1plq (Krishna *et al.*, 1994) did not match any structure within the Novotny test set, but did return a small number of positive matches in the PDB with the same CATH (Orengo *et al.*, 1997) code. The algorithm is not dependent on structure type and performed equally well in all four sensitivity tests. Searching for structure superposition with a CE score of 0.05 always gives the identity hit first and the true positives at the top of the hit list. It also usually returns a small number of false

positives (defined by CATH topology; Orengo *et al.*, 1997). At higher values of the CE score, the many false-positive results represent super-secondary structure matches including turn structures (see examples in Fig. 10). The algorithm is ideally suited for finding turn-structure superposition, with many common combinations of helix/strand–turn–helix/strand structures being found at higher CE score. The analysis of common loop structure and residue-type dependence within these turn structures, based on structure alignment, is the subject of ongoing research.

The evaluation shows that the alignment is not sensitive to the variability of NMR structures and works equally well with any of the models generated by an NMR experiment. Like other atom-based algorithms, it only requires the C^{α} atoms to form an alignment and does not require derived information such as hydrogen-bond patterns or accurate representation of local structure before an alignment is determined. It can be expected that the algorithm could be readily adapted for use with nucleic acid structures, where vector methods cannot be used.

The difficult-case example results in only four hits out of the 11 aligned pairs. It is probably to be expected that an alignment method based on C^{α} coordinates will not be good at aligning difficult cases because the coordinate superposition is limited to r.m.s.d. target values of 2 Å owing to ambiguity of atomic equivalence; in this case, the $C^{\alpha}-C^{\alpha}$ pseudo-bond distance. The continuity check provides some filtering of mismatches and extends the alignment limit to greater than 2.0 Å resolution, but experience has shown that the alignment sensitivity begins to falls above this limit. Vector methods are able to identify larger variance in fold packing because the atom superposition they determine is by inference from secondary-structure elements. Additionally, it is a little slower than vector methods when compared with the equivalent hardware. The aim of the new algorithm was to create a very sensitive search that is capable of aligning small volumes of protein structure independent of any secondary-structure specification, relying only on C^{α} positions. This allows research leads that are not possible with other alignment programs, such as the analysis of loop structure in proteins and the analysis of the sequence dependence of these loops. The ability to carry out MSA of very large numbers of unique structures allows detailed analysis of the conformational space of protein structure. Both of these types of analysis are the subject of ongoing research and this cannot be undertaken with any of the current structure-alignment algorithms.

5. Availability

The *CAALIGN* program was developed during 2000 to provide a method to carry out discovery-driven data analysis (data mining) for protein folds and has been used stand-alone or as part of the Accelrys package. The algorithm has been used to generate fragment libraries (http://www.ysbl.york.ac.uk/~tom/folds/) for molecular replacement and also to provide a means to carry out research into protein structure (Barry Grant, DPhil thesis, in preparation). The algorithm is under continued development by Accelrys (article forthcoming) in a modified form and is available from Accelrys as the program *3DMA*.

I would particularly like to thank Barry Grant for testing of the program and providing constructive comment with regard to parameterization. Detailed comparisons to the programs *DALI* (Holm & Sander, 1997) and *CE* (Shindyalov & Bourne, 1998) form part of his thesis. I would like to thank Leo Caves and other members of the YSBL for suggestions and much testing of the program. Figs. 4, 6 and 10 were generated using version 2 of the program *AstexViewer* (Hartshorn, 2002) and Figs. 5 and 8 were generated within the program *SQUID* (Oldfield, 1992). I also thank the referees and coeditor for guidance in revising the initial manuscript.

References

Alexandrov, N. N. (1996). *Protein Eng.* **9**, 727–732.
Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990). *J. Mol. Biol.* **215**, 403–410.
Baumann, H., Paulsen, K., Kovács, H., Berglund, H., Wright, A. P., Gustafsson, J. A. & Härd, T. (1993). *Biochemistry*, **32**, 13463–13471.
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.
Caffrey, M. (2001). *Biochim. Biophys. Acta*, **1536**, 116–122.
Dehdashti, S. J., Doan, C. N., Chao, K. L. & Yoder, M. D. (2003). *Acta Cryst.* D**59**, 1339–1342.
Fermi, G., Perutz, M. F., Shaanan, B. & Fourme, R. (1984). *J. Mol. Biol.* **175**, 159–174.
Fischer, D., Elofsson, A., Rice, D. & Eisenberg, D. (1996). *Pac. Symp. Biocomput.* **96**, 300–318.
Gerstein, M. & Levitt, M. (1996). *Proceedings of the Fourth International Conference on Intelligent Systems in Molecular Biology*, pp. 59–67. Menlo Park, CA, USA: AAAI Press.
Gibrat, J. F., Madeh, T. & Bryant, S. H. (1996). *Curr. Opin. Struct. Biol.* **6**, 377–385.
Grindley, H. M., Artymiuk, P. J., Rice, D. W. & Willett, P. (1993). *J. Mol. Biol.* **229**, 707–721.
Guda, C., Scheeff, E., Bourne, P. & Shindyalov, I. (2001). *Pac. Symp. Biocomput.* **6**, 275–286.
Hand, D., Mannila, H. & Smyth, P. (2001). *Principles of Data Mining.* Cambridge, MA, USA: MIT Press.
Hartshorn, M. J. (2002). *J. Comput. Aided Mol. Des.* **16**, 871–881.
Holm, L. & Sander, C. (1993). *J. Mol. Biol.* **233**, 123–138.
Jai, Y., Dewey, G., Shindyalov, I. N. & Bourne, P. E. (2004). *J. Comput. Biol.* **11**, 787–799.
Kabsch, W. (1976). *Acta Cryst.* A**32**, 922–923.
Kleywegt, G. J. & Jones, T. A. (1997). *Methods Enzymol.* **277**, 525–545.
Krishna, T. S., Kong, X. P., Gary, S., Burgers, P. M. & Kuriyan, J. (1994). *Cell*, **79**, 1233–1243.
Krissinel, E. & Henrick, K. (2004). *Acta Cryst.* D**60**, 2256–2268.
Krissinel, E. & Henrick, K. (2005). *CompLife 2005*, edited by M. R. Berthold, pp. 67–78. Berlin: Springer-Verlag.
Leibowitz, N., Flidelman, Z. Y., Nussinov, R. & Wolfson, H. J. (2001). *Proteins*, **43**, 234–245.
Mitchell, E. A., Artymiuk, P. J., Rice, D. W. & Willett, P. (1990). *J. Mol. Biol.* **212**, 151–166.
Mizuguchi, K. & Go, N. (1995). *Protein Eng.* **8**, 353–362.
Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. J. (1995). *J. Mol. Biol.* **247**, 536–540.
Novotny, M., Madsen, D. & Kleywegt, G. J. (2004). *Proteins*, **54**, 260–270.

Oldfield, T. J. (1992). *J. Mol. Graph.* **10**, 247–252.
Oldfield, T. J. (2001). *Acta Cryst.* D**57**, 1421–1427.
Oldfield, T. J. (2002). *Proteins*, **49**, 510–528.
Oldfield, T. J. & Hubbard, R. E. (1994). *Proteins*, **18**, 324–337.
Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B. & Thornton, J. M. (1997). *Structure*, **5**, 1093–1108.
Orengo, C. A. & Taylor, W. R. (1996). *Methods Enzymol.* **266**, 617–635.
Pearson, W. R. & Lipman, D. J. (1988). *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.
Phillips, S. E. (1980). *J. Mol. Biol.* **142**, 531–554.
Sali, A. & Blundell, T. (1990). *J. Mol. Biol.* **212**, 403–428.
Shimizu, T., Vassylyev, D. G., Kido, S., Doi, Y. & Morikawa, K. (1994). *EMBO J.* **13**, 1003–1010.
Shindyalov, I. N. & Bourne, P. E. (1998). *Protein Eng.* **11**, 739–747.
Subbiah, S., Laurents, D. V. & Levitt, M. (1993). *Curr. Biol.* **3**, 141–148.
Thomas, L. M., Doan, C. N., Oliver, R. L. & Yoder, M. D. (2002). *Acta Cryst.* D**58**, 1008–1015.
Vajdos, F. F., Yoo, S., Houseweart, M., Sundquist, W. I. & Hill, C. P. (1997). *Protein Sci.* **6**, 2297–2307.
Vriend, G. & Sander, C. (1991). *Proteins*, **11**, 52–58.
Wah, D. A., Hirsch, J. A., Dorner, L. F., Schildkraut, I. & Aggarwal, A. K. (1997). *Nature (London)*, **388**, 97–100.
Whitby, F. G. & Phillips, G. N. (2000). *Proteins*, **38**, 49–59.
Wiltscheck, R., Kammerer, R. A., Dames, S. A., Schulthess, T., Blommers, M. J., Engel, J. & Alexandrescu, A. T. (1997). *Protein Sci.* **6**, 1734–1745.
Yoder, M. D., Lietzke, S. E. & Jurnak, F. (1993). *Structure*, **1**, 241–251.
Zheng, J., Trafny, E. A., Knighton, D. R., Xuong, N.-H., Taylor, S. S., Ten Eyck, L. F. & Sowadski, J. M. (1993). *Acta Cryst.* D**49**, 362–365.