1 Introduction

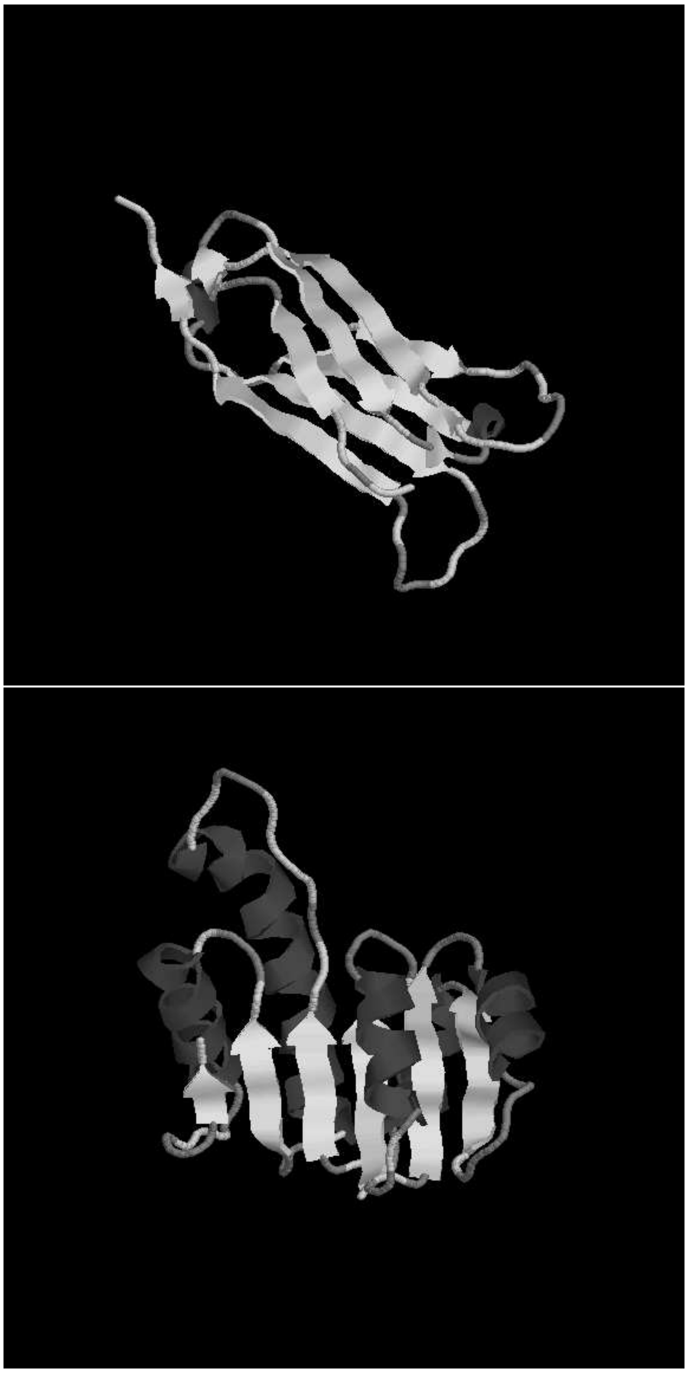
The relationship between a proteins sequence and its structure and function is complex and, as yet, not fully understood. To a large extent, the current understanding of protein sequence/structure/function has come from careful manual examination. However, the recent explosion in the amount of sequence data from genome projects and the ever increasing numbers of protein structures has highlighted the need for automatic approaches to the analysis of biological problems. One such problem is the identification of the key structural features that define a given protein fold. A previous study [1] that applied Inductive Logic Programming (ILP) [2] to the automatic identification of such structural signatures of a protein fold found that large insertions and deletions in a protein structure proved to be a major obstacle. This was because the structurally equivalent portions of proteins with the same overall fold could not easily be identified. In this study, we applied a multiple structure alignment technique to identify equivalent sub-structures in proteins with the same fold. Those sub-structures that had an equivalent sub-structure in a large proportion of proteins with the same fold were deemed to be important (core) to that fold, the remaining sub-structures were deemed to be unimportant (non-core). ILP was then applied to learn the principles governing that fold in terms of only the core sub-structures.

The fold of a protein can be described in terms of simpler structural units. A protein is a linear chain of molecular building blocks called amino acids, or residues. There are 20 different types of naturally occuring amino acids that occur in the chain. The chain itself is directional, with one end referred to as the N-terminus and the other the C-terminus. Each protein is defined by the order in which amino acids occur in the molecular chain and this is called the primary structure, or simply the sequence, of the protein. Segments of the protein sequence can adopt regular conformations, known as secondary structures, called α-helices and β-strands. The parts of the chain that link these secondary structures, and adopt less regular conformations, are generally referred to as loops or coils. The fold, or tertiary structure, of a protein can be defined and classified in terms of the sequential and spatial arrangements of its secondary structure elements. Three examples of different types of fold (TIM barrel, Immunoglobulin and Rossmann), and their particular arrangements of constituent secondary structure elements, are shown in Figure 1. Between secondary and tertiary structure, there is an intermediate level of regular structural unit used to describe the fold of a protein called a supersecondary structure element. The most important supersecondary structure element is the β-sheet. β-sheets are formed by β-strands that align themselves next to each other in three dimensional space to form flat (eg. Rossmann, Figure 1c) or barrel-like (eg. TIM barrel, Figure 1a) structures. Depending on the relative chain directions of the adjacent β-strands, strands can be said to be parallel (eg. adjacent pairs of β-strands in the Rossmann fold, Figure 1c) or antiparallel (eg. adjacent pairs of β-strands in each of the two β-sheets in the Immunoglobulin fold, Figure 1b) to one another. β-sheets consisting only of parallel or only of antiparallel β-strand pairs are referred to as parallel or antiparallel sheets respectively, otherwise they are referred to as mixed sheets. The topology of a β-sheet is also important in determining the tertiary structure of a protein. The topology of a sheet refers to the relative sequential order of the β-strands that occur next to one another in the sheet. For example, a flat β-sheet consisting of three strands could have the first, second or third

Figure 1: (Full-size, full-color versions of figures 1a, 1b, and 1c are found at the end of the present article (in the paper-printed issue), and as separately linked files in the web distribution, due to the size of their data sets). *Examples of protein folds. (a: above) TIM barrel-like fold, (b: next page, top) Immunoglobulin-like fold and (c: next page, bottom) Rossmann fold. The two different types of regular secondary structure elements are highlighted. α-helices are shown in red, β-strands are shown as yellow arrows. The direction of the arrows indicate the direction of the chain for that β-strand. The arrangements of these secondary structure elements both in space and in their relative order along the protein sequence define the type of fold.*

strand (in terms of their relative order in the sequence) in the centre of the sheet (ie. the sheet could have topology 213, 123 or 132 respectively). Supersecondary elements are harder to define in terms of groups of α-helices, or for mixed groups of α-helices and β-strands, because their relationships are less regular than that of β-strand pairs. Relationships between α-helix pairs are described in this study in terms of whether they make contact in space, their positions in the protein sequence, the angles they make with respect to each other (parallel, perpendicular or antiparallel) and the relative location on each helix at which contact is made (towards the N-terminal end of the sequences, towards the C-terminal end or an intermediate position). Spatial contacts between α-helices and β-strands are also used here to define protein folds.

Defining and classifying protein folds is a complex task. There are several fold classification techniques that have already been developed using manual (SCOP [3]), semi-automatic (CATH [4]) and fully automatic techniques (FSSP [5]). Despite the large overlap of these classifications the differences between them are quite significant [6]. The gap between the human expert's understanding of protein sequence/structure and current



fully automated procedures is probably best highlighted by the results of the CASP and CAFASP blind trials [7]. In these trials, human experts and automated web servers were asked to predict the structure of a protein from its sequence alone. Those methods that employed some level of human intervention outperformed fully automated techniques. This is largely because the human expert has the advantage of drawing on the vast amount of background knowledge collected over years of research that is not normally incorporated into fully automatic approaches. This could include knowledge of evolutionary relationships, biochemical principles or knowledge of structural features that are important for a given fold. Such knowledge would allow an expert to screen predictions for those that violate principles already known to them and eliminate them for consideration.

Although the intervention of a human expert may improve fold classification or prediction, such intervention is always subjective. Furthermore, knowledge of protein structural principles only extends beyond a small number of fold types for a few protein experts. Hence, it would be desirable to develop a fully automated method by which expert-like structural principles could be derived in an objective manner for all known types of protein fold. One such method is ILP, a form of machine learning, that can derive rules from examples and background knowledge. ILP has been applied previously to several probems in structural molecular biology [8-12]. In fact, ILP has also previously been applied to the discovery of protein structural principles [1]. In that study, significant local features of several folds were found, such as a short loop between the first α-helix and the following β-strand in proteins with a Rossmann fold, which is known to be part of a functional binding site. However, important global features of folds such as the topology of β-sheets (the order in which β-strands align with each other in space) and the spatial packing arrangements of α-helices have proved elusive to learning with ILP. Some folds with well-known global features (e.g. TIM barrels, Figure 1a) failed to yield any rules at all. This is because of the large number of exceptions in proteins with the same fold. Typically, a protein can have a segment inserted into its structure such that a secondary structure element that is still structurally equivalent to an element in another protein with the same fold can occur much further ahead in sequence and not be recognised as being equivalent. However, there are some standard tools available (such as SSAP [13]) by which a pair of protein structures can be optimally overlaid and aligned in space, so as to locate the structurally equivalent parts of each protein.

In this study, we build multiple structure alignments of all domains with the same fold from pairwise structure alignments calculated with the SSAP program. Obtaining a reliable multiple structure alignment is typically quite difficult [14] but this enables the identification of structurally equivalent (core) secondary structures in those domains and the elimination of inserted and unimportant secondary structures (non-core) that inhibit the learning of global structural features. From these core secondary structure elements rules were learnt for several types of fold and tested for their accuracy.

The SCOP (Structural Classification Of Proteins) database was used in this study to define the fold category of each protein. SCOP was constructed manually and is based on the knowledge of protein expert A. Murzin, taking into account evolutionary relationships between protein sequence, structure and function (Figure 2). The basic structural unit in SCOP is the domain. A domain is a structure that is thought to fold independently. A small protein

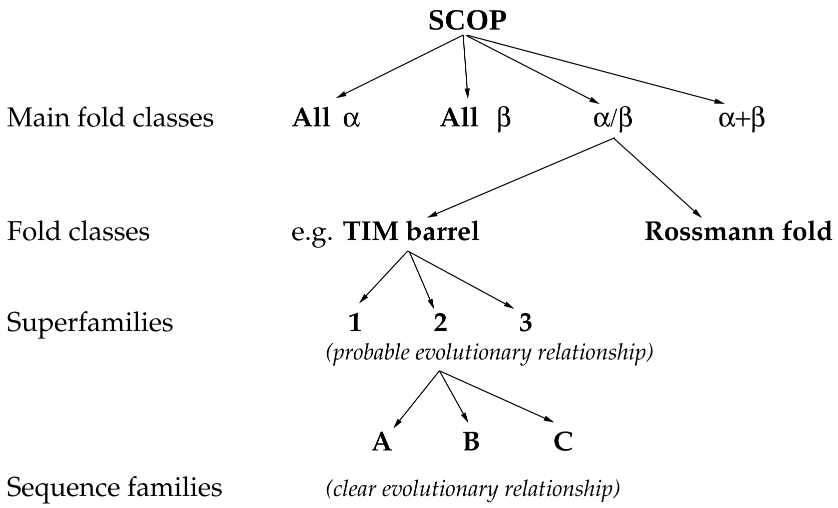


Figure 2: *The SCOP classification of protein structure. The basic structural unit in SCOP is the protein domain. The protein domains in SCOP can be placed into four broad structural categories; All-α (mainly α-helices), All-β (mainly β-strands), α/β (mixture of mainly parallel β-sheets and α-helices) and $\alpha + \beta$ (mixture of mainly antiparallel β-sheets and α-helices). Proteins in each of these main fold categories are further subdivided into distinct fold classes. Proteins in each fold class are classified further with regard to their evolutionary relationships. If two proteins in the same fold class have very similar sequences then they are clearly related and will be in the same sequence family (the bottom level of classification). Those sequence families within a given fold class that don't necessarily have overall sequence similarity with each other, but do have some other evidence from which an evolutionary relationship can be inferred (the presence of a common functional site, similar local structural motif etc.), are grouped into the same "superfamily".*

might consist of only one domain, a larger protein may consist of several domains. Domains are grouped into families. Domains from the same family have similar sequences, indicating that they have evolved from a common ancestor. The next level up in SCOP groups families into superfamilies. Domains from the same superfamily have probably evolved from a common ancestor but this cannot always be inferred from sequence similarity, an expert would have to consider other evidence such as similarities in function. The next level up in SCOP is the fold level. Domains in the same fold category have the same core secondary structure elements with the same spatial and sequential relationships. Lastly, these fold categories are placed into four main fold categories (all-α, all-β, α/β and $\alpha + \beta$) determined by the overall secondary structure distributions in the folds. The SCOP classification scheme also offers the advantage that each fold class has a brief text description of the principles on which each fold type is categorised so that the rules learnt by ILP can also be compared to human expert knowledge.

2 Method

2.1 Data set

The set of protein domains used for each fold category were obtained from the SCOP database [3], release number 1.50. For learning rules, a representative list of domains for each of the main fold classes (all α, all β, α/β and $\alpha + \beta$) were selected using the ASTRAL [15] database. This list of domains included some related domains (that is, some domains are from the same SCOP sequence family), however their inclusion was found to improve both the multiple structure alignments and the quality of the rules learnt as determined by our protein expert (M. Sternberg).

For the purposes of cross-validation, when rules were learnt for a given fold, all domains from each of the majority of SCOP families with that fold were used. However, when those rules were tested, they were evaluated on only one randomly selected domain from each remaining SCOP family in that fold group, in order to eliminate redundancy and bias in testing.

2.2 Multiple structural alignment

In order to define the core structural elements for each domain within a given fold category, a multiple structure alignment of those domains was performed. Multiple structure alignments indicate which residue positions in each of the aligned domain sequences can be considered structurally equivalent and eliminate many of the structurally variable regions from consideration. This is done by orientating the molecules in space to optimally overlay one another (eg. see Figure 3). The residues from different protein sequences that are closest to one another in space are said to be structurally equivalent and can be mapped to one another. These residue relationships can then be used to determine which secondary structure elements in each protein can be considered structurally equivalent (next section).

For the purposes of this work, a technique was employed whereby multiple alignments were constructed by clustering pairwise alignments of domains with the same fold. Such a method tends to neglect global features of the multiple alignment but is fast to calculate. A pairwise alignment is the orientation of two protein structures in space so as to optimise the extent to which they overlay one another. From this aligned pairwise orientation of structures, each given residue in one structure can be mapped to the spatially closest residue in the other structure (eg. for a pairwise alignment between domains D1 and D2, the residue at sequence position 1 in D1 may be structurally equivalent to residue 42 in D2, residue 2 in D1 may be structurally equivalent to residue 45 in D2, and so on). Residues in either structure that are not close in space to residues in the other structure are ignored. Multiple structure alignments can be constructed by clustering these pairwise alignments to find structurally equivalent residues across a number of domains. For example, if it is known from pairwise alignments that residue A in domain D1 is equivalent to residue B in D2, and also that residue B in D2 is equivalent to residue C in D3, then if we cluster these pairwise alignments we can say that A, B and C are all structurally equivalent to one another. By clustering pairwise alignments in this fashion we can build up lists of structurally equivalent residues across a set of domains.

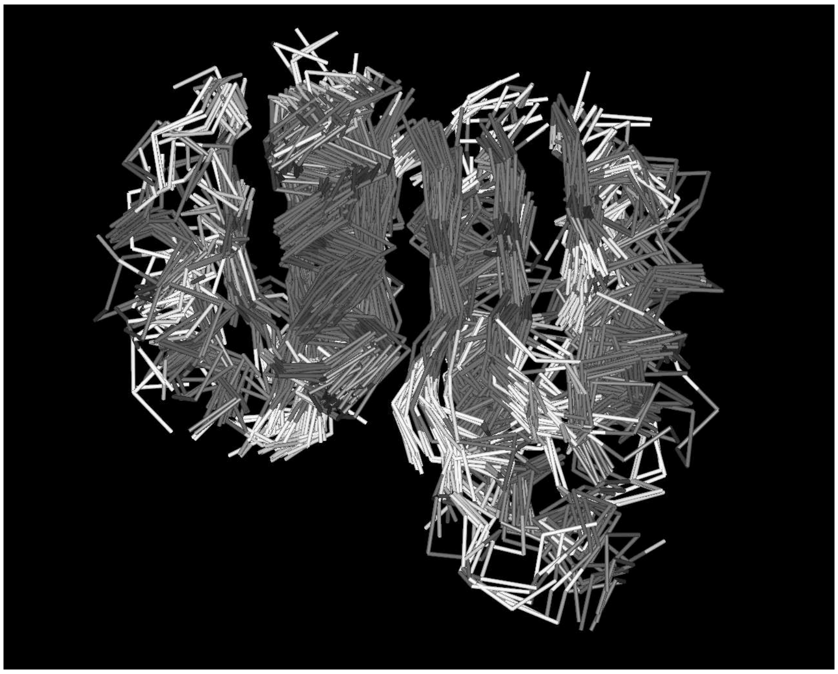


Figure 3: *(A full-size, full-color version of this figure is found at the end of the present article (in the paper-printed issue), and as a separately linked file in the web distribution, due to its size as a data set).* A multiple structure alignment of domains with a Rossmann-like fold. Lines represent the protein chains. Structures are orientated in space so as to optimally overlay each other. Residues from different domains that are close in space in the alignment are considered to be structurally equivalent. Structurally variable portions of the protein structures, that do not have structurally equivalent residues in the other proteins, have been removed. The aligned α-helices are shown in blue and the aligned β-strands are shown in magenta. The six β-strands form a parallel β-sheet.

The method by which pairwise alignments are calculated and the manner in which they are clustered are described below.

For each fold category considered in this study, pairwise alignments were generated for each possible pair of domains in that category using the SSAP program [13]. The structural similarity of each pair of domains could then be measured using the distributions of distances between structurally equivalent residues in the aligned pair of structures. The measure used here is RMSD (Root Mean Square Distance), the root mean square of the distances between structurally equivalent (mapped) residue pairs in the aligned pair of structures. The more similar a pair of structures, the smaller the RMSD calculated from their pairwise alignment will be. The pairwise alignments in each fold category were then clustered with respect to their pairwise structural similarity (RMSD), in a similar manner to that in a previous publication [16], to give the final multiple structure alignment.

The clustering process used here is shown schematically in Figure 4 and proceeded as follows for each fold category: Firstly, a master domain was selected by finding the domain with the lowest average pairwise RMSD to all other domains in that fold category. The master domain then acted as a seed for the subsequent alignment of the remaining domains. To eliminate outliers, any domains that had a pairwise RMSD > 6 Å with the master domain were firstly eliminated from further consideration. Then, the domain with the lowest pairwise RMSD to the master domain was selected and the multiple alignment then consisted simply of the pairwise alignment between that domain and the master domain. Then, the domain with the lowest average RMSD to the domains in the multiple alignment was selected. The pairwise alignment of the new domain with the closest domain in the multiple alignment was then used to determine which residues in the new domain were structurally equivalent to those residues in the rest of the multiple alignment. This process continued iteratively until all domains in that fold category have been considered. In order to avoid corrupting the multiple alignment with misaligned pairwise alignments, domains for which equivalence relations could be made for less than 2/3 of the residues in the multiple alignment at any given step were eliminated from consideration. For most fold categories, only a few domains were eliminated in this way.

2.3 Definition of core elements

The multiple structure alignment indicates which residues in each structure can be considered structurally equivalent. However, to learn rules for protein structure in terms of secondary structure elements (α-helical or β-strand) the elements that can be deemed equivalent have to be identified. To do this, a simple matching scheme was employed to match secondary structures units in different domains based on the extent to which their constituent residues are structurally equivalent, as determined by the multiple alignment calculated in the previous section.

The secondary structure for each protein in the multiple structure alignment was determined using the PROMOTIF [17] program. PROMOTIF takes the three dimensional coordinates of a protein structure and produces a set of files describing the secondary structures and their sequential and spatial relationships. The procedure for determining which secondary structure elements were core and could be considered equivalent to one another was as follows: Firstly, for each domain, those secondary structure elements that have less than half of their constituent residues aligned were removed from consideration. Then, all pairwise "matches" between secondary structure elements in each pair of proteins in the multiple alignment were determined. A "match" was deemed to have occurred between two secondary structure elements from different proteins if each was the largest overlapping element of the other in their respective proteins (Figure 5). Secondary structure elements were then grouped into "maximally matched" groups (ie. each member element of the group has a pairwise "match" with every other member element of that group) (Figure 6a). Surprisingly, in some cases this was enough to find equivalent secondary structures in every protein. However, a more relaxed matching scheme is required to find some of the less easily identifiable core element groups. Therefore, groups of "sub-maximally matched" elements were identified by breaking up the smallest maximally matched group and redistributing its individual member elements to the largest group for which that element matches (in a pairwise fashion) more than 1/2 of the constituent members of that group (Figure 6b). If no such group could be found then the element was eliminated from further consideration (that is, the element is considered non-core). This process continued iteratively until the only remaining groups contained elements in more than 2/3 of the aligned domains. The remaining elements were deemed to be core elements and equivalent to the other member elements in the same group. Each core group is labelled according to its position in the sequence (i.e. the first group is labelled "a", the second "b" and so on).

2.4 Background knowledge

Once the core elements for a protein structure have be defined, the background knowledge containing the structural information for that example can be determined in terms of those core elements. The predicates describing the attributes of, and relationships between, core elements that were considered here and their descriptions are listed in Table 1. All of the structural information required for determining the background knowledge was taken from the output of the PROMOTIF [17] program for that example.

2.5 Learning experiments

Rules were learnt for each SCOP fold category in which protein domains from more than 5 sequence families could be aligned (shown in Table 2) using the Progol-4.4 ILP system. Progol is described in more detail elsewhere [18,19], below is a brief description of the algorithm.

Positive examples were taken from the fold category of interest and the negative examples were taken from all other fold categories in the same SCOP main fold class (all α, all β, α/β or $\alpha + \beta$). Thus, the negative examples were selected only from the most similar folds to the positive examples on the premise that it is more difficult to discriminate against these folds. Examples are given as Prolog statements. For example, a domain with SCOP code d1hdr__, that is a positive example of a Rossmann fold would be represented as:

```
fold(d1hdr__,'NAD(P)-binding Rossmann-fold domains').
```

Progol then proceeds to learn a rule by selecting a positive example and collecting all related background information, also represented as Prolog statements (Table 1), constructing the most specific clause for that example. For example, the most specific clause generated for the domain with SCOP code d1hdr__ is:

```
fold(A,'NAD(P)-binding Rossmann-fold domains') :-
  number_helices('$sk1'=<(A=<'$sk2')),sheet(A,B,para),
  helix(A,C,h,b), helix(A,D,h,g), helix(A,E,h,i),
  strand_position(B,F,3), strand_position(B,G,2),
  strand_position(B,H,1), strand_position(B,I,4),
  strand_position(B,J,5), strand_position(B,K,6),
  contact(D,E), pair(D,E,nterm,nterm),
  helix_angle(D,E,para), has_n_strands(B,6),
  sheet_top_6(B,3,2,1,4,5,6), contains(C,g,nterm),
  contains(C,g,inter), contains(E,g,nterm),
  contains(E,g,inter), contains(C,g), contains(E,g),
  adjacent(C,G), adjacent(E,K), coil(C,G,3), coil(E,K,10),
  contact(C,G), contact(C,H), contact(C,I), contact(D,H),
  contact(D,I), contact(D,J), contact(E,J), contact(E,K),
```

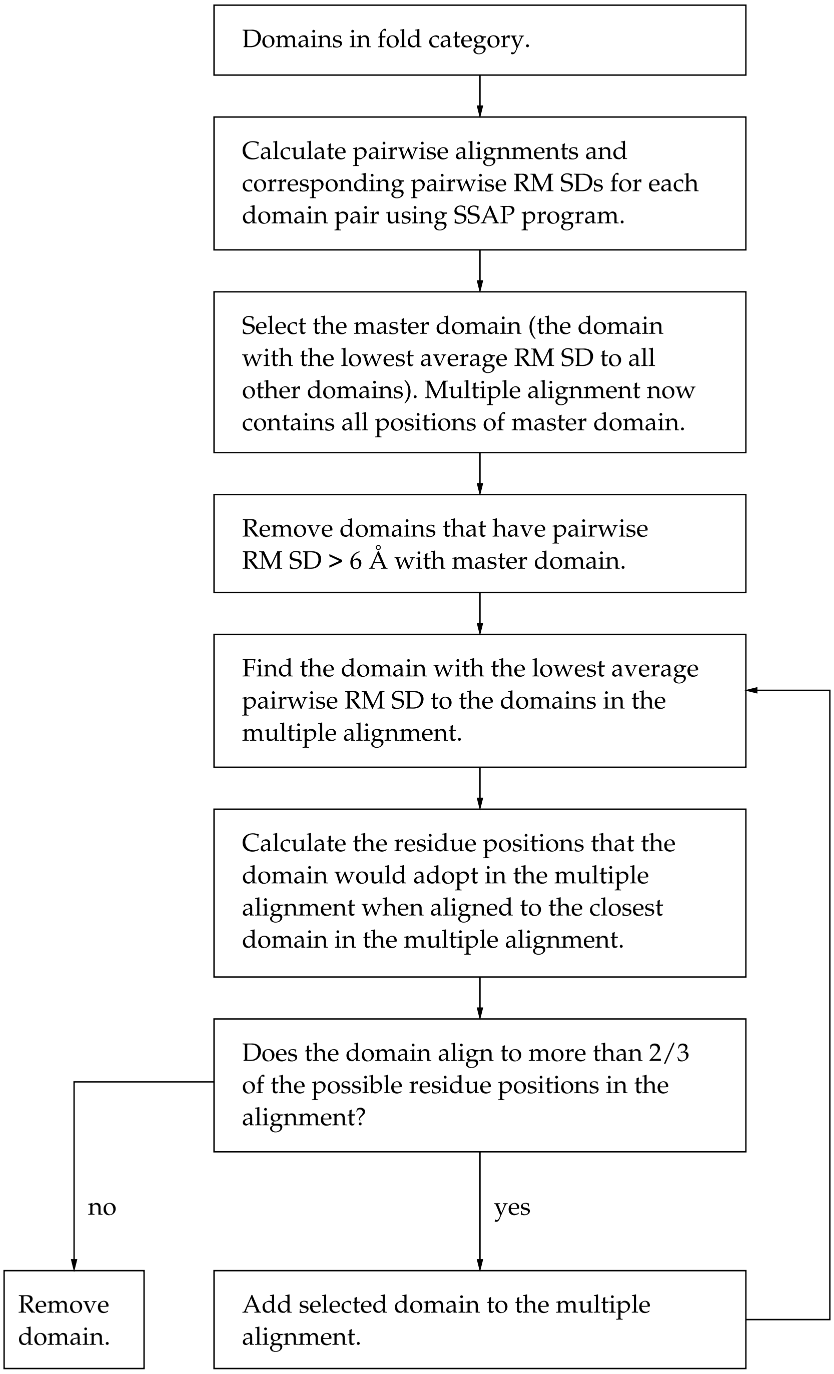


Figure 4: *Flow diagram describing the construction of a multiple structure alignment for a given fold category. RMSD (Root Mean Square Distance) is a measure of how similar two structures are to one another.*

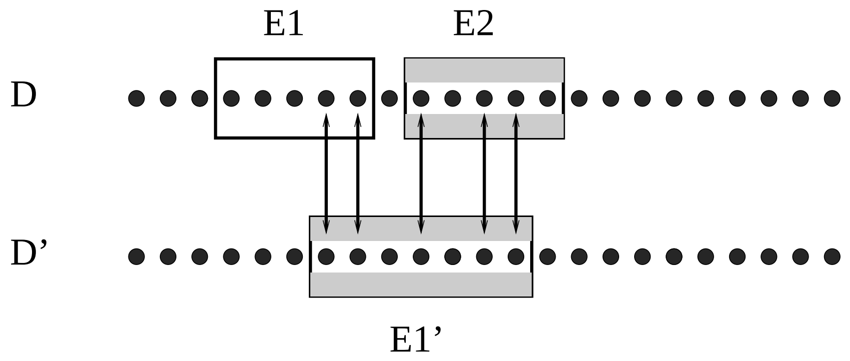


Figure 5: *Matching equivalent secondary structure elements in an aligned pair of protein domains, D and D'. Dots running horizontally represent residues (amino acids) along each protein domains sequence in an alignment of D and D'. Residues (dots) that are directly above or below each other in the alignment are structurally equivalent. Rectangles represent secondary structure elements. Arrows indicate structurally equivalent pairs of residues in which both residues belong to secondary structure elements in their respective domains. A secondary structure element in D is matched to an element in D' if they have a sufficient number of structurally equivalent residues in the alignment. For a pair of elements to match, each must be the element with the largest overlap of aligned residues with the other in their respective proteins. In the example above, the element in D' with the largest overlap with E1 is E1'. However, the element in D with the largest overlap with E1' is not E1 (it is E2). Therefore, E1 and E1' do not match. The element in D' with the largest overlap with E2 is E1' and the element in D with the largest overlap with E1' is E2. Therefore, elements E2 and E1' are said to match (shaded in grey).*

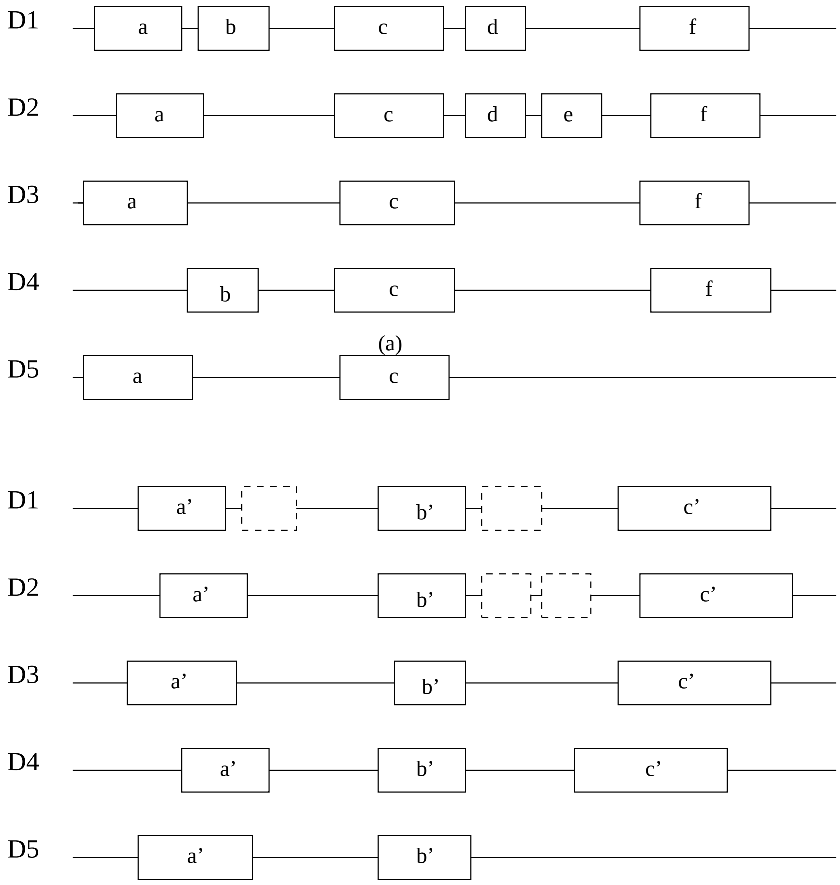


Figure 6: *Finding equivalent secondary structure elements in a multiple alignment. Represented are 5 protein domains (D1, D2, D3, D4 and D5) in a multiple structure alignment. Lines running horizontally represent aligned domain sequences. Those parts of each sequence vertically above or below one correspond to structurally equivalent residues. Rectangles represent secondary structure elements. Groups of equivalent elements are determined from pairwise matches (Figure 5) of elements (see 2.3). (a) Maximally matched groups of elements are identified (groups are labelled "a", "b", etc.). (b) Then, the smaller groups from (a) are disbanded and their member elements are tested for inclusion with the larger groups of elements using a less strict matching criterion. The final groups are then relabelled. Dotted rectangles are the elements deemed to be non-core.*

```
contact(G,F), contact(H,G), contact(H,I), contact(I,J),
contact(J,K),
parallel(F,G),parallel(G,F),parallel(G,H),parallel(H,G),
parallel(H,I),parallel(I,H),parallel(I,J),parallel(J,I),
parallel(J,K), parallel(K,J),
end_strand_distance(B,F,K,19.450),
contains(J,g,inter), contains(J,g).
```

Progol then builds steadily more general rules from the information in the most specific clause until its measure of compression is maximised. The measure of compression used *f* is:

$$f = p - n - c$$

where p is the number of positive examples covered by the rule, n is the number of negative examples covered and c is the length of the rule. The parameter c ensures that for rules with equal coverage of positive and negative examples the shorter one is favoured (i.e. the one that obeys the principle of parsimony). Once an optimal rule is found, the positive examples covered by that rule are removed and the process begins again. This continues until no positive examples remain. Examples of rules found (expressed as Prolog statements) can be found in section 3.

2.6 Parameters

The maximum number of nodes (or hypotheses) tested allowed for an individual search was set to 1000. The noise parameter controlled the number of negative examples that a rule was allowed to cover. The level of noise allowed was 20% (i.e. up to 20% of examples covered by a rule could be false positives).

2.7 Cross–validation

5-fold cross-validation testing was performed for rules learnt for each fold category considered, the accuracy and significance figures are given in Table 2.

3 Results

3.1 Accuracy of rules

Rules were learnt for each SCOP fold type in the four main classes (all-α, all-β, α/β and $\alpha+\beta$) for which representative domains for more than 5 sequence families (and hence more than 5 test examples could be aligned) (Table 2). 5-fold cross-validation tests were conducted for each fold type to determine the accuracy, precision and recall of those rules found. The overall accuracy for the folds considered here is 98% (a random result would be 91%) which is significant according to a χ^2 test, giving a probability $<< 0.01$ that the result could have occurred by chance. This result is dominated by the testing of negative examples but the overall precision and recall (85% and 63% respectively) is reasonably high.

The overall results for each of SCOPs main fold classes (all-α, all-β, α/β and $\alpha+\beta$) are also significant, although rules for several individual fold classes within the main classes had less than 1% significance. Overall, those folds of the all-β class have a much lower recall (34%) than the remaining classes. Several individual fold types in this main fold class do not find any rules at all (those classes for which p = nan (not a number)). This appears to be due to problems with the alignments of β-sheets, although this has not presented as much of a problem with the α/β or $\alpha+\beta$ main fold classes. Indeed, the latter two classes appear to have better overall recall and precision than either the all-α or all-β classes for the fold types studied here. This contrasts with the results of previous ILP learning experiments without the aid of multiple alignments [1].

3.2 Rule composition

The composition of the rules that were learnt for all folds as given in the previous section are shown in Table 3.

The rules learnt for the fold types here appear to be dominated by sheet topology overall. Combining the occurrence of all sheet_top_X predicates, where X is the number of β-strands in the sheet, reveals that 52% of rules learnt contain such a predicate. Learning the topology of β-sheets in a fold is a difficult task. However, once the core elements have been extracted from the folds via a multiple alignment, sheet topology in terms of those elements can be learnt more easily. Other prominent predicates proved to be those describing the angles between contacting helices, general contacts between secondary structure elements and the presence of glycine or proline. Descriptors that proved to be quite prevalent in rules learnt previously for folds [1], such as the length of loops, did not occur at all in rules learnt in this study.

3.3 Interesting rules

Perhaps the most interesting difference between this study and previous work using ILP to discover structural signatures is the ability of this method to capture the global features of folds familiar to human experts. In this section, rules are presented for three important folds and compared to rules learnt previously with ILP without structural alignments [1] and the text descriptions of those folds that have been provided on the SCOP [3] website. The descriptions provided by SCOP are preliminary, and do not represent the sum total of expert-knowledge of the fold, but do give a general expert guide to the general features of the fold. Rules learnt using Progol are output in terms of clauses consisting of the combinations of the types of predicate listed in Table 1, rules for several examples are shown below. But for clarity and ease of comparison, the rules so learnt have also been interpreted into english statements similar to that of the SCOP descriptions. The english statements corresponding to ILP-derived rules, and those given by SCOP, for the Immunoglobulin-like β-sandwich, TIM β/α-barrels and NAD(P)-binding Rossmann-like folds are shown in Table 4.

Table 1: *The predicates from which rules were learnt. Each predicate is a logical expression in Prolog describing attributes of, or relationships between, elements in a protein domain.*

number_helices(Lo =< D =< Hi): The number of helices in domain D.

sheet(D, A Stype): Domain D has a β-sheet A of type Stype, where Stype could be antiparallel, parallel or mixed.

helix(D, B, Htype, Core): Domain D has an helix B at core position Core. B is of type Htype, where Htype can be an α-helix or a 3-10-helix.

strand_position(A, B, N): β-Sheet A has a β-strand B which is the Nth strand in that sheet.

adjacent(B, C): Secondary structure elements B and C are adjacent in sequence.

coil(B, C, N): Elements B and C are adjacent in sequence, separated by a coil of N residues.

contact(B, C): Elements B and C are in contact in space.

antiparallel(B, C): β-strands B and C are antiparallel.

parallel(B, C): β-strands B and C are parallel.

end_strand_distance(A, B, C, Dist): Strands B and C are the end strands of sheet A and are separated by distance Dist in space.

pair(B, C, Bloc, Cloc): Helices B and C are in contact. The parts (N-terminal, C-terminal or middle) of the helices B and C in contact are Bloc and Cloc respectively.

helix_angle(B, C, Angle): Helices B and C are in contact. B and C make angle Angle with each other, where Angle could be antiparallel, parallel or perpendicular.

has_n_strands(A, N): Sheet A has a total of N strands.

barrel(A): Sheet A is a barrel.

bifurcated(A): Sheet A contains a bifurcation.

sheet_top_X(A, N_1, N_2,.., N_X): Sheet A contains X strands, with topology $N_1N_2...N_X$ (i.e. the N's give the relative sequence order of the strands that are spatially adjacent in the sheet).

contains(B, AA, Loc): Element B contains amino acid AA at location Loc, where AA can be either glycine or proline and Loc can be the N-terminal, C-terminal or middle of the element.

contains(B, AA): As above, but independent of location in the element.

The rules learnt by ILP for the Immunoglobulin-like β-sandwich fold (Figure 1b) in this study were:

```
fold(A,'Immunoglobulin-like beta-sandwich') :-
  sheet(A,B,anti), sheet(A,C,anti),
  sheet_top_3(B,1,2,3), sheet_top_4(C,2,1,3,4).

fold(A,'Immunoglobulin-likebeta-sandwich') :-
  sheet(A,B,anti), sheet(A,C,anti),
```

```
strand_position(B,D,1), strand_position(B,E,2),
strand_position(C,F,1), strand_position(C,G,2),
sheet_top_3(C,1,2,3), contact(D,G), contact(E,F).
```

For the Immunoglobulin fold, the important features of the fold according to the experts who designed SCOP are two β-sheets, consisting of 7 strands between them, flat against each other in space much like two layers in a sandwich. It also contains a small β-strand motif involving connections between strands in opposite sheets, known as a greek key motif. The previous application of ILP without structural alignments identified one attribute (that Immunoglobulins sometimes have a helix present) and also found a local feature (a small loop between the 5^{th} and 6^{th} strands) of the fold. With the use of multiple structure alignments however, a global structure description much closer to that of a human expert is obtained. In one rule, it not only finds that there are 7 strands in two sheets but identifies the topology of each. The second rule gives a partial description of which strands in the sheets come into contact in order to form the tertiary structure (the "sandwich" packing of the two sheets).

The rule learnt by ILP for the TIM barrel fold (Figure 1a) in this study was:

```
fold(A,'TIM beta/alpha-barrel') :-
  number_helices(5=<(A=<9)), sheet(A,B,para),
  has_n_strands(B,8).
```

The previous application of ILP to this problem failed to find a rule for the TIM barrel fold. This was largely due the large number of structural variations in TIM barrels. However, the overall fold and global features of TIM barrels are well known. SCOP describes the fold as having a parallel β-sheet with 8 strands (n=8) folded around so that the end strands meet each other to form a closed barrel. It also states that the strands in the sheet are ordered 12345678 and gives other properties that are not included in our ILP representation here, such as the degree to which the strands are "staggered" with respect to each other (S=8). In this study, ILP found a rule for the TIM barrel fold in terms of global features, although it did not have the same depth of detail with regard to sheet topology and geometry as the SCOP description. ILP found a rule that described the number of helices in a TIM barrel (between 5 and 9 core α-helices) and the number of strands in the parallel sheet (8) but did not identify the order of the strands in the sheet or identify the sheet as a closed barrel. However, it is known that some TIM barrel domains have barrels that are not entirely "closed" [20] i.e. the sheet is curved in space but the strands on either end of the sheet do not quite meet (this is known as an "open" barrel). Such a structure is not recognised by the representation used here as being a barrel and hence, such a rule was not found by ILP.

The rules learnt by ILP for the Rossmann fold (Figure 1c) in this study were:

```
fold(A,'NAD(P)-binding Rossmann-fold domains'):-
  number_helices(3=<(A=<4)), helix(A,B,h,b),
  contains(B,g,nterm), contains(B,g,inter).

fold(A,'NAD(P)-binding Rossmann-fold domains') :-
  sheet(A,B,para), helix(A,C,h,g),
```

```
helix(A,D,h,i), helix_angle(C,D,para),
sheet_top_6(B,3,2,1,4,5,6).
```

For the Rossmann-like folds, the method used in this study finds two rules. Firstly, it identifies a Rossmann-like fold as a domain with between 3 and 4 α-helices, with the α-helix at core position "b" having a glycine in both its middle and at the N-terminal end of the helix. This rule has identified two glycines of a known conserved G-X-G-X-X-G sequence motif [21], where G is a glycine and X is any type of amino acid, involved with binding an NAD molecule. This is a conserved functional, rather than simply a structural, feature. The previous application of ILP to this problem [1] also found a rule describing the loop between the 1^{st} strand and the 2^{nd} helix where this conserved region is located. The second rule found in this study for the Rossmann-like fold describes global structural features of the fold. It identifies that the fold has a 6 strand parallel sheet with topology 321456 and also describes two of the helices that are in contact and parallel to one another in space. This is quite similar to the global features given in the SCOP description of the fold. SCOP describes a 6 strand parallel sheet with topology 321456 but does not give structural details of the α-helices except to point out that helices pack on either side of the sheet in 3 layers (for which the shorthand used by SCOP is "a/b/a").

4 Discussion

This study shows that for those fold types that have a reasonable number of examples, expert-like rules can be learnt in a systematic fashion. Furthermore, given that the core sub-structures of the fold can be reliably identified, the significant global features of folds, such as the topology of β-sheets or packing of α-helices, can be described. Some of the rules that have been learnt here clearly reflect some of the principles used by the expert who manually constructed the fold classification system from which the rules were learnt. Given the explosion in the number of structures in recent times, constructing such fold classification schemes manually will become increasingly difficult and an automated approach to derive principles of protein structure, such as the one used in this study, will be increasingly necessary.

However, the approach to learning structural principles from multiple structure alignments of protein domains used here is currently limited to the well-represented SCOP fold types, as multiple structure alignments used here become far less reliable in defining core structural components when there are only a few domains as examples. As the majority of fold types defined by the SCOP classification have very few domain examples, the method used here may not prove to be as useful when applied to all possible fold categories. An automated method that could extract structural principles from less well represented folds would be far more useful generally. Human experts have a better understanding of the well-represented folds and an automated method may simply give them structural features that they already know for these folds. However, human expertise does not generally extend to many of the less-well represented folds and automated methods of knowledge discovery could yield useful insight in these cases.

Given the low level of data for most fold types, directly application of multiple structure alignment may be difficult and inaccurate in determining the core sub-structures. However, it may be possible to learn principles that can predict which secondary structure elements are core and which are non-core. Intuitively, one might suspect that secondary structure elements that are quite small, on the ends of β-sheets or do not make many contacts with other parts of the structure may be more variable than those that are not. Such elements, with fewer physical constraints, may be more likely to be non-core. ILP may be able to learn such rules for core and non-core elements from those examples here whose multiple alignments are more reliable. If such rules proved to be physically and biologically sensible, they could be transferred to those folds with fewer examples to predict the core elements of a fold. Then, ILP could again be used to derive structural principles from the predicted core elements in a similar way to that used in this study.

Apart from extending this method to folds with fewer examples, other improvements could be made to this method. The representation used here (Table 1) does not include sequence motifs known to be associated with particular functions, such as those collated in the PROSITE [22] database. The inclusion of such motifs could give further insight into the relationships between sequence/structure/function and assist in fold classification and prediction. Currently, the only sequence properties that are represented here are the presence of glycines and prolines in secondary structure elements.

This study shows that ILP can learn global structure principles for fold after identifying the core structural elements via a multiple alignment. The rules learned for well-represented folds reflect principles known to protein experts.

Acknowledgements

This work was supported by a BBSRC grant.

References

[1] Turcotte, M., Muggleton, S.H. and Sternberg, M.J. (2001). Automated discovery of structural signatures of protein fold and function. *Journal of Molecular Biology*, 306, 591-605.

[2] Muggleton, S.H. and Raedt, L.D. (1994). Inductive logic programming: theory and methods. *Journal of Logic Programming*, 19/20, 629-679.

[3] LoConte, L., Ailey, B., Hubbard, T.J.P., Brenner, S.E., Murzin, A.G. and Chothia, C. (2000). SCOP: a structural classification of proteins database. *Nucleic Acids Research*, 28, 257-259.

[4] Pearl, F.M.G., Lee, D., Bray, J.E., Sillitoe, I., Todd, A.E., Harrison, A.P., Thornton, J.M. and Orengo, C.A. (2000). Assigning genome sequences to CATH. *Nucleic Acids Research*, 28, 277-282.

[5] Holm, L. and Sander, C. (1998). Touring protein fold space with DALI/FSSP. *Nucleic Acids Research*, 26, 316-319.

[6] Hadley, C, Jones, D.T. (1999). A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. *Structure Fold. Des.*, 7,1099-112.

[7] Moult, J., Hubbard, T., Fidelis, K. and Pedersen, J.T. (1999). Critical assessment of methods of protein structure prediction (CASP): round III. *Proteins*, 37 (S3), 2-6.

[8] Muggleton, S.H., King, R.D. and Sternberg, M.J. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Engineering*, 5, 647-657.

[9] King, R.D., Muggleton, S.H., Lewis,R.A. and Sternberg, M.J. (1992). Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. Nat. Acad. Sci. USA*, 89, 11322-6.

[10] King, R.D., Clark, D.A., Shirazi, J. and Sternberg, M.J. (1994). On the use of machine learning to identify topological rules in the packing of beta-strands. *Protein Engineering*, 7, 1295-1303.

[11] Hirst, J.D., King, R.D., and Sternberg, M.J. (1994). Quantitative structure-activity relationships by neural networks and inductive logic programming. I. The inhibition of dihydrofolate reductase by pyrimidines. *Journal of Computer-aided Molecular Design*, 8, 405-20.

[12] King, R.D., Muggleton, S.H., Srinivasan, A. and Sternberg, M.J. (1996). Structure-activity relationships derived by machine learning: the use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proc. Nat. Acad. Sci. USA*, 93, 438-42.

[13] Taylor, W .R. and Orengo, C.A. (1989). Protein structure alignment. *Journal of Molecular Biology*, 208,1-22.

[14] Gerstein, M. and Levitt, M. (1998). Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Science*, 7, 445-456.

[15] Brenner, S.E., Koehl, P. and Levitt, M. (2000). The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Research*, 28, 254-256.

[16] Kelley, L.A., MacCallum, R.M. and Sternberg, M.J.E. (2000). Enhanced genome annotation using structural profiles in the program 3D-PSSM. *Journal of Molecular Biology* 299, 499-520.

[17] Hutchinson, E.G. and Thornton, J.M. (1996). PROMOTIF - a program to identify and analyze structural motifs in proteins. *Protein Science*, 5, 212-220.

[18] Muggleton, S.H. (ed.) (1992). *Inductive Logic Programming.* Academic Press, London.

[19] Muggleton, S.H. (1995). Inverse entailment and progol. *New Generation Computing Journal*, 13, 245-286.

[20] Nagano, N., Hutchinson, E.G. and Thornton, J.M. (1999). Barrel structures in proteins: automatic identification and classification including a sequence analysis of TIM barrels. *Protein Science*, 8, 2072-84.

[21] Wierenga, R.K., Terpstra, P. and Hol, W.G.J. (1986). Prediction of the occurrence of the ADP-binding β-α-β-fold in proteins. *Journal of Molecular Biology*, 187, 101-107.

[22] Hofmann, K., Bucher, P., Falquet, L. and Bairoch, A. (1999). The prosite database, its status in 1999. *Nucleic Acids Research*, 27, 215-219.

Table 2: *Cross-validation results. The cross-validated accuracy is shown for each individual fold category, the four main SCOP fold classes and for all folds combined. The columns give, respectively, the SCOP fold class, the numbers of positive and negative examples, the cross-validated accuracy and error, the accuracy expected given a random guess, and on the next page, the* χ^2 *significance, the corresponding probability* p, *the precision (proportion of positive predictions that are true positives) and the recall (proportion of positive examples that are correctly predicted).*

Fold	pos/neg	Acc.(%)	Rand (%)
Long α-hairpin	7/229	96 ± 1	96
3-helical bundle	30/206	97 ± 1	77
4-helical, up-and-down bundle	10/226	96 ± 1	93
EF-hand	9/227	97 ± 1	93
SAM-domain	10/226	96 ± 1	94
α-α superhelix	8/228	97 ± 1	96
	74/1342	97 ± 0	91
Immunoglobulin β-sandwich	16/174	96 ± 1	88
Diphteria toxin/etc	7/183	99 ± 1	94
Galactose-binding domain	7/183	96 ± 1	96
SH3 barrel	7/183	96 ± 1	96
OB-fold	12/178	97 ± 1	90
β-trefoil	6/184	97 ± 1	97
Reductase/etc.	7/183	97 ± 1	95
7-bladed β-propeller	6/184	97 ± 1	96
	68/1452	97 ± 0	94
TIM β/α barrel	30/181	94 ± 2	78
Rossmann	6/205	99 ± 1	94
Flavodoxin	15/196	97 ± 1	88
Thioredoxin	6/205	99 ± 1	94
α/β hydrolases	17/194	98 ± 1	86
	74/981	97 ± 0	88
β-grasp	8/255	99 ± 1	94
FAD-linked reductases	6/257	100 ± 0	96
Cystatin	7/256	100 ± 0	94
Ferrodoxin	32/231	98 ± 1	79
Zincin	7/256	98 ± 1	94
	60/1255	99 ± 0	91
Overall	276/5030	98 ± 0	91

Fold	χ^2	p	Prec. (%)	Rec. (%)
Long α-hairpin	3.4	0.07	0	0
3-helical bundle	178.9	0.00	88	93
4-helical, up-and-down bundle	31.9	0.00	50	40
EF-hand	62.1	0.00	62	56
SAM-domain	26.4	0.00	60	30
α-α superhelix	31.6	0.00	100	25
	547.7	0.00	74	57
Immunoglobulin β-sandwich	78.9	0.00	100	50
Diphteria toxin/etc	107.8	0.00	100	71
Galactose-binding domain	nan	nan	0	0
SH3 barrel	6.1	0.01	0	0
OB-fold	92.0	0.00	100	58
β-trefoil	nan	nan	0	0
Reductase/etc.	29.0	0.00	100	29
7-bladed β-propeller	3.2	0.08	50	17
	435.1	0.00	92	34
TIM β/α barrel	111.5	0.00	91	67
Rossmann	116.4	0.00	83	83
Flavodoxin	110.4	0.00	91	67
Thioredoxin	116.4	0.00	83	83
α/β hydrolases	133.6	0.00	93	76
	643.8	0.00	90	72
β-grasp	171.1	0.00	88	88
FAD-linked reductases	220.1	0.00	100	100
Cystatin	196.7	0.00	88	100
Ferrodoxin	218.3	0.00	94	94
Zincin	122.9	0.00	67	86
	1060.3	0.00	89	93
Overall	2743.1	0.00	85	63

Table 3: *Composition of the rules learnt. Given are the relative proportion of rules learnt containing at least one of each type of predicate.*

Predicate	Percentage of rules containing predicate
helix	40.74
sheet	33.33
strand_position	22.22
helix_angle	22.22
contact	18.52
sheet_top_4	18.52
contains	18.52
sheet_top_5	18.52
pair	11.11
sheet_top_3	7.41
parallel	7.41
has_n_strands	7.41
end_strand_distance	7.41
sheet_top_6	3.70
sheet_top_7	3.70
antiparallel	3.70
adjacent	3.70

Table 4: *Rules learnt for several fold types. Rules learnt using the method used in this study (ILP(new)) are compared to the rules learnt previously with ILP without multiple structure alignment (ILP (old)) and expert-like descriptions of those folds taken from the SCOP database (SCOP). Terms used in the SCOP descriptions are described in section 3.3.*

Immunoglobulin (1 002 001)

Rule type	Rule
SCOP	Sandwich; 7 strands in 2 sheets; greek-key; some members of the fold have additional strands.
ILP (old)	There is at most one helix, the loop between the 5^{th} and 6^{th} strands is three to seven residues long.
ILP (new)	Has antiparallel sheets B and C; B has 3 strands, topology 123; C has 4 strands, topology 2134. OR Has antiparallel sheets B and C; C has 3 strands, topology 123; the 1^{st} and 2^{nd} strands in B are D and E respectively; the 1^{st} and 2^{nd} strands in C are F and G respectively; E and F are in contact; D and G are in contact.

TIM barrel (1 003 001)

Rule type	Rule
SCOP	Contains parallel beta-sheet barrel, closed; n=8, S=8; strand order 12345678; the first six superfamilies have similar phosphate-binding sites.
ILP (old)	No rule found.
ILP (new)	Has between 5 and 9 helices; Has a parallel sheet of 8 strands

Rossman-like (1 003 002)

Rule type	Rule
SCOP	Core: 3 layers, a/b/a; parallel beta-sheet of 6 strands, order 321456; the nucleotide-binding modes of this and the next two folds/superfamilies (1 003 003 and 1 003 004) are similar.
ILP (old)	The 1^{st} strand is followed by a helix, the two elements are separated by a coil of about one residue. The 6^{th} strand is followed by a helix.
ILP (new)	Has between 3 and 4 helices, has α-helix B at core position "b"; B contains a glycine in both its middle and n-terminal regions OR Has a parallel sheet B of six strands with topology 321456; has α-helices C and D at core positions "g" and "i" respectively; C and D are in contact and parallel.