Title: The automatic discovery of structural principles describing protein fold space

Adrian P. Cootes[1,3], Stephen H. Muggleton[2,4] and Michael J.E. Sternberg[1,3]*

Current addresses

[1] Imperial College of Science, Technology and Medicine, Department of Biological Sciences, South Kensington, London, SW7 2AZ, UK.

[2] Imperial College of Science, Technology and Medicine, Department of Computing, South Kensington, London, SW7 2AZ, UK.

Previous addresses

[3] Cancer Research UK, Biomolecular Modelling Laboratory, 44 Lincoln's Inn Fields, London, WC2A 3PX, UK.

[4] University of York, Department of Computer Science, Heslington, York, YO1 5DD, UK.

a.cootes@ic.ac.uk, s.muggleton@ic.ac.uk, m.sternberg@ic.ac.uk

* corresponding author

Short title

Describing protein fold space

Summary

The study of protein structure has largely been driven by the careful inspection of experimental data by human experts. However, the rapid production of protein structures from structural-genomics projects will make it increasingly difficult to analyse (and determine the principles responsible for) the distribution of proteins in fold space by inspection alone. Here, we demonstrate a machine-learning strategy that automatically determines the structural principles describing 45 classes of fold. The rules learnt were shown to be both statistically significant and meaningful to protein experts. With the increasing emphasis on high-throughput experimental initiatives, machine-learning and other automated methods of analysis will become increasingly important for many biological problems.

Keywords

Protein structure, machine learning, inductive logic programming, fold space,.

Introduction

Structural-genomics is predicted to enhance the understanding of protein fold space greatly in the near future through an explosion in the number of experimentally-determined protein structures. Several hundred different types of fold have already been observed. Proteins are not distributed evenly amongst these fold types, many adopting a limited number known as "superfolds" [1]. In contrast, the vast majority of observed folds are adopted by only a small number of proteins. The distribution of proteins throughout fold space needs to be understood in terms of their internal structural arrangements and in the wider context of protein folding, function and evolution. Given the complicated nature of any given proteins three-dimensional molecular arrangement, the analysis of fold space is a difficult task even with the current, relatively low, number of known folds. With structural-genomics projects aiming to rapidly determine all protein folds in biota (predicted to be anywhere from 1000 to 10000 different types[1; 2]) this will only become more difficult. Such a large influx of new experimental data will require rapid, automated methods of analysis in order to understand this complex problem fully.

The first step in understanding complex phenomena in biology has often been classification. There are currently several classification schemes that group the current set of known protein structures according to the similarity of their folds. The SCOP[3], CATH[4] and FSSP[5] databases have been developed using manual, semi-automated and fully automatic methods of structure comparison respectively. Recently, a method has also been developed to classify proteins in terms of their proximity to a set of

idealised protein structural units[6]. Some of these databases have been shown to be largely similar but, nonetheless, significantly different in their assignment of structural similarity[7]. The identification of proteins with similar structure is important, particularly in highlighting evolutionary relationships not easily identified by sequence comparison alone. However, in order to understand a biological problem, it is not enough to simply identify classes of like objects. Classification alone will not explain why some types of fold are more prevalent than others or why some potential protein folds are not observed at all. To do so would require an understanding of how protein folds differ in terms of their fundamental structural properties in the context of protein folding and function.

Experts usually describe the fold of a protein in terms of the spatial and topological arrangements of their regular secondary structure elements. The only database carrying such descriptions in detail for all fold classes (albeit "preliminary" ones for the $\alpha+\beta$ main fold class) is the SCOP[3] database. The SCOP database, widely used by the protein structure community, is manually curated and annotated by the protein expert A. Murzin. While these descriptions give expert-like structural principles behind each fold class, many of them do not discriminate between that fold and other fold classes. For example, the SCOP descriptions for the Immunoglobulin, Prealbumin and Cupredoxin folds are almost identical. Furthermore, these descriptions are subjective. Given this, and the rapidly expanding number of folds expected from structural-genomics programs, it would be useful to generate such descriptions automatically. This would enable the objective identification of features

that make each fold unique and, as a consequence, give the structural principles underpinning fold space.

Here, we have applied Inductive Logic Programming (ILP)[8; 9; 10], a machine learning technique, to the problem of automatically generating descriptions for fold classes in SCOP. In this way, the rules generated by ILP could be compared to those given in SCOP. However, the method can be applied to learn structure principles for any database. ILP has previously been applied to many problems in molecular biology[11; 12; 13; 14; 15]. In a previous application of ILP to the learning of protein structure principles[16] only local features of folds (that is, features relating to a short section of sequence) were identified. It was noted that insertions and deletions made the learning of global fold features extremely difficult due to the large number of exceptions presented. Here, we circumvent this problem by utilising multiple structure alignments as well as ILP to obtain global descriptions. This enabled us for the first time to learn expert-like rules describing protein structure folds in an automatic fashion.

The approach

The overall scheme for learning fold descriptions is shown in Figure 1. Rules for each fold were learnt using the Progol-4.4 ILP system[8; 10]. Progol learns rules from known examples and background knowledge. Examples in this study were defined using the SCOP protein structure database[3]. When learning rules for a given fold, positive examples were selected from the domains within the corresponding SCOP fold category while negative examples were selected from domains within all other fold

categories in the same SCOP main fold class (all-α, all-β, α/β or α+β). Background knowledge consisted of structural information for each example considered, derived from secondary structure and multiple structure alignment information (as described in the Methods section). For each fold category in the four major main fold classes in SCOP (all-α, all-β, α/β or α+β), a multiple structure alignment was constructed from selected domains with that fold. Structurally equivalent secondary structure elements were identified by the relative degree of overlap with one another in the alignment. Core secondary structure elements were then defined as those elements that had a structurally equivalent element in a majority of aligned domains. Non-core elements were subsequently ignored. Thus, the background knowledge of each domain consisted of the properties of, and relationships between, only the core secondary structure elements. Element properties such as the relative sequence position of a strand in a sheet or the presence of a glycine or proline were considered. One of the major advantages of ILP over other forms of machine learning is that relations between objects can be easily represented. Relations between core elements were included in the background knowledge, such as contacts between core elements in space and the length of coil between elements adjacent in sequence. The full list of attributes and relations considered are described in the Methods section.

Progol takes as input both examples and background knowledge represented as logic programs. Progol builds rules by selecting a positive example and constructing hypotheses from logic programs that make up that examples background knowledge. Rules are constructed so as to maximise compression. The measure of compression used is *f*, where:

$$f = p - n - c$$

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. When a rule with maximal compression has been found, the positive examples matching that rule are removed. Progol then proceeds to learn rules from the remaining examples in a similar fashion to that described above. This process is iterated until there are no remaining positive examples.

The rules output by Progol are also expressed as logic programs and can be readily interpreted by a human expert. For example, the rules for the Rossmann fold were output as follows:

```
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).
```

and:

```
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).
```

These rules are written in plain English in Table 2 and will be analysed in Results.

Results

Rules were learnt for 45 of the more common protein folds using ILP. The total number of rules learnt for these folds was 66, an average of ~1.5 rules per fold. The full list of rules learnt can be found on our website (http://www.sbg.bio.ic.ac.uk/~cootes/rules.html).

Cross-validated accuracy

The ILP scheme used here was subjected to a rigorous cross-validation procedure, the results of which are shown in Table 1. The overall accuracy was high (97%) but dominated by predictions for one class of example, the negative examples. A large number of negative examples were included in order to minimise the learning of spurious rules. Therefore, the accuracy expected if one were to simply predict that every example was a member of the largest (negative) class was also high (95%). However, a Pearson's ^{2} test indicated that the results were statistically significant when compared to such a largest class prediction ($^2 = 58.5$, $p << 0.01$).

In order to isolate the performance on the prediction of positive examples, the recall and precision have been included in Table 1. The recall is the percentage of positive examples that are predicted to be positive. The precision is the percentage of examples predicted to be positive that are actually positive examples. For the 45 fold classes examined here, the overall precision was found to be reasonably high (77%) although the recall was relatively low (55%). This was largely due to the difficulties

of producing stable multiple structure alignments, particularly for those folds that had a low number of examples. For the 10 fold classes with the highest number of positive examples used here, the overall precision and recall were 83% and 69% respectively.

Fold rules

Several examples of the rules learnt automatically for well-known folds are explored further here and are shown in Table 2, with corresponding structures and features of interest shown in Figure 2. These folds were selected for their biological interest and also to highlight improvements in the automatic descriptions of folds and discrepancies with the current understanding of protein structure.

In order to compare the rules learnt with ILP to those of a protein structure expert, the ILP rules were compared to the corresponding SCOP descriptions. A rigorous comparison of rules is difficult given that the SCOP descriptions were manually generated and use a different glossary of terms to that used in this study. However, inspection of the rules reveals that the principles learnt automatically using ILP are often similar to those given by the expert responsible for SCOP. Table 2 lists several examples of folds, the corresponding ILP rules learnt in this study and the SCOP description for that fold.

The rules learnt in this study were also compared with those of a previous study[16] that did not utilise multiple structure alignments. As different sets of protein folds were considered in these studies, a rigorous comparison is also difficult in this case.

However, for those fold examples listed in Table 2 that were also considered in the previous study, the corresponding rules are listed in Table 3 for the purposes of comparison by inspection.

Global fold descriptions

In contrast to the previous study[16] referred to above, the incorporation of structural superpositions into the background knowledge has enabled important global, as well as local, fold properties to be automatically identified. In particular, descriptors giving the size and topology of β-sheets, and those describing the total number of helices in a fold, were among the most prevalent found in the rules generated (Table 5). For example, the 321456 topology of the Rossmann fold parallel sheet was identified (Figure 2b) as well as the topologies of both antiparallel sheets making up the Immunoglobulin sandwich structure (Figure 2c). The TIM-barrel fold is identified as having an 8-stranded parallel β-sheet (Figure 2a). Without structural superpositions[16], the rules learnt previously (Table 3) described such features as a short loop between the first helix and the following strand in the Rossmann fold, part of the NADH binding motif, and the loop between the fifth and sixth strands of the Immunoglobulin fold. No meaningful rule was found for the TIM-barrel fold previously.

Clearly, the reduction in the number of exceptions due to insertions and deletions enables ILP to learn global fold properties but it remains to be seen if those properties can be recognised by protein experts as the important features of the fold.

Comparison to a manual standard (SCOP)

Rossmann fold

In many cases, ILP recalls those properties that are noted by the curators of SCOP. For example, the ILP rules found for the Rossmann fold and SCOP description both give the topology of the main parallel β-sheet as an important feature. ILP also describes a helix at core position "b" (the second core element in the sequence) containing glycines in the middle and n-terminal sections. This is part of a conserved G-X-G-X-X-G sequence motif[17] involved in binding nucleotide groups. While SCOP mentions nucleotide binding in its description of the Rossmann fold, it does not explicitly detail the features involved in binding.

Immunoglobulin fold

For some folds, ILP rules describe the folds in more detail than the SCOP database. SCOP describes Immunoglobulin folds as having 7 strands in 2 sheets, with some variation. This is very similar to the descriptions for a number of other folds in the same main fold class (all-β), including the Prealbumin-like fold (see Table 2). While these descriptions match each fold class, they do not identify the features that distinguish them from one another. In contrast, the rules learnt for the core of the Immunoglobulin fold using ILP include a level of detail necessary to distinguish it from other fold classes. The ILP rule given in Table 2 describes an Immunoglobulin fold as having two antiparallel sheets, one with 4 strands and the other with 3 strands,

in agreement with SCOP. However, it also includes the topology of those sheets (2134 and 123, respectively). This clearly distinguishes this fold from the Prealbumin fold, which contains a sheet with 3 core strands that is mixed and has the first strand in the centre of the sheet (topology 213). The Immunoglobulin fold class is known to contain substantial variation between individual structures[18]. Even with the use of multiple structure alignments, ILP finds several rules (including the rule discussed above) describing the core structure of this fold (http://www.sbg.bio.ic.ac.uk/~cootes/rules.html). While the variation amongst Immunoglobulin structures is referred to by the SCOP description (see Table 2), explicit details are not given. The expert responsible for the classifications construction is likely to know these details but has not clearly articulated them for the non-expert to analyse. While such details may yet be included in subsequent versions of SCOP, providing such levels of detail manually will become increasingly difficult as large numbers of structures are produced by structural genomics projects.

TIM-barrel fold

For other cases, ILP rules do not reflect the same level of detail contained in the corresponding SCOP description. For example, the ILP rule describing the TIM barrel fold given in Table 2 includes the size of the main parallel sheet and the total number of core helices (not described by SCOP). However, the ILP rules give no information about the topology of the β-sheet and do not identify the sheet as a closed barrel (both described by SCOP). In contrast to the SCOP description, a recent study has shown that a large number of TIM barrels are not in fact fully closed barrel structures[19].

Thus, we might expect an objective description to include both “open” and closed barrel structures. The end_strand_distance predicate (see Table 4) was included in this study to try and detect such “open” barrels, in which the sheet is highly curved with the end strands close together in space but not in contact. However, no rules were learnt describing such “open” barrels either. The reason that such features were not learnt for the TIM-barrel lies in the way that ILP generates fold rules. ILP searches for the shortest rule that distinguishes the fold of interest from all other folds in the same main fold class. Very few proteins in the α/β main fold class have parallel sheets consisting of 8 core strands (data not shown) apart from the TIM-barrels. Hence, a rule describing a TIM-barrel as an 8-stranded parallel sheet does not need to include many more details in order to explain why that fold is unique.

SH3 fold

Included among the structural properties that folds were examined for were the presence of glycines and prolines in secondary structure elements. These residues are particularly interesting in terms of structure in that they greatly increase and decrease backbone conformational freedom respectively. It might be expected that there are some structural contexts in which these residues are strongly preferred in order for a protein to adopt a given fold.

One such fold may be the SH3 fold. The rule learnt automatically using ILP in this study described this fold as including a four-stranded antiparallel sheet, with the first and fourth strands at the end of the sheet being close together in space so that the

sheet formed an open barrel structure (Table 2). These features are also described, or indirectly implied, by the corresponding SCOP description. However, the Progol rule also describes the presence of a proline at the end of the fourth strand (Figure 2d) that is not given by the SCOP description. This proline occurs just before the 3-10 helix interrupting the fourth strand in the sheet (referred to by the SCOP description) in SH3 domains from three different superfamilies. The proline, with its peculiar property of highly restricted backbone conformational freedom, might be strongly preferred in order to form the break in the regular secondary structure of the fourth strand. It might also play a role in ensuring that the SH3 barrel is an open one, as a proline cannot contribute to hydrogen bonding to two neighbouring strands (which would be required in order for the barrel to be closed).

Barrel-sandwich hybrid fold

For other folds, a glycine may be required in order to give protein chain the necessary conformational freedom to form the required structure. The rule learnt in this study for the Barrel-sandwich hybrid fold describes an antiparallel sheet with topology 3214, in which the n-terminal portion of the first strand contains a glycine. This feature is found in domains with this fold from three different superfamilies. Example domains from two of these superfamilies exhibit phi/psi angles for the glycine of interest that are outside the normal range for a residue within a beta-strand ($-60 < \text{phi} < -150$, $90 < \text{psi} < 180$). This indicates that a glycine might be preferred at this structural position in order to achieve an unusual bend in the strand and sheet. SCOP does not describe the presence of this glycine or indeed the topology of the sheet.

Long alpha-hairpin fold

The presence of a particular residue that ILP has indicated to be important is more difficult to interpret structurally for other folds. One such example is the Long alpha-hairpin fold, in which the derived rule indicated that a glycine in the middle of the second of two helices is an important feature for this fold. Domains from four different superfamilies in this fold class exhibit this feature, implying that the glycine is important structurally. However, the glycines have phi/psi angles in the normal range for a residue in an alpha-helix (-60 < phi < -150, -45 < psi < -60) indicating that backbone conformational freedom is not required in order to form this fold.

The rule learnt in this study also describes the way in which the alpha-helices are arranged with respect to one another in a little more detail than SCOP. It does not describe the left-handed twist of the helices referred to SCOP as no attempt was made to learn this type of feature.

Conclusion

The structural principles underpinning much of fold space can be automatically described using ILP. Such rules are objective and are discriminatory with respect to different fold classes by construction. Furthermore, the rules produced can be readily interpreted and analysed by human experts in protein structure. In contrast to previous

work[16], the incorporation of structural superpositions into the background knowledge employed here has enabled the global properties of a fold to be captured effectively. Using this approach, the rules obtained often reflect the principles given in the manual fold descriptions of the SCOP database but are derived in a completely objective way. In some cases, folds in the SCOP database with ambiguous or non-discriminating descriptions (for example, the Immunoglobulin and Prealbumin-like folds) can be distinguished using the rules learnt with ILP.

This study has concentrated on automatically learning structural principles from the manually curated SCOP database. However, the approach used here can be applied generally to any classification of protein structure. Several classification schemes already exist that employ semi-[4] or fully-automated[5] methods of protein structure comparison. While these schemes and SCOP are largely similar, they have also been shown to differ significantly[7]. The application of the ILP scheme used in this study to different classification schemes would enable the principles behind their respective fold categories to be compared objectively using language comprehensible to a protein expert. More importantly, ILP could be used as part of a two-step, fully automatic approach to derive the principles of protein structure from coordinates produced by structural genomics. The first step of fully automated protein structure classification can be achieved with schemes such as DALI[5]. Here, the challenging second step, of objectively deriving the structural principles behind such a classification, has been demonstrated using ILP. More generally, given the increasing emphasis on high-throughput experimental projects, machine-learning techniques such as ILP are going to become crucial to learning principles from biological data.

Methods

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 list of domains for each of the four main fold classes (all α, all β, α/β and α+β) were selected using the ASTRAL[20] database, selecting one domain per protein/species. This list of domains thus included some related domains (that is, domains 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). When testing the rules, one representative domain per SCOP sequence family was selected randomly in order to eliminate bias.

The 45 folds considered in this study correspond to those SCOP fold categories in which protein domains from 4 or more sequence families could be clustered in a multiple structure alignment (see next section). Two fold categories (TIM barrels and Immunoglobulins) have a very large number of examples from the ASTRAL set. For these folds, 50 positive examples were selected randomly with a weighting to ensure equal preference for each sequence family.

Multiple structure alignment

A brief outline of the techniques used to generate multiple structure alignments and define the core secondary structure elements is given in this and the next section. A more detailed explanation of these methods can be found elsewhere[21].

Multiple alignments were constructed by clustering pairwise alignments of domains with the same fold. Pairwise alignments were generated for each possible pair of domains in that category using the SSAP program[22]. The pairwise alignments in each fold category were then clustered with respect to their Root Mean Square Distance (RMSD), in a similar manner to that of a previous publication[23], to give the final multiple structure alignment. Firstly, a master domain was selected by finding the domain with the lowest average pairwise RMSD to all other domains in that SCOP 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 alignment with the master domain with RMSD > 6 Å were firstly eliminated from further consideration. Also, in order to avoid corrupting the multiple alignment with misaligned pairwise alignments, domains for which less than 2/3 of the residues participated in the multiple alignment at any given step were eliminated from consideration. For most fold categories considered here, only a few domains were eliminated in this way.

Core secondary structure element definition

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 conserved (core) secondary structure elements (α-helical or β-strand) the elements that can be deemed equivalent have to be identified. To do this, the secondary structure for each protein in the multiple structure alignment was defined using the PROMOTIF[24] program. Then a simple matching scheme was employed to match secondary structures units in the different domains based on the extent to which their constituent residues are structurally equivalent, as determined by the multiple alignment for that fold category. Those elements that are equivalent to another element in the majority of aligned domains are considered to be a core element. The groups of equivalent, core secondary structure elements were labelled according to their relative position in the sequence (that is, the first group were labelled "a", the second "b" and so on).

Learning rules with Progol

Rules were learnt using the Progol-4.4 ILP system[8; 10]. The positive examples for a given fold were those domains that were clustered to form the multiple structure alignment for that SCOP fold category. The negative examples were those domains that formed the multiple structure alignments in all other fold categories in the same SCOP main fold class (all-α, all-β, α/β or α+β). Thus, rules were learnt to discriminate between the fold type of interest and the most similar, yet distinct, types of fold. Background information for each example was generated from the

PROMOTIF output for those secondary structure elements that were defined to be core (Table 4).

Progol parameters

The maximum number of nodes (or hypotheses) tested for an individual search was set to 1000. The noise parameter was set to 20% (that is, up to 20% of examples covered by a rule could be false positives). The inflate parameter was set to 200% (that is, positive examples were given a weighting twice that of the negative examples). In addition to these parameters, a constraint was applied to force the rules to include information about each secondary structure element other than its relative position in the sequence. For example, a rule stating "Fold A has an α-helix B at core position 'b' and an α-helix C at core position 'c'." would not be considered a viable rule. However, a rule stating "Fold A has an α-helix B at core position 'b' and an α-helix C at core position 'c', B and C are in contact." would be considered valid. This constraint was applied to enrich the rules in terms of their biological description and insight.

Cross-validation testing

5-fold cross-validation was carried out on all 45 folds considered in this study. Domains for each fold category were divided into learning and test sets such that no domain in the test set was related to (that is, in the same SCOP sequence family as) any domain in the learning set. Rules were learnt on the learning set as described

above. The test set was then included and the multiple structure alignment re-calculated. One example per sequence family was randomly selected from the aligned members of the test set for testing. If no example from a sequence family was aligned, this sequence family was ignored. However, even if these test examples that could not be aligned were included as false negatives, the overall result for the 45 folds was still statistically significant (data not shown).

Acknowledgements

This work was supported by a BBSRC grant. The authors thank Suhail Islam for his assistance in preparing figures.

Figure captions

Figure 1. Information flow in ILP. ILP is driven by examples and background knowledge to produce new rules and principles. Examples of a given fold are taken from the SCOP database. Background knowledge is generated from structurally aligned protein coordinates and general structural principles defined by an expert.

Figure 2. Structures demonstrating features learnt using ILP. The features highlighted correspond to the rules learnt using ILP in this study, given in Table 2. The numbering of strands shows sheet topology, relevant glycines are highlighted in green

and prolines in red. The structures shown have the following folds: (a) TIM barrel-like, (b) Immunoglobulin-like, (c) Rossmann-like, (d) SH3-like, (e) Barrel-sandwich hybrid and (f) Long alpha-hairpin.

References

1. Orengo, C. A., Jones, D. T. & Thornton, J. M. (1994). Protein superfamilies and domain superfolds. *Nature* 372, 631-634.
2. Chothia, C. (1992). One thousand families for the molecular biologist. *Nature* 357, 543-544.
3. LoConte, L., Ailey, B., Hubbard, T. J. P., Brenner, S. E., Murzin, A. G. & 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. & Orengo, C. A. (2000). Assigning genome sequences to CATH. *Nucleic Acids Research* 28, 277-282.
5. Holm, L. & Sander, C. (1998). Touring protein fold space with DALI/FSSP. *Nucleic Acids Research* 26, 316-319.
6. Taylor, W. R. (2002). A ' periodic table' for protein structures. *Nature* 416, 657-660.
7. Hadley, C. & Jones, D. T. (1999). A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. *Structure Folding and Design* 7, 1099-1112.
8. Muggleton, S. H. (1992). *Inductive Logic Programming*, Academic Press, London.
9. Muggleton, S. H. & Raedt, L. D. (1994). Inductive logic programming: theory and methods. *Journal of Logic Programming* 19/20, 629-679.
10. Muggleton, S. H. (1995). Inverse entailment and progol. *New Generation Computing Journal* 13, 245-286.
11. King, R. D., Muggleton, S. H., Srinivasan, A. & 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. *Proceedings of the National Academy of Sciences of the United States of America* 93, 438-442.
12. Hirst, J. D., King, R. D. & 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, 405-420.
13. Muggleton, S. H., King, R. D. & Sternberg, M. J. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Engineering* 5.
14. King, R. D., Clark, D. A., Shirazi, J. & 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.
15. King, R. D., Muggleton, S. H., Lewis, R. A. & 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. *Proceedings of the National Academy of Sciences of the United States of America* 89, 11322-11326.
16. Turcotte, M., Muggleton, S. H. & Sternberg, M. J. (2001). Automated discovery of structural signatures of protein fold and function. *Journal of Molecular Biology* 306, 591-605.
17. Wierenga, R. K., Terpstra, P. & Hol, W. G. J. (1986). Prediction of the occurrence of the ADP-binding b-a-b-fold in proteins. *Journal of Molecular Biology* 187, 101-107.
18. Bork, P., Holm, L. & Sander, C. (1994). The Immunoglobulin fold. *Journal of Molecular Biology* 242, 309-320.
19. Nagano, N., Hutchinson, E. G. & Thornton, J. M. (1999). Barrel structures in proteins: automatic identification and classification including a sequence analysis of TIM barrels. *Protein Science* 8.
20. Brenner, S. E., Koehl, P. & Levitt, M. (2000). The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Research* 28, 254-256.
21. Cootes, A. P., Muggleton, S. H., Greaves, R. B. & Sternberg, M. J. (2002). Automatic determination of protein fold signatures from structural superpositions. *Electronic Transactions on Artificial Intelligence* 5, 245-274.
22. Taylor, W. R. & Orengo, C. A. (1989). Protein structure alignment. *Journal of Molecular Biology* 208, 1-22.
23. Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. E. (2000). Enhanced genome annotation using structural profiles in the program 3D-PSSM. *Journal of Molecular Biology* 299, 499-520.
24. Hutchinson, E. G. & Thornton, J. M. (1996). PROMOTIF - a program to identify and analyze structural motifs in proteins. *Protein Science* 5, 212-220.

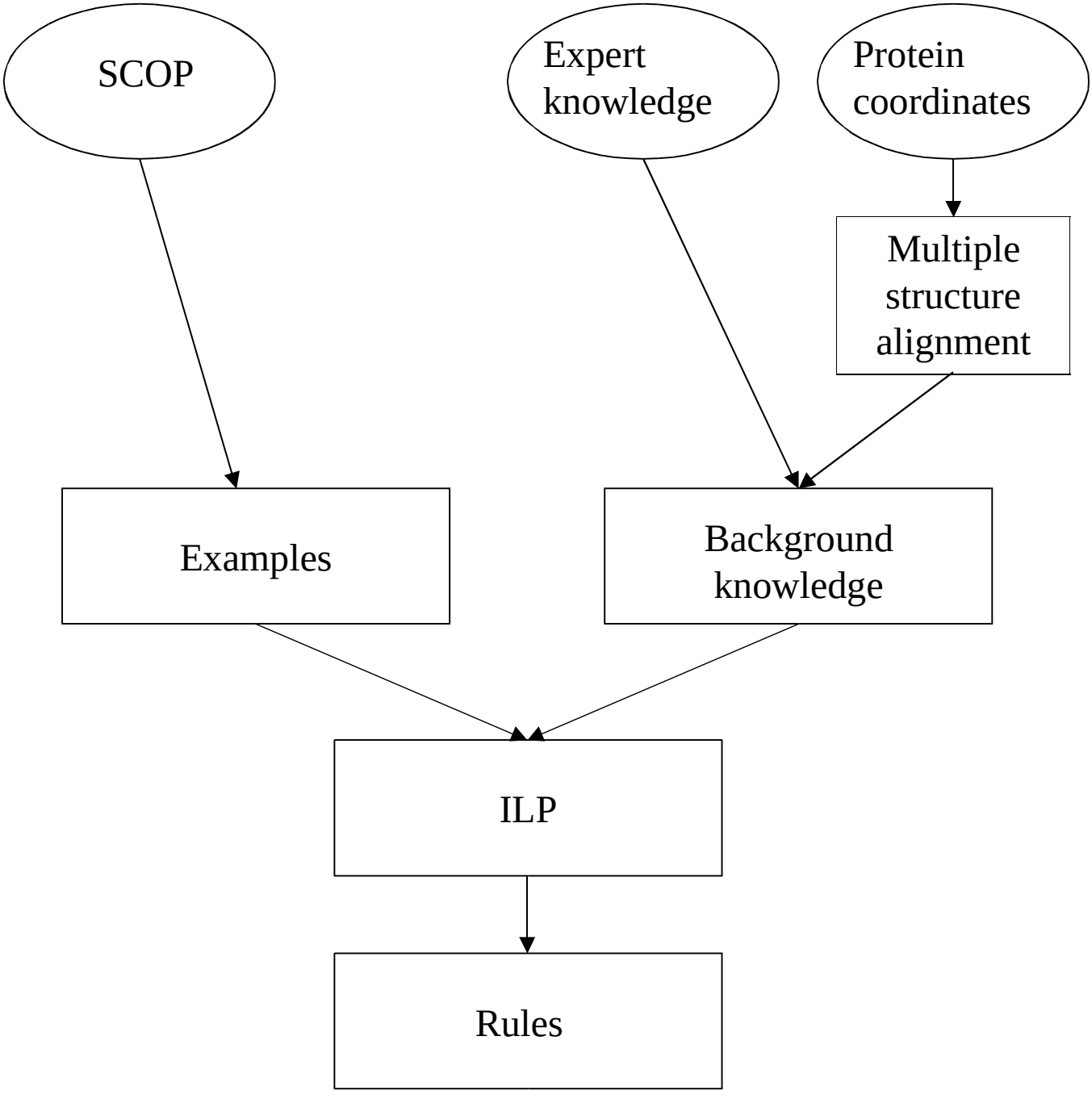
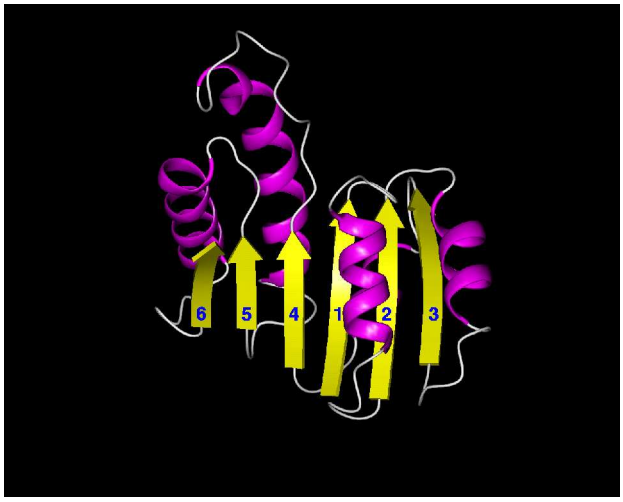
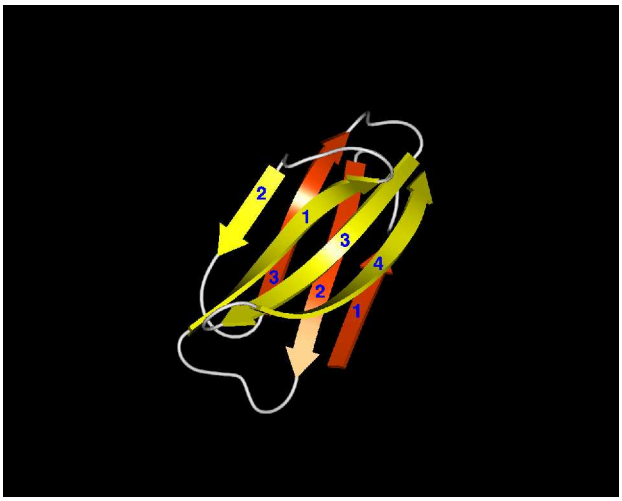

SCOP
Expert knowledge
Protein coordinates
Multiple structure alignment
Examples
Background knowledge
ILP
Rules

Figure 1

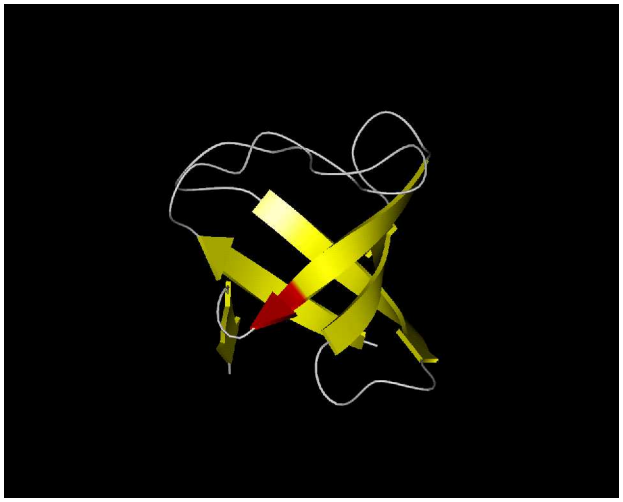
(a)



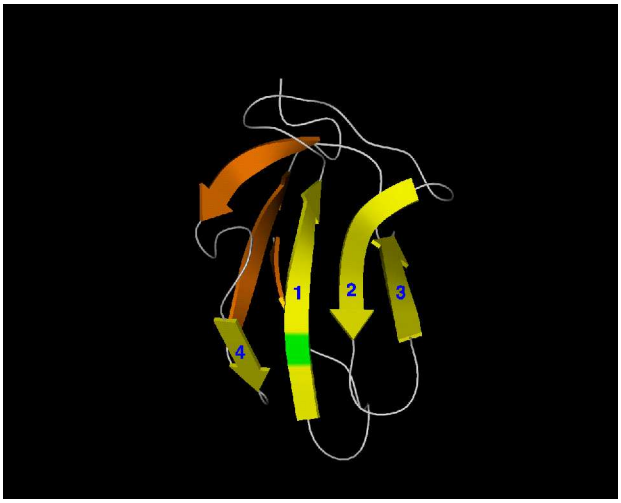
(b)



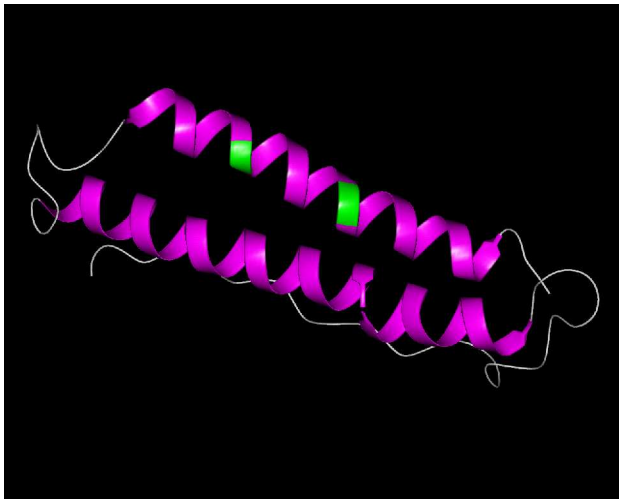
(c)



(d)



(e)



(f)

Figure 2

Table 1. Cross-validated accuracy for fold rules. The number of positive examples (+), number of negative examples (-), accuracy, expected accuracy, precision and recall statistics are given for each fold. The expected accuracy is the accuracy that would be obtained if every example were predicted to be a negative example. Recall is the percentage of positive examples that have been correctly predicted to have that fold. Precision is the percentage of examples predicted to have that fold that have been predicted correctly. The overall accuracy for these 45 folds was found to be statistically significant ($p << 0.01$) according to a ^{2} test.

Fold class	+	-	Accuracy	Expected	Precision	Recall
Long alpha-hairpin	7	146	95%	95%	50%	29%
DNA/RNA-binding 3-helical bundle	30	123	97%	80%	96%	90%
Four-helical up-and-down bundle	10	143	96%	93%	75%	60%
EF Hand-like	9	144	95%	94%	56%	56%
SAM domain-like	10	143	95%	93%	100%	20%
Alpha/alpha toroid	5	148	97%	97%	0%	0%
Alpha-alpha superhelix	8	145	93%	95%	33%	25%
Multiheme cytochromes	4	149	97%	97%	0%	0%
All-α class	83	1141	96%	93%	76%	53%
Immunoglobulin-like beta-sandwich	16	129	90%	89%	53%	62%
Diptheria toxin/transcription factors/cytochrome f	7	138	97%	95%	100%	29%
Prealbumin-like	4	141	99%	97%	100%	75%
Crystallins/protein S/yeast killer toxin	4	141	98%	97%	100%	25%
Galactose-binding domain-like	7	138	95%	95%	50%	14%
ConA-like lectins/glucanases	5	140	92%	97%	12%	20%
SH3-like barrel	7	138	94%	95%	40%	29%
OB-fold	12	133	97%	92%	100%	58%
Beta-Trefoil	6	139	97%	96%	100%	33%
Reductase/isomerase/elongation factor	7	138	97%	95%	100%	43%
PH domain-like	4	141	99%	97%	80%	100%
7-bladed beta-propeller	6	139	97%	96%	100%	33%
Double-stranded beta-helix	5	140	97%	97%	50%	20%
Barrel-sandwich hybrid	4	141	98%	97%	60%	75%
All-β class	94	1936	96%	95%	64%	45%
TIM beta/alpha-barrel	30	135	91%	82%	80%	67%
NAD(P)-binding Rossmann-fold domains	6	159	99%	96%	100%	83%
Flavodoxin-like	15	150	95%	91%	82%	60%
Ferredoxin reductase-like	4	161	99%	98%	100%	50%
Adenine nucleotide alpha hydrolase	4	161	96%	98%	0%	0%

Biotin carboxylase N-terminal domain-like	5	160	98%	97%	67%	40%
DHS-like NAD/FAD-binding domain	4	161	97%	98%	0%	0%
Thiamin-binding	4	161	98%	98%	0%	0%
Thioredoxin fold	6	159	98%	96%	67%	67%
Restriction endonuclease-like	4	161	97%	98%	33%	25%
Ribonuclease H-like motif	5	160	97%	97%	50%	20%
S-aden.-L-meth.-dependent methyltransferases	5	160	98%	97%	100%	20%
PLP-dependent transferases	5	160	100%	97%	100%	100%
Alpha/beta-Hydrolases	17	148	96%	90%	92%	71%
α/β class	114	2196	97%	95%	78%	54%
Lysozyme-like	5	156	98%	97%	100%	20%
Beta-Grasp (ubiquitin-like)	8	153	98%	95%	75%	75%
FAD-linked reductases, C-terminal domain	6	155	99%	96%	100%	67%
Cystatin-like	7	154	99%	96%	100%	86%
Ferredoxin-like	32	129	96%	80%	96%	84%
Zincin-like	7	154	99%	96%	100%	86%
T-fold	4	157	98%	98%	50%	25%
TBP-like	5	156	98%	97%	100%	40%
ATP-grasp	4	157	99%	98%	100%	50%
α+β class	78	1371	98%	95%	93%	71%
Total	369	6644	97%	95%	77%	55%

Table 2. Comparison of ILP rules to SCOP descriptions for several folds. Some of the rules learnt using ILP are compared to the expert-like descriptions of those folds taken from the SCOP database (SCOP). The ILP rules are written in English for ease of comparison with the manual SCOP descriptions.

SCOP fold class (version 1.50)	Rule type	Rule
Immunoglobulin (1 002 001)	SCOP	sandwich; 7 strands in 2 sheets; greek-key; some members of the fold have additional strands
	ILP	Has antiparallel sheets B and C; B has 3 strands, topology 123; C has 4 strands, topology 2134.
Prealbumin-like (1 002 003)	SCOP	Sandwich; 7 strands in 2 sheets, greek-key; variations: some members have additional 1-2 strands to common fold
	ILP	Has a mixed sheet B. B has 3 strands with topology 213.
TIM barrel (1 003 001)	SCOP	contains parallel beta-sheet barrel, closed; n=8, S=8; strand order 12345678; the first six superfamilies have similar phosphate-binding sites
	ILP	Has between 5 and 9 helices; Has a parallel sheet of 8 strands.

Rossmann-like (1 003 002)	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	Has between 3 and 4 helices; Has α-helix B as the second core element in the sequence; 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 as the seventh and the ninth core elements in the sequence respectively; C and D are in contact and parallel.
SH3 (1 002 001)	SCOP	barrel, partly opened; n*=4, S*=8; meander the last strand is interrupted by a turn of 3-10 helix
	ILP	Has an antiparallel sheet B. C and D are the 1st and 4th strands in the sheet B respectively. C and D are the end strands of B and are 4.360 (+/- 2.18) angstroms apart. D contains a proline in the c-terminal end.
Barrel-sandwich hybrid (1 002 079)	SCOP	sandwich of half-barrel shaped beta-sheets
	ILP	Has an antiparallel sheet B. B has 4 strands with topology 3214. C and D are the 1st and 4th strands in B respectively. C and D are in contact. C contains a glycine in the n-terminal end.

Long alpha-hairpin (1 001 002)	SCOP	2 helices; antiparallel hairpin, left-handed twist
	ILP	Has a total number of 2 helices. α-helices B and C are the 1st and 2nd core elements in the sequence respectively. B and C are in contact, the closest points are the middle of B and the middle of C. B and C are antiparallel (180 +/- 45 degrees) to one another. C contains a glycine in the middle region.

Table 3. Rules learnt with ILP in the absence of multiple structure alignments. The rule corresponding to each fold in Table 2 is taken directly from a previous publication[16] (where available). Several folds in Table 2 were not considered in the previous study and are absent from the table below.

SCOP fold class	Rule
Immunoglobulin	There is at most one helix, the loop between the 5th and 6th strands is three to seven residues long.
TIM barrel	No rule given. There was no rule with > 30% coverage found in the previous study.
Rossmann-like	The 1st strand is followed by a helix, the two elements are separated by a coil of about one residue. The 6th strand is followed by a helix.
SH3	There are four to six strands, the loop between the 3rd and 4th strand is one to three residues long.

Table 4. Predicates describing protein fold properties. Each predicate is a logic expression in Prolog describing attributes of, or relationships between, core secondary structure elements in a protein domain.

Predicate	Description
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 a 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.

Table 5. Occurrence of predicates in rules.

Predicate	Number of occurrences
sheet	56
helix	53
strand_position	37
sheet_top_X	31
number_helices	24
contact	24
helix_angle	11
contains	10
pair	7
end_strand_distance	5
coil	4
has_n_strands	4
antiparallel	2
parallel	2
barrel	1
adjacent	0
bifurcated	0
total	271