doi:10.1006/jmbi.2000.4414 available online at http://www.idealibrary.com on IDEAL® J. Mol. Biol. (2001) **306**, 591–605

JMB

Automated Discovery of Structural Signatures of Protein Fold and Function

Marcel Turcotte[1], Stephen H. Muggleton[2] and Michael J. E. Sternberg[1]*

[1] *Imperial Cancer Research Fund, Biomolecular Modelling Laboratory, P.O. Box 123 London, WC2A 3PX, UK*

[2] *University of York Department of Computer Science, Heslington, York YO1 5DD UK*

*Corresponding author

There are constraints on a protein sequence/structure for it to adopt a particular fold. These constraints could be either a local signature involving particular sequences or arrangements of secondary structure or a global signature involving features along the entire chain. To search systematically for protein fold signatures, we have explored the use of Inductive Logic Programming (ILP). ILP is a machine learning technique which derives rules from observation and encoded principles. The derived rules are readily interpreted in terms of concepts used by experts. For 20 populated folds in SCOP, 59 rules were found automatically. The accuracy of these rules, which is defined as the number of true positive plus true negative over the total number of examples, is 74 % (cross-validated value). Further analysis was carried out for 23 signatures covering 30 % or more positive examples of a particular fold. The work showed that signatures of protein folds exist, about half of rules discovered automatically coincide with the level of fold in the SCOP classification. Other signatures correspond to homologous family and may be the consequence of a functional requirement. Examination of the rules shows that many correspond to established principles published in specific literature. However, in general, the list of signatures is not part of standard biological databases of protein patterns. We find that the length of the loops makes an important contribution to the signatures, suggesting that this is an important determinant of the identity of protein folds. With the expansion in the number of determined protein structures, stimulated by structural genomics initiatives, there will be an increased need for automated methods to extract principles of protein folding from coordinates.

Keywords: protein structure; protein function; knowledge discovery; folding

Introduction

Within the collection of determined protein structures, there is a wealth of principles governing the complex sequence-conformation-function relationships. Historically, many of these principles have been identified by extensive human examination. However, there is now a rapid expansion in the number of determined protein structures due to structural genomics projects[1] combined with improvements in the experimental methodology.[2] Computational methods will therefore be required to extract structural features that are common to a particular tertiary fold or to a function. This paper explores the use of Inductive Logic Programming (ILP),[3] a form of machine learning to extract automatically local three-dimensional signatures that are specific to a particular protein fold due to a stereochemical or a functional requirement.

Protein domains adopt a common fold if there is a similar sequential and three-dimensional arrangement of their regular α- and β-secondary structures. Recently several classification of protein structures such as SCOP (Structural Classification of Proteins),[4] CATH (which clusters proteins at four major levels, Class, Architecture, Topology and Homologous superfamily)[5] and FSSP (Fold classification based on Structure-Structure alignment of Proteins)[6] have been developed that identify proteins with common folds. In SCOP, which is

Abbreviations used: ILP, inductive logic programming; SCOP, structural classification of proteins.

E-mail address of the corresponding author: M.Sternberg@icrf.icnet.uk

0022-2836/01/030591–15 $35.00/0

the scheme considered here, proteins of the same fold are subdivided into superfamilies which represent a grouping of all proteins that are homologous (i.e. diverged from a common ancestor). Proteins from different superfamilies within the same fold are presumed to be analogues that converged to a stable folding arrangement. The basis for this classification scheme is the protein expert A. Murzin who considers the evolutionary evidence provided by the sequences, structures and functions of proteins.

There are several constraints on a protein sequence/structure in order for it to adopt a particular fold. Firstly, there is the thermodynamic stability of the final structure. For example, there could be a requirement for certain lengths of secondary structures to form a stable protein core.[7–9] There can also be constraints on the folding pathway and certain local substructures might act as efficient nucleation sites.[10–12] In addition to structural needs, the required function can dictate a common structural feature, particularly for proteins from the same superfamily.[13] These constraints can be considered as global or local. A global signature should identify a limited number of important residues or structural arrangements. Global constraints should be more focussed than just a sequence profile that is generated from a multiple alignment. Local features, which are the focus of this paper, relate to a short region that may involve a particular sequence or arrangement of secondary structures.

Sequence signatures are a well characterised feature of proteins (e.g. Prosite[14] and Prints[15]). These sequence patterns were primarily derived from visual inspection of aligned sequences. Automated methods of discovery can also be used to identify sequence patterns using a variety of approaches including unsupervised searching for commonly occurring sequence words,[16–18] Gibbs sampling[19] and the use of pattern graphs.[20]

Structural patterns are harder to classify but have also been identified. Kasuya *et al.*[13] started with Prosite sequence patterns and mapped them onto the protein coordinates to identify the local structure corresponding to the sequence motif. Other groups[21–23] have focussed on conserved arrangements of sequentially distant residues, particularly of their side-chain atoms, to discover automatically global signatures of residues that form protein active sites. Mirny *et al.*[24] performed structural superpositions to identify global patterns of key residues in the five most populated protein folds. There are, however, many structural local signatures that have been discovered primarily by manual inspection, e.g. features of the globin fold[25] and of the Rossmann fold.[26]

A powerful demonstration of the presence of such local structural signatures was the success of Murzin *et al.*[27] in using these signatures to predict successfully protein folds from sequence during the blind trial known as CASP2 (the second Critical Assessment of techniques for protein Structure Prediction). However, knowledge of these structural signatures primarily relies on human expertise which is based on in-depth analysis together with knowledge of the relevant literature. Indeed, to our knowledge, there is no single source that documents these structural signatures. With the increase in the number of protein structures, automated methods are required therefore to identify systematically structural signatures.

Machine learning techniques, such as artificial neural networks and hidden Markov models, have been applied to study several problems of molecular biology.[28] Those techniques have been particularly successful in analysing sequence data but methods to study the three-dimensional structure are much less developed. One factor limiting their application is the ability to model long range interactions.

We present an application of Inductive Logic Programming (ILP) to learn rules relating local structures to the concept of fold defined by SCOP. ILP algorithms automatically derive rules from examples and background knowledge. Previously ILP has been applied to several structural molecular biology problems including protein secondary structure prediction,[29] drug design,[30,31] packing of beta-strands,[32] and chemical mutagenesis.[33] Several features suggest it may be particularly well suited to study problems encountered in protein structure. Firstly, structures are the result of complex interactions between sub-structures, and the ability of ILP algorithms to learn relations may prove to be a key feature, for example to model the interaction of two consecutive secondary structures. Secondly, ILP systems can make use of problem-specific background knowledge. Vast amounts of knowledge have been accumulated over the years of research on protein structure and can be used effectively. Thirdly, ILP is a logic-based approach to machine learning and the formalism of logic is very expressive. The hypotheses (rules) are easily amenable to human interpretation. Thus, we examine here whether ILP can discover principles of protein fold and function.

Approach

In this application, sets of rules were learnt separately for each fold. A machine learning task therefore consist in learning rules for a fold from examples and background knowledge (see Figure 1). Herein, positive and negative examples were used. A list of positive examples was derived from SCOP by selecting representative domains for the fold under study. A list of counter (negative) examples was also derived from SCOP, but this time by selecting domains from different folds of the same structural class (all-α, all-β, α/β and $\alpha + \beta$), see below for details of the selection process. The background knowledge contained structural information about the positive and negative examples and also general principles, such as how

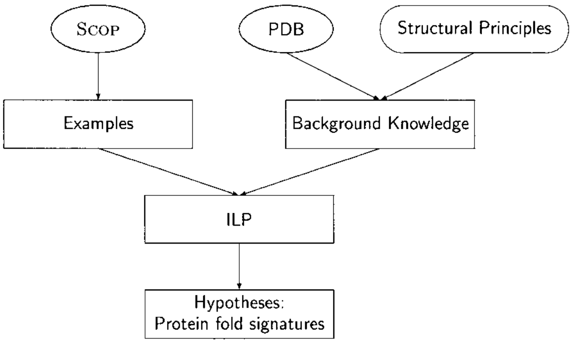

Figure 1. Flow of information. ILP uses background knowledge and examples to derive new rules.

to calculate the hydrophobic moment of a secondary structure element.

For instance, in the globin fold, the list of positive examples comprised 13 domains which includes haemoglobin I, myoglobin and phycocyanin. Although Progol can learn from positive examples only, negative examples were included, since they were readily available; the list of counter examples for the globins includes cytochrome *c*, four-helical cytokines and cyclin. For each example, positive and negative, structural information was derived. This includes attributes, such as the total number of residues, but also relational information, such as the adjacency of the secondary structures, and local information, such as the average hydrophobicity of each secondary structure element and the presence of proline residues. See below for a complete description.

To learn a rule, Progol selects a positive example and extracts all the related information from the background knowledge. This information is used in a combinatorial search that incrementally builds more specific rules until it finds one that maximises its measure of compression. Specifically, the measure of compression, equation (1), seeks to maximise the number of positive examples covered (p), minimises the number of negative examples covered (n), while minimising the length of the rule (c):

$$f = p - n - c \qquad (1)$$

Given two rules covering the same number of positive and negative examples, this measure favours the shortest one, i.e. the one that obeys the principle of parsimony or Occam's razor. Once an optimal rule has been found, the positive examples it matches are removed from the database and the algorithm resumes with the next available positive example. The process continues until no positive example remains. At this point, Progol has found a set of rules that represents all the positive examples.

Our study was restricted to the five most populated folds of each of the four main classes (see Table 1). This ensured that each fold contained a sufficient number of examples. Since the folds are well populated this also gives a good coverage of the domains found in SCOP. Indeed, the subset comprises 381 domains and therefore represents 30 % of the total number of domains found in the four main classes.

Results

Protein signatures

A total of 59 signatures were derived automatically for 20 populated folds in SCOP (see Table 1). The complete list of rules can also be found from our web site `www.bmm.icnet.uk/ilp/fold`. This implies that on average three rules per fold were generated to match all the examples. For the (TIM)-barrels as many as ten rules were needed. In general, the number of rules correlates well with the number of positive examples (correlation coefficient 0.875).

The predicted overall accuracy, defined as the number of correct assignments over the total number of examples, was estimated using standard cross-validation tests. The estimated accuracy is 74.35(±1.31) %, which is statistically better at 99 % level of confidence than random assignments (*t*-test); see later for details. The accuracy for folds of the all-α and all-β classes is slightly better. This might be because proteins of these two classes are in general smaller and less complex than those of the two remaining classes, α/β and $\alpha + \beta$.

Recall is defined as the number of true positives over the total number of positives whilst precision is the number of true positives over true positives plus false positives. The rules with recall of 30 % or more are presented in Table 2, those rules will be referred to as "power rules". In our study, it was impossible to derive rules with recall greater than 17 % for the (TIM)-barrel fold. This fold has also the largest number of superfamilies and families in our data set, 17 and 28, respectively. The combined effects of diversity and complexity of the fold has prevented Progol from finding general signatures. As mentioned above, ten rules were produced for the (TIM)-barrels, at the other end of the scale, a single rule per fold was generated for three folds, lipocalins, P-loop and SH2. Since there were some outliers (false negative) and some domains do not match the rule that was found, the recall ranges from 86 % to 100 % for those folds. The precision is high, ranging from 80 to 100 %. This is in agreement with the predefined parameter noise which was set to 20 %.

In the following sections five signatures are detailed. Those rules were selected because of their particular biological interest. For example, in the Rossmann fold, the local signature is characteristic of nucleotide binding site. The Prolog representation of these signatures and their English trans-

Table 1. Cross-validation results

Fold	Examples	Families	Superfamilies	Rules	FN	% Accuracy
All-α						
DNA 3-helical	30	17	4	4	1	82 ± 4
EF hand-like	14	7	2	2	1	69 ± 7
Globin-like	13	2	1	2	1	95 ± 4
4-Helical cytokines	10	3	1	2	2	73 ± 8
Lambda repressor	10	3	1	2	0	63 ± 9
Other folds (92)	210	139	111	-	-	-
						78 ± 3
All-β						
Ig beta-sandwich	45	12	8	6	3	71 ± 4
Tryp ser proteases	21	4	1	3	1	82 ± 5
OB-fold	20	11	4	3	4	77 ± 5
SH3-like barrel	16	7	6	2	2	77 ± 6
Lipocalins	14	2	1	1	2	79 ± 6
Other folds (56)	220	123	90	-	-	-
						76 ± 2
α/β						
β/α (TIM)-barrel	55	28	17	10	5	66 ± 4
Rossmann-fold	21	7	1	2	3	78 ± 5
P-loop	14	4	1	1	2	81 ± 6
Periplasmic II	13	2	1	3	1	64 ± 8
α/β-Hydrolases	12	10	1	3	0	75 ± 7
Other folds (70)	200	131	88	-	-	-
						71 ± 2
α + β						
Ferredoxin-like	26	21	17	3	2	80 ± 5
Zincin-like	13	8	2	4	1	56 ± 8
SH2-like	13	1	1	1	0	79 ± 6
beta-Grasp	12	6	6	3	1	64 ± 8
Interleukin 8	9	1	1	2	0	85 ± 7
Other folds (96)	240	158	113	-	-	-
						73 ± 3
						74 ± 1

For each fold, the Table lists the number of positive examples, the number of families and superfamilies, the total number of rules automatically derived, the number of false negative examples (FN), i.e. positive examples which are not represented by any of the rules, and the accuracy.

lation are listed in Table 3. This is followed by a discussion of the relationship between the signatures and the SCOP hierarchy.

Globin

The globin fold comprises diverse sequences such as myoglobin, hemoglobin and phycocyanins. Yet the three-dimensional structures of these proteins are similar. One hallmark of this fold is the presence of a conserved proline in helix B. This observation has been reported previously by Bashford *et al.*[25] and was here discovered automatically. This is illustrated in Figure 2(a) where the proline residues (represented as ball-and-stricks) coincide with a sharp bend in the main chain. The signature is present in the two families, globins and phycocyanins, that constitute the globin fold. Progol has also produced a second rule because three domains appeared to be exceptions. Looking back at the data revealed that in the case of myoglobin (PDB 1myg) the first and second helix were merged together and prevented the match. However, in the case of glycera globin (PDB 2hbg) and leghemoglobin (PDB 1bin) the second helix, which is reported as a contiguous yet bent helix in all our examples, was separated in two and precluded the match. If the problems of secondary structure assignment are corrected, the three domains match the general rule as well.

Lambda repressor

The λ repressor-like fold contains small domains that bind DNA in a similar way. The second helix of a helix-turn-helix motif (called the recognition helix) makes sequence-specific contacts with the edge of the base-pairs situated in the major groove of the DNA molecule.[34] Although very important, the network of hydrogen bonds between the recognition helix and DNA is not the only determinant of the specificity. For the well studied proteins 434 repressor[35] and Cro,[36] it has been proposed that the affinity for the central base-pairs of the operator depends on the extensive van der Waals interactions of the loop that follows the helix-turn-helix motif. It is thought that the proteins also recognise the conformation of the sugar-phosphate backbone which, in turn, depends on a specific DNA sequence. A-T base-pairs favour bending toward the minor groove while G-C base-pairs favour bending toward the major groove.[37] This mechanism is referred to as ''indirect readout''.[38] The structure of the eukaryotic Oct-1 POU-specific

Table 2. Protein signatures with recall greater than 30 %

All-α:

DNA-binding three-helical bundle: (recall = 60 %, precisio*n* = 81 %, *n* = 30, fold). The length is 34 to 105 residues, there are exactly three helices, the loop between the 1st and 2nd helix is two to four residues long.

EF hand-like: (recall = 64 %, precision = 90 %, *n* = 14, fold). The 1st strand is followed by a helix, the 2nd strand is immediately followed by a helix.

Globin-like: (recall = 69 %, precision = 100 %, *n* = 13, superfamily/unique). The 1st helix is followed by a 2nd one that contains a proline.

Four-helical cytokines: (recall = 40 %, precision = 80 %, *n* = 10, family). The 2nd strand is immediately followed by a helix (i.e. no coil).

Four-helical cytokines: (recall = 40 %, precision = 80 %, *n* = 10, family). The 1st helix is long and followed by another helix.

λ Repressor-like DNA-binding domains: (recall = 70 %, precision = 88 %, *n* = 10, superfamily/unique). The length varies 53 to 88 residues, the loop between the 3rd and 4th helix is three to nine residues long.

All-β:

Immunoglobulin β-sandwich: (recall = 49 %, precision = 79 %, *n* = 45, fold). There is at most one helix, the loop between the 5th and 6th strands is three to seven residues long.

Trypsin serine proteases: (recall = 57 %, precision = 93 %, *n* = 21 superfamily/unique). The loop between the 12th and 13th strand is four to 12 residues long.

OB-fold: (recall = 35 %, precision = 100 %, *n* = 20, fold). The 1st strand is long and followed by an other strand. The 1st helix is followed by a strand.

OB-fold: (recall = 30 %, precision = 86 %, *n* = 20, fold). There are five or six strands, the loop between the 1st and 2nd strand is two to four residues long, the 4th strand is followed by another strand.

SH3-like barrel: (recall = 69 %, precision = 92 %, *n* = 16, fold). There are four to six strands, the loop between the 3rd and 4th strand is one to three residues long.

Lipocalins: (recall = 86 %, precision = 86 %, *n* = 14, superfamily/unique). The length varies from 110 to 179 residues. The loop between the 7th and following strand is two to four residues long.

α/β class:

NAD(P)-binding Rossmann-fold domains: (recall = 62 %, precision = 100 %, *n* = 21, superfamily/unique). The 1st strand is followed by a helix, the two structures are separated by a short connection, about one residue. Also, the 6th strand is followed by a helix.

P-loop containing nucleotide triphosphate hydrolases: (recall = 86 %, precision = 80 %, *n* = 14, superfamily/unique). The loop which connects 1st strand and following helix is three to seven residues long.

Periplasmic binding protein-like II: (recall = 38 %, precision = 100 %, *n* = 13, superfamily/unique). The loop between the 10th strand and the helix that follows is one residue long.

Periplasmic binding protein-like II: (recall = 31 %, precision = 80 %, *n* = 13, superfamily/unique). The 4th strand contains a proline and is followed by a helix.

α/β-Hydrolases: (recall = 58 %, precisio*n* = 88 %, *n* = 12, superfamily/unique). The loop connecting the 1st strand and the one that follows is one to three residues long and the 7th helix is followed by a strand.

α + β class:

Ferredoxin-like: (recall = 54 %, precisio*n* = 100 %, *n* = 26, fold). There are three to five β-strands, the 1st strand is followed by a helix and the loop between the 2nd helix the strand that follows it is two to four residues long.

Zincin-like: (recall = 31 %, precision = 100 %, *n* = 13, superfamily) The loop between the 2nd and 3rd strand is 2 to 4 residues long, the 2nd helix is followed by a helix.

Zincin-like: (recall = 31 %, precisio*n* = 100 %, *n* = 13, superfamily). The loop between the 1st helix and the following strand is four to ten residues long, the 7th strand is followed by a strand.

SH2-like: (recall = 100 %, precision = 81 %, *n* = 13, family/unique). The length varies from 97 to 116 residues. The loop between the 2nd and 3rd strand is two to four residues long.

β-Grasp: (recall = 42 %, precision = 100 %, *n* = 12, fold). The domain contains the following motif: β2-α1-β3 and the loop between α1 and β3 is two to four residues long.

Interleukin 8-like chemokines: (recall = 78 %, precision = 100 %, *n* = 9, family/unique). The length range from 62 to 78 residues long, the loop which separates the 2nd and 3rd strand is one to three residues long.

The statistics in parentheses are: recall, defined as true positive/total number of positive examples, precision, defined as true positive/total number of prediction, *n* is the total number of positive examples, a final comment indicates if the rule coincides with a fold, a superfamily or a family. Superfamily/unique denotes the fact that a rule coincides with a superfamily but the fold has only one superfamily therefore it is impossible at this time to know if the rule represents the fold or the superfamily.

domain also shows similar contacts in the loop region.[39] The signature generated by Progol shows that the length of this loop is conserved amongst the bacteriophage domains: lambda C1 repressor (1lmb), 434 C1 repressor (2r63), cro 434 (2cro) and P22 C2 repressor (1adr), and also the eukaryotic Oct-1 POU-specific domain (1pou). This region is shown in red in Figure 2(b).

Cytokines

Cytokines regulate the development and activities of many cell types. The four-helical cytokines fold, studied here, contains three families: the long chain cytokines, the short chain cytokines and the interferons/interleukin-10. In our data-set, the interferons/interleukin-10 family contained only one example and it did not match any of the two rules that were found. For the majority of the folds, Progol produced more than one rule to cover all the positive examples. In the cytokines, the mapping of the rules onto the examples coincides with the classification of SCOP into families. One rule describes the long-chain family and its signature highlights the fact that the domains start with a long helix (see Figure 2(c)). The second rule covers the domains of the short-chain family. The

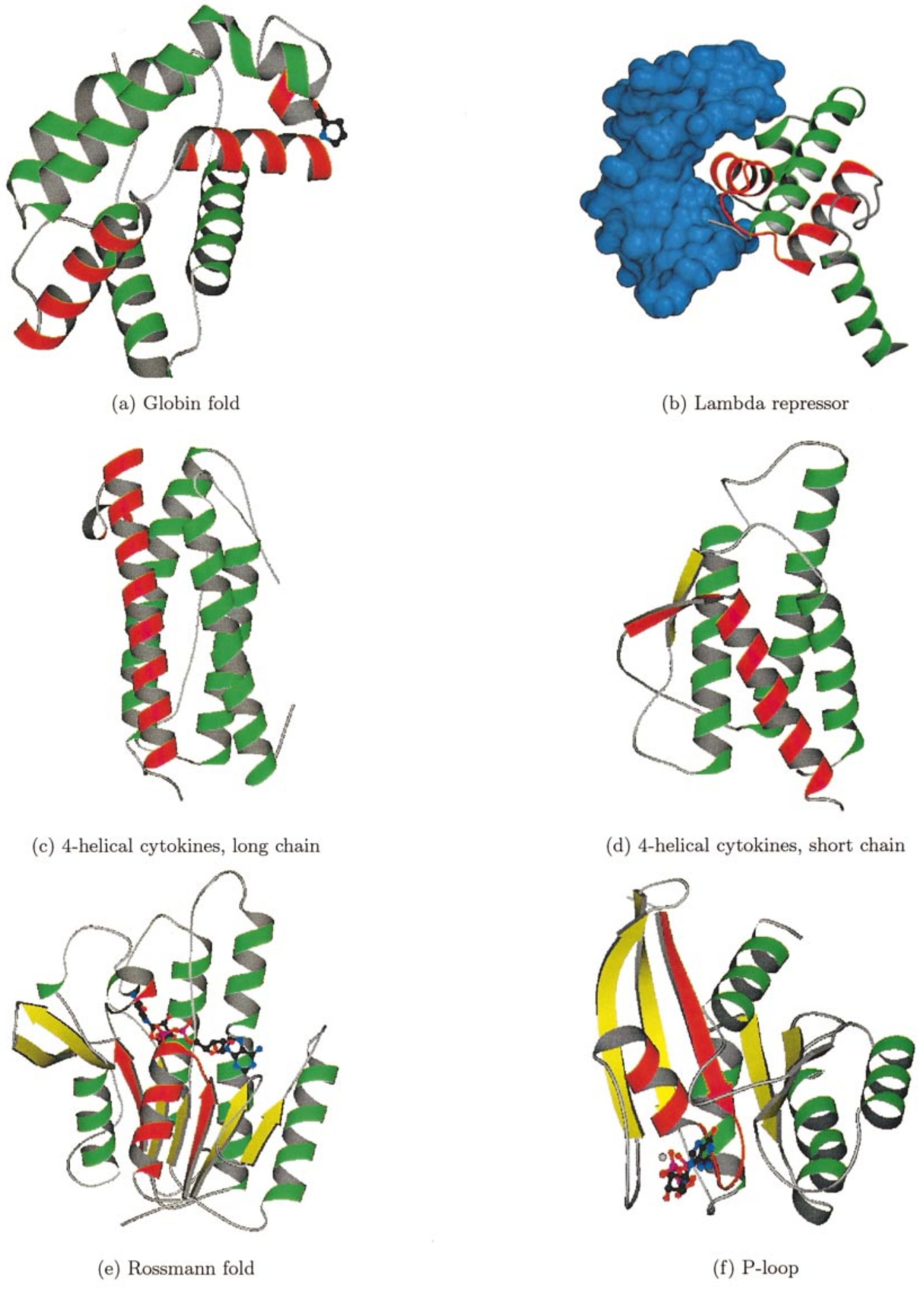
(a) Globin fold
(b) Lambda repressor
(c) 4-helical cytokines, long chain
(d) 4-helical cytokines, short chain
(e) Rossmann fold
(f) P-loop

Figure 2. The Figure illustrates the structural elements making up the rules. Elements involved in structural signatures are shown in red. All the figures were prepared using MOLSCRIPT version 2.156, except that of the lambda repressor which was produced using PREPI (S. Islam unpublished).

distinctive signature in this case is the absence of a coil between the last strand-helix pair (see Figure 2(d)). This fact was used by Rozwarski *et al.*[40] to tether a structural alignment in a comparison of the shot-chain helical cytokines. Looking at the structures reveals that the first residue of the helix also participates in the hydrogen bonding network of the sheet, except for one domain where the sheet is distorted. In IL-4, Kruse *et al.*[41] showed that residues from the C terminus of this helix are involved in either receptor binding or receptor activation.

Table 3. Prolog representation for the six rules presented in the detailed analysis together with the English translation

Rule 1 (Globin fold) Helix A at position 1 is followed by helix B. B contains a proline.

```
fold(’Globin-like’, X) :-
 adjacent(X, A, B, 1, h, h),
 has_pro(B).
```

Rule 2 (lambda repressor) The protein is between 53 and 88 residues long. Helix A at position 3 is followed by helix B. The coil between A and B is about six residues long.

```
fold(’lambda repressor’, X) :-
 total_length(53 =< X =< 88),
 adjacent(X, A, B, 3, h, h),
 length_loop(A, B, 6).
```

Rule 3 (4-helical cytokines) The first helix is long and followed by another helix.

```
fold(’4-helical cytokines’, X) :-
 adjacent(X, A, B, 1, h, h),
 length_sec_structure(A, hi).
```

Rule 4 (4-helical cytokines) The second strand is immediately followed by a helix.

```
fold(’4-helical cytokines’, X) :-
 adjacent(X, A, B, 2, e, h),
 length_loop(A, B, 0).
```

Rule 5 (Rossmann fold) Strand A at position 1 is followed by helix B. Strand C at position 6 is followed by helix D. The length_loop between A and B is about one residue long.

```
fold(’NAD(P)-binding Rossmann-fold’, X) :-
 adjacent(X, A, B, 1, e, h),
 adjacent(X, C, D, 6, e, h),
 length_loop(A, B, 1).
```

Rule 6 (P-loop) Strand A at position 1 is followed by helix B. The coil between A and B is about five residues long.

```
fold(’P-loop’, X) :-
 adjacent(X, A, B, 1, e, h),
 length_loop(A, B, 5).
```

Rossmann fold

NAD-binding domains of the Rossmann fold all have a similar binding mechanism. The adenosine is bound to the short loop between the first strand and the following helix. This region is highly conserved and contains the sequence motif G-X-G-X-X-G.[26] The fifth and sixth secondary structures clamp the nicotinamide moiety of NAD (see Figure 2(e)). The signature discovered by Progol is shared by members of six of the seven families that constitute the unique superfamily of the NAD(P)-binding Rossmann-fold.

P-loop

The P-loop fold comprises three families: the nucleotide and nucleoside kinases, the G proteins and nitrogenase iron protein-like. The first loop is necessary for the proper binding of the guanine nucleotide and gives its name to the fold: the diphosphate-binding loop or P-loop. This region, strand-loop-helix, contains the conserved sequence motif, [AG]-x(4)-G-K-[ST], which is used in the Prosite database[14] to characterise several ATP/GTP-binding proteins (Figure 2(f)). The signature found by Progol matches 12 of the 14 domains found in SCOP, the exceptions are PDB 1dar, which has a shorter loop and PDB 2reb, which contains a large insertion more than 50 residues in length, including two helices and two strands.

Composition of the signatures

We now look at the composition of the signatures. In particular, we investigate the frequency of use of each predicate with a view to understand the fold signatures; see later for a detailed description the exact definitions. The frequency of occurrence for each predicate is presented in Table 4. The frequency of occurrence is skewed. For example, the four most abundant predicates, `adjacent`, `length_loop`, `length_sec_structure` and `total_length`, account for 90 % of all occurrences.

Adjacent is the most frequent and occurs twice as often as the second in the list, `length_loop`. Several factors can explain this. Firstly, the structure of the background knowledge imposes constraints. For the predicates: `length_loop`, `length_sec_structure`, `has_pro`, `average_hydrophobicity` or `hydrophobic_moment`, to occur in a rule there must be at least one adjacent predicate, since those are used to introduce secondary structure identifiers into a rule. Secondly, the predicate `adjacent` is rich in information content as it serves several purposes. As mentioned above, it introduces a pair of secondary structure identifiers into a rule so that local characteristics, such as `length_sec_structure` can be applied. This predicate is used to denote the position of the first element of the pair. Finally, it also serves to define the type of the secondary structures involved. `adjacent` is used more frequently in folds of α/β class, which possibly indicates that other features, such as length, are more variable. Combinations of `adjacent` relations can be used to describe patterns involving three and four secondary structures. We found nine rules of this kind. A relationship involving three secondary structures was generated for the β-grasp fold, although no explicit relationship of three structures was defined in the background knowledge. This highlights the ability of ILP to discover new relationships. In other machine learning tech-

Table 4. Composition of the signatures

Feature	All rules	Power rules
adjacent	80	32
length_loop	42	19
length_sec_structure	15	2
total_length	13	5
number_strands	5	3
number_helices	4	2
has_pro	3	2
average_hydrophobicity	3	0
hydrophobic_moment	2	0

The Table indicates the number of occurrence of each feature in the rules, we distinguish between two subsets: all rules and power rules, the later refers to rules with 30 % or more recall.

niques, such as decision trees, relationships of three structures would have to be listed explicitly, otherwise they would not be discovered. Usage of a higher number of `adjacent` predicates was prevented in order to speed up calculations and ensure that Progol could completely sample the space of possible rules.

The length of connecting loops is the second most frequently used predicate. Surprisingly, the length of the loop appears almost as frequently in signatures of the folds as it does in the superfamilies; seven and nine occurrences, respectively, in the power rules. The length of a loop affects the packing of the secondary structures it connects and this might explain their occurrences in fold rules. For superfamily and family signatures, the conservation of the length of connecting loops can also be explained by the requirements for binding sites and functional regions (see Rossmann and P-loop folds, above). The length of the loops used to construct signatures varies from one structural class to another, with the loop of the all-β class being the longest.

Of all four classes, the signatures of the all-α class make more use of `total_length` which indicates that length is conserved amongst these folds.

Lastly, to our surprise, the hydrophobicity as well as the presence of proline residues do not make significant contribution to the signatures. In the work of King *et al.*[32] the hydrophobicity was found to be a main determinant of the topology of sheets. However, this study suggests that hydrophobicity is not important in dictating how a sequence will adopt a particular fold. Further analyses are required to investigate this surprising result. In contrast, our study suggests that the length of the loops plays an important role in determining protein folds identity, perhaps this is because the loop affects the packing of the flanking elements.

Relationship between the signatures and SCOP hierarchy

The protocol used for the analysis does not force rules to be learnt at the fold level. The system has the ability to learn more than one rule to cover all the positive examples. The choice will depend on what is considered most favourable according to equation (1). Since Progol produced on average three rules per fold, it is possible that the signatures coincide with superfamilies and families in SCOP. Table 5 shows the number of signatures that correspond to each level. A distinction is made between (1) signatures that coincide with a superfamily because the fold has only one superfamily and (2) the genuine case where the fold has more than one superfamily but the signature occurs only in one. Of course, in the former case, the signatures may represent characteristics which are specific to the superfamily but there are no data to support it. The same distinction is made for families.

About half of the 59 rules represent fold signatures. This number goes down to about a third when only the power rules are considered. In both cases only a small number of family rules were produced. The four-helical cytokines provide examples of family rules. In our data-set this fold has two families, the short and the long chains, and Progol produced two rules each one covering exactly one family.

The signatures often indicate commonality between superfamilies without always obeying the rigid structure of SCOP. This is seen in the three-helical bundle fold where some of the rules involve strands and occur in all but the homeodomains, which contain only helices.

Fold signatures are likely to be the result of stereochemical constraints. For example, the length of a loop affects the packing of the flanking secondary structures. Proteins from different superfamilies of the same fold are presumed to be analogues that have converged toward a stable folding arrangement. However, as Russell *et al.*[42] suggest, certain folds demonstrate binding site similarity in the absence of homology. For those folds, the signatures might indicate functional constraints.

Discussion

This analysis is currently limited by the absence of structure superposition. It is difficult to align reliably and systematically all the structures of a given fold and identify the common secondary

Table 5. Relationship between the signatures and SCOP hierarchy

	All rules			Power rules		
Level	Multiple	Unique	**Total**	Multiple	Unique	**Total**
Fold	28	-	**28**	8	-	**8**
Superfamilies	9	15	**24**	2	9	**11**
Family	4	3	**7**	2	2	**4**

The Table presents the number of rules that coincide with a particular level of SCOP. Two subsets of the rules are presented: all rules and power rules, the latter refers to rules with 30% or more recall. We also distinguish between rules which coincide with a fold and characterise domains from more than one superfamily, indicated by multiple, from those who coincide with a fold but have one superfamily, indicated by unique, in that case further data is needed to decide if the rule represents a genuine feature of a fold or a superfamily. Similar distinctions apply to the other two levels, superfamilies and families.

structure elements.[43] This causes problems for folds with large insertions, although several mechanisms have been put in place to alleviate these problems. In some proteins the secondary structures matched by a rule are not structurally equivalent. Such large insertions mean that for the (TIM)-barrel and the immunoglobulin folds the algorithm was prevented from learning general rules. As a further measure, preliminary studies suggest that a simple solution such as numbering the secondary structures from C- as well as N-terminal region can be quite effective.

This study shows that signatures of protein folds and superfamilies can be automatically and systematically discovered. Once the background knowledge has been defined, this approach provides an unbiased way to test hypotheses. Once a rule has been found, it can be tested experimentally, for example, by varying the length of the loops in engineered proteins.

Our key conclusions are:

(1) Protein fold signatures exist. Indeed, half the rules produced represent fold signatures. The other half characterises aspects of protein function and the signatures were probably conserved for that reason.

(2) The length of loops makes a major contribution to the identity of protein folds. Surprisingly, the length of loops is used almost as frequently in signatures of folds as in signatures of superfamilies. However, the reasons for this might differ. In folds, the length of the loops may be critical to induce the correct packing of the flanking elements. While for superfamilies, signatures involving loops often correspond to functionally important regions.

(3) Hydrophobicity and the presence of proline seem to play a less important role than we previously anticipated.

(4) The length of the domains is a conserved feature of the all-α class.

The majority of the principles of protein structure have been discovered by extensive human analyses. New initiatives and improvement of the methods for structure determination will lead to an explosion of the number of structures. Bourne *et al.*[44] predicts that over the next five years the number of structures could grow by as much as a factor of 3.

Increasingly, protein structure is involved in the process of genome annotation. Sophisticated methods are developed to detect remote homology in the hope that knowledge can be transfered to the protein under study. However, high-throughput structure determination projects by nature will populate the databases with proteins for which very little is known about their biology. Therefore there is a need for tools that automatically extract principles and assist the construction of classification schemes.

An interesting step towards a structural characterisation of protein signatures was taken by Kasuya *et al.*[13] who studied the three-dimensional motifs corresponding to Prosite patterns. Their analysis also showed that structure can help to distinguish false positive matches. However, we have analysed the relationship between Prosite motifs and SCOP and found that Prosite motifs mainly coincide with families (12 %) or even sub-families (40 %) of SCOP and a mere 8 % coincide with folds. In the globins, only leghaemoglobin has a Prosite motif. In λ-repressor, there are three motifs but those are specific to a family or even a subset of a SCOP family. Similarly, in the Rossmann fold, there are two Prosite motifs: one that represents members of the tyrosine-dependent oxidoreductases family, and another one that represents formate/glycerate dehydrogenases and NAD-domains. Finally, in the P-loop fold, the `ATA_GTP_A` motif matches seven out of 14 domains distributed over three families. Thus, in general, the signatures discovered by ILP are not described in Prosite.

The rules discovered automatically in this work can be classified into four groups. Firstly, there are signatures which are easily available to the community *via* the biological databases of protein patterns. The Prosite motif `ATA_GTP_A` matches half of the domains in the P-loop fold and is probably the best example of this kind. However, the signatures found in the patterns databases contain primarily sequence rather than structural motifs. As a consequence, these databases focus mainly on families of proteins. Secondly, there are signatures which can be found in the literature. The signatures for the globin, λ-repressor, cytokines, Rossmann-fold and P-loop are examples of this kind. These signatures are known but often require extensive literature search to be found. Thirdly, there are signatures which are not found in the literature nor in the public databases but are known to experts. The power rules are an example of this kind, 15 of the 23 signatures from Table 2 were presented to the expert developing SCOP, A. Murzin, in the form of a questionnaire with multiple choices. Murzin identified correctly virtually all the rules, which illustrates that the rules are genuine signatures that are known to the world's expert in protein structure. Finally, some of the rules are difficult to analyse and it is possible that further literature search or detailed analysis would reveal their meaning. The work presented here used a restricted background knowledge and yet produced rules which are known to the world experts on protein structure, which shows the potential of this approach. The next step will be to incorporate additional information in the background knowledge to see how the rules can be improved.

This study shows that ILP can learn expert type rules from complex biological data. Other areas of bioinformatics may well be amenable to knowledge discovery using ILP.

Materials and Methods

Data set

The version 1.39 of SCOP database was used in our study.[4] To reduce the redundancy in the data-set, one representative domain per protein was selected using `scoplib.pl`.[45] The data were also curated manually. When Progol was unable to find a rule for a given example, visual inspection often revealed abnormalities in the data. The most frequent problem was an error in the definition of the domain boundaries. Secondary structure information for each domain was calculated from the experimental three-dimensional structures using PROMOTIF.[46]

Data representation

ILP systems represent their data as logic programs. Progol, the ILP system used throughout this work, utilises the formalism of Prolog.[47] The basic syntactic structure in Prolog is a relation, also called a predicate, an example of which would be `adjacent(A, B)`, which states that the objects designated by A and B are adjacent in the primary structure. Rules, also called clauses, have the form *Head*:- *Body*, and are interpreted as follows: ''if the conditions in the *Body* of the clause are true then *Head* is a logical consequence''. For example,

```
fold('Globin-like', X) :-
adjacent(X, A, B, 1, h, h),
has_pro(B).
```

is interpreted as: ''if the first helix, designated by A, is followed by another one, designated by B, and B contains a proline then domain X must have fold type Globin''.

Table 6. Background knowledge, i.e. the building blocks of the protein fold signatures

`total_length(Lo =< D =< Hi)` : the length of the polypeptide chain of the domain D, i.e. the total number of amino acids.

`number_helices(Lo =< D =< Hi)` : the number of α-helices in domain D.

`number_strands(Lo =< D =< Hi)` : the number of β-strands in domain D.

`adjacent(D, A, B, Pos, TypA, TypB)` : this predicate indicates that the secondary structures A and B are consecutive. Furthermore, their respective types are `TypA` and `TypB`. `Pos` is the serial number of the secondary structure element `A` - helices and strands are numbered separately. D identifies the domain. E.g. `adjacent(d1hdr__, A, B, 6, e, h)` means that the sixth strand, here labelled `A`, is followed by a helix, here labelled `B` in domain `d1hdr__`. The predicate is true if the sequential number of the structure `A` is in the range: min(`Pos - 3, 0.6 × Pos`) to max(`Pos + 3, 1.4 × Pos`).

`length_sec_structure(A, L)` : the secondary structure `A` has length `L` where `L` takes one of the following symbolic values: `very_lo` , `lo`, `hi` and `very_hi`.

`average_hydrophobicity(A, H)` : the average hydrophobicity of the secondary structure element `A` is `H`. Possible values for `H` are: `very_lo`, `lo`, `hi` and `very_hi`.

`hydrophobic_moment(A, H)` : the hydrophobic moment of the secondary structure element `A` is `H`[55]. Possible values for `H` are: `very_lo`, `lo`, `hi` and `very_hi`.

`has_pro(A)` : the predicate is true if the secondary structure `A` contains a proline.

`length_loop(A, B, L)` : *L* is the length of the loop connecting the secondary structures `A` and `B`. The predicate works in two different ways. If the length is not supplied, then `length_loop` returns the exact length of loop connecting between `A` and `B`. If the length is supplied, therefore asking the question ''is the length of this loop *L* residues long'', the predicate is true if the exact length of the loop is in the range 0.5 × *L* to 1.5 × *L*. The length is measured as the number of residues.

Background knowledge

In preparation for a machine learning task, the user must define the predicates that can be used to construct new rules and how they can be combined with one another. This effectively defines the space of possible rules.

The background knowledge for the protein fold application comprises nine predicates, see Table 6. They can be classified into three categories: global, relational and local. The global predicates include the domain length, the number of helices and the number of strands. Other machine learning techniques, such as neural network and decision trees, only use this type of representation.

Relational information comes from two different sources in this application. The first source is provided by the predicate `adjacent`, which describes the relationship between two consecutive secondary structure elements. The other source comes from the association of variables, as exemplified by the rule for the β-Grasp fold (see Table 2), whose signature involves two `adjacent` predicates sharing a secondary structure which effectively defines a triple, β_2-α_1-β_3, although no triple relationship was explicitly encoded in the background knowledge (see Table 6). The ability to manipulate relational information is a distinctive feature of ILP systems. To incorporate this information into a neural network for example, all the possible pairs for `adjacent` and all possible associations would need to be predefined. Not only would this be a tedious process but more importantly, for any real world application, the number of input variables therefore created would exceed the capacity of the machine learning algorithm.

Local information makes up the third category. The predicates describe properties of the secondary structures and connecting loops. We have included the average hydrophobicity, the hydrophobic moment, the length and the presence of a proline. For the loops, only the length is accounted for.

Several mechanisms have been incorporated to circumvent problems caused by variations of the attributes such as length but also caused by the insertions and deletions. Intervals of values are used by the global predicates and the values for the boundaries are learnt by the algorithm. For the predicates `adjacent` and `length_loop`, we have used predefined intervals. Another measure was to number the helices and strands separately to allow the learning algorithm to rely on the most conserved numbering scheme. Finally, for the predicates modelling local information, the actual values have been replaced by symbolic constants. The distribution of the length of the secondary structure, average hydrophobicity and hydrophobic moment were calculated for the four main classes. If the actual value was lower or equal to the mean minus (or plus) two standard deviations the value was replaced by `very_lo` (`very_hi`), if it was lower or equal to the mean minus (or plus) one standard deviation the value was assigned `lo` (`hi`).

The background knowledge also contains two types of constraints. Integrity constraints are used to ensure that every rule considered contains at least one of the following predicates: `length_sec_structure`, `average_hydrophobicity`, `hydrophobic_moment`, `length_loop` or `has_pro`. In a preliminary study,[48,49] we compared the rules obtained in the presence or absence of integrity constraints. We concluded that forcing the rules to contain local features adds complexity and richness to the rules as judged by our knowledge of protein structure. The second form of constraint prevents rules from having more than two `adjacent` predicates and allows Progol to search the entire hypothesis space.

Machine learning algorithm

The results were obtained with the ILP system Progol version 4.4.[50] A detailed presentation of the algorithm can be found by Muggleton *et al.*[51,52] and only the main steps will be outlined here. The learning procedure involves two steps. First, a positive example is randomly selected and the most specific clause is calculated; it consists in finding all related predicates from the background knowledge.[52] The second step consists in using the predicates found in the first step to construct a hypothesis which maximises compression (see equation (1)). The best hypothesis is selected and the examples it matches are removed. The procedure resumes with the first step and continues until no more examples remain.

Example of the execution of Progol

The learning algorithm performs two main steps: (i) the construction of the most specific clause and (ii) a general to specific search to find an optimal rule. To illustrate the execution of Progol we use an example from the globin fold. In the first step, Progol selects a positive example randomly, here represented by the domain identifier `d1scta_`, and derives all the relevant information for this example from the background knowledge:

```
  [Generalising fold('Globin',d1scta_).]
  [Most specific clause is]
  fold('Globin-like',A) :-
   adjacent(A,B,C,1,h,h),
   adjacent(A,C,D,2,h,h),
   adjacent(A,D,E,3,h,h),
   adjacent(A,E,F,4,h,h),
   adjacent(A,F,G,5,h,h),
   total_length('$sk0' =< A =< '$sk2'),
   number_helices('$sk0' =< A =< '$sk2'),
   number_strands('$sk0' =< A =< '$sk2'),
    length_loop(B,C,1), length_loop(C,D,3),
length_loop(D,E,2),
   length_loop(E,F,2), length_loop(F,G,1),
   length_sec_structure(B,hi), length_sec_-
structure(D,hi),
   length_sec_structure(F,lo), length_sec_-
structure(G,hi),
   average_hydrophobicity(F,hi),
    hydrophobic_moment(F,lo), hydrophobic_-
moment(G,lo),
   has_pro(C), has_pro(G).
```

This constitutes a reservoir of information that can be used to construct new rules. In the second step, Progol searches the space of all possible rules, starting with the most general one, which is "everything is a Globin":

```
  [C:-8,13,20,0 fold('Globin', X).]
```

The search uses a branch-and-bound-like algorithm guided by a measure of compression. This measure depends on the number of positive and negative examples covered as well as the length of the clause. The rule is specialised: "every domain such that the first helix is followed by another helix":

```
  [C:-6,13,17,0 fold('Globin', X) :- adja-
cent(X,A,B,1,h,h).]
```

which leads to a new value for the compression measure. The clause is further specialised: "every domain such that the first helix is followed by another helix and another helix".

```
  [C:-2,13,12,0 fold('Globin', X) :- adja-
cent(X,A,B,1,h,h),
   adjacent(X,B,C,2,h,h).]
  ...
```

At the end of the search, Progol has found the rule that maximises its measure of compression:

```
  f = 8,p = 13,n = 1,h = 0
  [Result of search is]
  fold('Globin', X) :-
   adjacent(X,A,B,1,h,h),
   adjacent(X,B,C,2,h,h),
   total_length(135 =< X =< 166).
```

Parameters selection

An empirical investigation into the effect of a variety of parameters was made. Three different percentages of *noise* were sampled: 0, 10 and 20 %. The parameter *noise* controls the percentage of false positive examples allowed. Three *inflate* rates were tested: 100, 200 and 400 %. This parameter gives more weight to the positive examples. Three sets of negative examples each having: ×1, ×2 and ×4 more negative examples than positives were tested. This was done for pragmatic reasons, machine learning algorithms were developed, in general, to work with an equal number of positive and negative examples. All combinations of parameters were tested on the most populated fold of each class: DNA-binding three-helical bundle, immunoglobulin-like β-sandwich, β/α (TIM)-barrel and ferredoxin-like. The combination: `noise = 20, inflate = 200` and ×2 gave the best result, in the sense that it minimises the sum of the number of rules and false negatives. This combination was used throughout the tests.

The maximum number of hypotheses (nodes) explored during the search was restricted to 1000 during the cross-validation tests, and 10,000 otherwise. This allowed us to keep the execution time under two days. The maximum number of nodes was reached for 34 % of the runs and an average of 578 nodes were explored per run. Finally, the parameter *c*, which controls the maximum number of predicates in the body of a rule, was set to nine; however, this limit was never reached.

Cross-validation

Performance analyses were carried out over cross-validation test sets. Two different tests were applied depending on how much data were available. If the total number of examples (positive + negative) was greater than 60, a tenfold cross-validation test was applied, otherwise it was leave-one-out. This made it possible to run the entire cross-validation under two days using six of the 12 processors of our SiliconGraphics Power Challenge computer. The final rules were learnt on the complete data sets.

Acknowledgements

This work is supported by a BBSRC/EPSRC Bioinformatics grant.

References

1. Burley, S. K., Almo, S. C., Bonanno, J. B., Capel, M., Chance, M. R., Gaasterland, T., Lin, D. W., Šali, A., Studier, F. W. & Swaminathan, S. (1999). Structural genomics: beyond the human genome project. *Nature Genet.* **23**, 151-157.
2. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. (2000). The Protein Data Bank. *Nucl. Acids Res.* **28**, 235-242.
3. Muggleton, S. & De Raedt, L. D. (1994). Inductive logic programming: theory and methods. *J. Logic Programming,* **19/20**, 629-679.
4. LoConte, L., Ailey, B., Hubbard, T. J. P., Brenner, S. E., Murzin, A. G. & Chothia, C. (2000). SCOP: a structural classification of proteins database. *Nucl. Acids Res.* **28**, 257-259.
5. 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 genomic sequences to CATH. *Nucl. Acids Res.* **28**, 277-282.
6. Holm, L. & Sander, C. (1998). Touring protein fold space with DALI/FSSP. *Nucl. Acids Res.* **26**, 316-319.
7. Chothia, C. & Finkelstein, A. V. (1990). The classification and origins of protein folding patterns. *Annu. Rev. Biochem.* **59**, 1007-1039.
8. Holm, L. & Sander, C. (1998). Dictionary of recurrent domains in protein structures. *Proteins: Struct. Funct. Genet.* **33**, 88-96.
9. Salem, G. M., Hutchinson, E. G., Orengo, C. A. & Thornton, J. M. (1999). Correlation of observed fold frequency with the occurrence of local structural motifs. *J. Mol. Biol.* **287**, 969-981.
10. Shakhnovich, E. (1997). Theoretical studies of protein folding thermodynamics and kinetics. *Curr. Opin. Struct. Biol.* **7**, 29-40.
11. Fersht, A. (1997). Nucleation mechanism in protein folding. *Curr. Opin. Struct. Biol.* **7**, 3-9.
12. Dobson, C. M. & Karplus, M. (1999). The fundamentals of protein folding: bringing together theory and experiment. *Curr. Opin. Struct. Biol.* **9**, 92-101.
13. Kasuya, A. & Thornton, J. M. (1999). Three-dimensional structure analysis of prosite patterns. *J. Mol. Biol.* **286**, 1673-1691.
14. Hofmann, K., Bucher, P., Falquet, L. & Bairoch, A. (1999). The Prosite database, its status in 1999. *Nucl. Acids Res.* **27**, 215-219.
15. Attwood, T. K., Croning, M. D. R., Flower, D. R., Lewis, A. P., Mabey, J. E., Scordis, P., Selley, J. N. & Wright, W. (2000). PRINTS-S: the database formerly known as PRINTS. *Nucl. Acids Res.* **28**, 225-227.
16. Smith, R. F. & Smith, T. F. (1990). Automatic generation of primary sequence patterns from sets of related protein sequences. *Proc. Natl Acad. Sci. USA,* **97**, 118-122.
17. Saqi, M. A. & Sternberg, M. J. E. (1994). Identification of sequence motifs from a set of proteins with related function. *Protein Eng.* **7**, 165-71.
18. Rigoutsos, I., Floratos, A., Ouzounis, C., Gao, Y. & Parida, L. (1999). Dictionary building *via* unsupervised hierarchical motif disacovery in the sequence space of natural proteins. *Proteins: Struct. Funct. Genet.* **37**, 264-277.
19. Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F. & Wootton, J. C. (1993). Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science,* **262**, 208-214.
20. Johassen, I. (1997). Efficient discovery of conserved patterns using a pattern graph. *CABIOS,* **13**, 509-522.
21. Artymiuk, P., Poirrette, A. R., Grindley, H. M., Rice, D. W. & Willett, P. (1994). A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. *J. Mol. Biol.* **243**, 327-344.
22. Russell, R. B. (1998). Detection of protein three-dimensional side-chain patterns - new examples of convergent evolution. *J. Mol. Biol.* **279**, 1211-1227.
23. Jonassen, I., Eidhammer, I. & Taylor, W. R. (1999). Discovery of local packing motifs in protein structures. *Proteins: Struct. Funct. Genet.* **34**, 206-219.
24. Mirny, L. A. & Shakhnovich, E. I. (1999). Universally conserved positions in protein folds: reading evolutionary signals about stability, folding kinetics and function. *J. Mol. Biol.* **291**, 177-196.
25. Bashford, D., Chothia, C. & Lesk, A. M. (1987). Determinants of a protein fold. Unique features of the globin amino acid sequences. *J. Mol. Biol.* **196**, 199-216.
26. Wierenga, R. K., Terpstra, P. & Hol, W. G. J. (1986). Prediction of the occurence of the ADP-binding β-α-β-fold in proteins, using and amino acid sequence fingerprint. *J. Mol. Biol.* **187**, 101-107.
27. Murzin, A. G. & Bateman, A. (1997). Distant homology recognition using structural classification of proteins. *Proteins: Struct. Funct. Genet.* **Supplement 1**, 105-112.
28. Baldi, P. & Brunak, S. (1998). *Bioinformatics: The Machine Learning Approach*, MIT Press.
29. Muggleton, S., King, R. & Sternberg, M. J. E. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Eng.* **5**, 647-657.
30. King, R. D., Muggleton, S., 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. *Proc. Natl Acad. Sci. USA,* **89**, 11322-11326.
31. Hirst, J. D., King, R. D. & Sternberg, M. J. E. (1994). Quantitative structure-activity relationships by neural networks and inductive logic programming. I. The inhibition of dihydrofolate reductase by pyrimidines. *J. Comput. Aided Mol. Des.* **8**, 405-420.

32. 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 Eng.* **7**, 1295-1303.
33. King, R. D., Muggleton, S. H., Srinivasan, A. & Sternberg, M. J. E. (1996). Structure-activity relationships derived by machine learning: the use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proc. Natl Acad. Sci. USA,* **93**, 438-442.
34. Harrison, S. C. & Aggarwal, A. K. (1990). DNA recognition by proteins with the helix-turn-helix motif. *Annu. Rev. Biochem.* **59**, 933-969.
35. Aggarwal, A. K., Rodgers, D. W., Drottar, M., Ptashne, M. & Harrison, S. C. (1988). Rocognition of a DNA operator by the repressor 434: a view at high resolution. *Science,* **242**, 899-907.
36. Mondragón, A. & Harrison, S. C. (1991). The phage 434 Cro/O_R1 complex at 2.5 Å resolution. *J. Mol. Biol.* **219**, 321-334.
37. Calladine, C. R. & Drew, H. R. (1986). Principles of sequence-dependent flexure of DNA. *J. Mol. Biol.* **192**, 907-918.
38. Otwinowski, Z., Schevitz, R. W., Zhang, R. G., Lawson, C. L., Joachimiak, A., Marmorstein, R. Q., Luisi, B. F. & Sigler, P. B. (1988). Crystal structure of trp repressor/operator complex at atomic resolution. *Nature,* **335**, 321-329.
39. Klemm, J. D., Rould, M. A., Aurora, R., Herr, W. & Pabo, C. O. (1994). Crystal structure of the Oct-1 POU domain bound to an octamer site: DNA recognition with tethered DNA-binding modules. *Cell,* **77**, 21-32.
40. Rozwarski, D. A., Gronenborn, A. M., Clore, G. M., Bazan, J. F., Bohm, A., Wlodawer, A., Hatada, M. & Karplus, P. A. (1994). Structural comparisons among the short-chain helical cytokines. *Structure,* **2**, 159-173.
41. Kruse, N., Shen, B.-J., Arnold, S., Tony, H.-P., Müller, T. & Sebald, W. (1993). Two distinct functional sites of human interleukin 4 are identified by variants impaired in either receptor binding or receptor activation. *EMBO J.* **12**, 5121-5129.
42. Russell, R. B., Sasieni, P. D. & Sternberg, M. J. E. (1998). Supersites within superfolds. Binding site similarity in the absence of homology. *J. Mol. Biol.* **282**, 903-918.
43. Gerstein, M. & Levitt, M. (1998). Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins. *Protein Sci.* **7**, 445-456.
44. Bourne, P. E. (1999). Editorial. *Bioinformatics,* **15**, 715-716.
45. Kelley, L. A., MacCallum, R. M. & Sternberg, M. J. E. (2000). Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol.* **299**, 510-522.
46. Hutchinson, E. G. & Thornton, J. M. (1996). PROMOTIF - a program to identify and analyze structural motifs in proteins. *Protein Sci.* **5**, 212-220.
47. Clocksin, W. & Mellish, C. (1981). *Programming in Prolog*, Springer-Verlag, Berlin.
48. Turcotte, M., Muggleton, S. & Sternberg, M. (1998). Protein fold recognition. In *Proc. of the 8th International Workshop on Inductive Logic Programming (ILP-98)* (Page, C., ed.), pp. 53-64, LNAI 1446, Springer-Verlag, Berlin.
49. Turcotte, M., Muggleton, S. & Sternberg, M. (2000). The effect of relational background knowledge on learning of protein three-dimensional fold signatures. *Machine Learn.* **43**, 81-95.
50. Muggleton, S. & Firth, J. (1999). CProgol4.4: theory and use. In *Inductive Logic Programing and Knowledge Discovery in Databases* (Džeroski, S. & Lavrac, N., eds), in the press.
51. Muggleton, S. (1992). *Inductive Logic Programming*, Academic Press, London.
52. Muggleton, S. (1995). Inverser Entailment and Progol. *New Gen. Comput. J.* **13**, 245-286.
53. Pearson, W. R. & Lipman, D. J. (1988). Improved tools for biological sequence analysis. *Proc. Natl Acad. Sci. USA,* **85**, 2444-2448.
54. Pesce, A., Couture, M., Dewilde, S., Guertin, M., Yamauchi, K., Ascenzi, P., Moens, L. & Bolognesi, M. (2000). A novel two-over-two α-helical sandwich fold is characteristic of the truncated hemoglobin family. *EMBO J.* **19**, 2424-2434.
55. Eisenberg, D., Weiss, R. M. & Terwilliger, T. C. (1984). The hydrophobic moment detects periodicity in protein hydrophobicity. *Proc. Natl Acad. Sci. USA,* **81**, 140-144.
56. Kraulis, P. J. (1991). MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures. *J. Appl. Crystallog.* **24**, 946-950.

Appendix

A Testing the rules on the latest release of SCOP

Rules derived automatically from the version 1.39 of SCOP, were tested on new entries only found in the latest release, 1.50. This analysis supports the same conclusions as the cross-validation test and provides further evidence that the rules represent genuine features of protein fold and function.

To establish a list of new entries, we compared all the sequences of SCOP 1.50 to those of version 1.39 using fasta[53] and a cutoff E value of 10^{-5} (number of matches expected by chance). The resulting test set was made non-redundant at 40 % identity. All entries with no match with the old SCOP were included in the study. To allow for a direct comparison with the results reported in the main text, the new test sets were constructed with the same ratio of positive and negative examples, i.e. 1/3 and 2/3, respectively. For the globins and interleukin-8 folds there were no new entries that could not be detected by sequence comparison method alone, therefore they are not included.

Table A1 summarises the results. The overall accuracy is 73(±2), compared to 74(±1) measured on cross-validation data-sets. Both accuracies are higher than that of a random prediction (with 99 % confidence level) and the difference between the two is not significant (with 99 % confidence level). Since the number of new entries is low, it is difficult to analyse the rules individually; Table A2 lists the rules for folds with more than three positive examples. The rules for DNA-binding three-helical bundle, immunoglobulin, P-loop, α/β-hydrolases and ferredoxin performed well on both statistics, recall and precision. However, the rules for OB-

Table A1. A test of time, applying the rules learnt from SCOP 1.39 to new entries only found in the latest release, 1.50

Fold	Examples	Families	Superfamilies	FN	% Accuracy
All-α:					
DNA 3-helical	11	11	3	4	85 ± 6
EF hand-like	5	4	2	2	60 ± 13
4-Helical cytokines	2	2	1	1	67 ± 19
Lambda repressor	1	1	1	0	100 ± 0
All-β:					
Ig beta-sandwich	19	9	7	9	81 ± 5
Tryp ser proteases	2	2	1	2	67 ± 19
OB-fold	12	5	3	10	68 ± 8
SH3-like barrel	4	4	3	2	57 ± 13
Lipocalins	2	2	1	1	83 ± 15
α/β:					
β/α (TIM)-barrel	21	18	14	11	72 ± 6
Rossmann-fold	3	2	6	22	76 ± 7
P-loop	16	10	1	9	70 ± 6
Periplasmic II	2	1	1	1	83 ± 15
α/β-Hydrolases	5	5	1	2	67 ± 12
α + β:					
Ferredoxin-like	16	13	10	7	77 ± 6
Zincin-like	2	2	1	1	50 ± 20
SH2-like	2	2	1	2	75 ± 15
Beta-Grasp	9	5	5	6	67 ± 9
					73 ± 2

For each fold, the Table lists the number of positive examples, the number of families and superfamilies, the number of false negative examples (FN), i.e. positive examples which are not represented by any of the rules, and the accuracy.

fold and Rossmann-fold have not, although the number of new examples is large, 12 and nine, respectively. Looking at the positive examples missed by the first rule of the OB-fold, we find that seven out of 12 have the correct ordering of the secondary structures specified by the rule, but failed because the length of the first strand is out of range. In three cases, the first strand is too long, it is classified as `very_long`. For example, the first strand of the N-terminal domain of aspartyl-

Table A2. Protein signatures learnt on SCOP 1.39 applied to new entries only found in SCOP 1.50

All-α:
DNA-binding 3-helical bundle: (recall = 36 %, precision = 100 %, *n* = 11). The length is 34 to 105 residues, there are exactly three helices, the loop between the 1st and 2nd helix is two to four residues long.
EF Hand-like: (recall = 20 %, precision = 100 %, *n* = 5). The 1st strand is followed by a helix, the 2nd strand is immediately followed by a helix.
All-β:
Immunoglobulin β-sandwich: (recall = 24 %, precision = 63 %, *n* = 21). There is at most one helix, the loop between the 5th and 6th strands is three to seven residues long.
Trypsin serine proteases: (recall = 0 %, precision = N/A, *n* = 2). The loop between the 12th and 13th strand is four to 12 residues long.
OB-fold: (recall = 0 %, precision = 0 %, *n* = 12). The 1st strand is long and followed by another strand. The 1st helix is followed by a strand.
OB-fold: (recall = 17 %, precisio*n* = 67 %, *n* = 12). There are five or six strands, the loop between the 1st and 2nd strand is two to four residues long, the 4th strand is followed by another strand.
SH3-like barrel: (recall = 25 %, precision = 50 %, *n* = 4). There are four to six strands, the loop between the 3rd and 4th strand is one to three residues long.
α/β class:
NAD(P)-binding Rossmann-fold domains: (recall = 11 %, precisio*n* = 33 %, *n* = 9). The 1st strand is followed by a helix, the two structures are separated by a short connection, about one residue. Also, the 6th strand is followed by a helix.
P-loop containing nucleotide triphosphate hydrolases: (recall = 44 %, precisio*n* = 54 %, *n* = 16). The loop which connects 1st strand and following helix is three to seven residues long.
α/β-Hydrolases: (recall = 60 %, precision = 75 %, *n* = 5). The loop connecting the 1st strand and the one that follows is one to three residues long and the 7th helix is followed by a strand.
α + β class:
Ferredoxin-like: (recall = 44 %, precision = 88 %, *n* = 16). There are 3 to 5 β-strands, the 1st strand is followed by a helix and the loop between the 2nd helix the strand that follows it is two to four residues long.
β-Grasp: (recall = 22 %, precision = 50 %, *n* = 9). The domain contains the following motif: β_2-α_1-β_3 and the loop between α_1 and β_3 is two to four residues long.

The statistics in parentheses are: recall, defined as true positive/total number of positive examples, precision, defined as true positive/total number of prediction, n is the total number of positive examples.

tRNA synthetase (PDB 1b8a) is 15 amino acid residues long and our definition for long requires 11 to 13 amino acid residues. This suggests that the categories should perhaps be defined hierarchically, that is `very_long` secondary structures should also be classified as `long` as well. For the remaining five examples, the strands are too short. Looking at the structures reveals a problem with the definition of the boundaries of the secondary structures. In aspartyl-tRNA synthetase the first strand bridges both sides of the open barrel, whilst in TIMP-2 (PDB 1bqq) the equivalent region contains two strands, which when considered equivalent to the same region in 1b8a, would amount to 14 amino acid residues. Such problems can only be solved by including superposition information in the background knowledge. The situation for the Rossmann fold is slightly different. The requirements for the first part of the rule, i.e. the first strand followed by a helix separated by a short coil, are satisfied by most domains. In all positive examples, the first strand is followed by a helix, and for five out of nine domains the length of the loop is in the correct range. However, only one domain satisfies the second part of the rule which requires the sixth strand to be followed by a helix. For two domains, a sheet is inserted and disrupts the numbering scheme. For the other two domains the topology of the sheet is different from the usual 321456: for L-alanine dehydrogenase (PDB 1pjc) the topology is 3214576 and for *S*-adenosylhomocystein hydrolase (PDB 1a7a) the topology is 32145876, here even structure superposition would not help.

The recent determination of two new globin structures reminds us that even well established rules are sometimes violated. Those are the first crystal structures representative of the truncated hemoglobin (trHb) family.[54] Unlike the classical fold, forming a three-over-three α-helical sandwich, the trHbs form a two-over-two α-helical sandwich. The two structures are very similar, 1.3 Å un-weighted r.m.s. over 115 residues and share 37% identity. The protist structure (PDB 1dlw, *Paramecium caudatum*) has no proline residue in the BC corner but the green algae structure (PDB 1dly, *Chlamydomonas eugametos*) does.

Edited by J. Thornton

(Received 6 April 2000; received in revised form 20 December 2000; accepted 22 December 2000)