doi:10.1016/S0022-2836(03)00620-X J. Mol. Biol. (2003) 330, 839–850

JMB

Available online at www.sciencedirect.com

SCIENCE DIRECT®

The Automatic Discovery of Structural Principles Describing Protein Fold Space

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

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

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

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

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

**Corresponding author*

The study of protein structure has been driven largely by the careful inspection of experimental data by human experts. However, the rapid determination 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 folds. 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. To date, several hundred different types of fold have 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 the 3D molecular arrangement of any protein, 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 10,000 different types[1,2]), this will 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 been developed to classify proteins in terms of their proximity

Present addresses: A. P. Cootes and M. J. E. Sternberg, Department of Biological Sciences, Imperial College of Science, Technology and Medicine, South Kensington, London SW7 2AZ, UK; S. H. Muggleton, Department of Computing, Imperial College of Science, Technology and Medicine, South Kensington, London SW7 2AZ, UK.

Abbreviations used: ILP, inductive logic programming.

E-mail address of the corresponding author: m.sternberg@ic.ac.uk

to a set of idealised protein structural units.[6] Some of these databases have been shown to be largely similar but, nonetheless, importantly 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 types (albeit "preliminary" ones for the $\alpha + \beta$ main fold class) is the SCOP database.[3] The SCOP database, used widely 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, many of them do not discriminate between that fold and other folds. 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),[8–10] a machine learning technique, to the problem of automatically generating descriptions for folds 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 been applied to many problems in molecular biology.[11–15] 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 $\alpha + \beta$). Background knowledge consisted of structural information for each example considered, derived from secondary structure and multiple structure alignment information (as described in Methods). For each fold category in the four major main fold classes in SCOP (all-α, all-β, α/β or $\alpha + \beta$), 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 the 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 residue were considered. One of the major advantages of ILP over other forms of machine learning is that relations

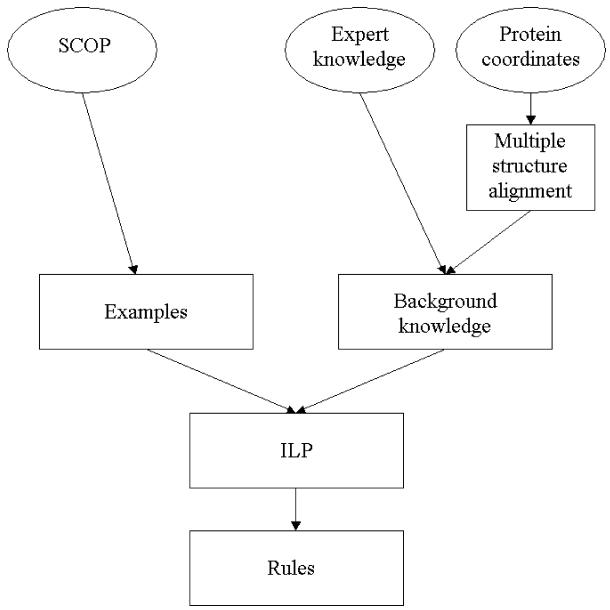

Figure 1. Information flow in ILP, which is driven by examples and background knowledge to produce new rules and principles. Examples of a given fold are taken from the SCOP database. Background knowledge is generated from structurally aligned protein coordinates and general structural principles defined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

between objects can be represented easily. 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. All of the attributes and relations considered are described in Methods.

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 example's background knowledge. Rules are constructed so as to maximise compression. The measure of compression used, f, is given by:

$$f = p - n - c$$

where p is the number of positive examples covered by the rule, n is the number of negative examples covered and c is the length of the rule. The parameter c ensures that for rules with equal coverage of positive and negative examples, the shorter one is favoured. 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 fashion similar to that described above. This process is iterated until there are no remaining positive examples.

The rules output by Progol are expressed also as logic programs and can be interpreted readily 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 are 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 cross-validation, 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 χ^2 test indicated that the results were statistically significant when compared to such a largest class prediction ($\chi^2 = 58.5$, $p \ll 0.01$).

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 folds 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 ten fold categories 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 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 generated manually and use a glossary of terms different from 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 compared also with those of a previous study that did not utilise multiple structure alignments.[16] 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 considered in the previous

Table 1. Cross-validated accuracy for fold rules

Fold category	+	−	Accuracy (%)	Expected (%)	Precision (%)	Recall (%)
Long α-hairpin	7	146	95	95	50	29
DNA/RNA-binding three-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
α/α Toroid	5	148	97	97	0	0
α-α 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 β-sandwich	16	129	90	89	53	62
Diphtheria 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
β-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
Seven-bladed β-propeller	6	139	97	96	100	33
Double-stranded β-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 β/α-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 α-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-Ado-L-Met-dependent methyltransferases	5	160	98	97	100	20
PLP-dependent transferases	5	160	100	97	100	100
α/β-Hydrolases	17	148	96	90	92	71
α/β Class	114	2196	97	95	78	54
Lysozyme-like	5	156	98	97	100	20
β-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

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 ($p \ll 0.01$) according to a χ^2 test.

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 referred to above,[16] the incorporation of structural superpositions into the background knowledge has enabled important global, as well as local, fold properties to be identified automatically. 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 2(b)), as

well as the topologies of both antiparallel sheets making up the immunoglobulin sandwich structure (Figure 2(c)). The TIM-barrel fold is identified as having an eight-stranded parallel β-sheet (Figure 2(a)). 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 the 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 glycine residues 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. This feature is not described in detail by SCOP but the text descriptions given in SCOP do not generally include information regarding sequence

Table 2. Comparison of ILP rules to SCOP descriptions for several folds

SCOP (version 1.50) fold category	Rule type	Rule
Immunoglobulin (1 002 001)	SCOP	Sandwich; seven strands in two sheets; greek-key; some members of the fold have additional strands
	ILP	Has antiparallel sheets B and C; B has three strands, topology 123; C has four strands, topology 2134
Prealbumin-like (1 002 003)	SCOP	Sandwich; seven strands in two sheets, greek-key; variations: some members have additional one or two strands to common fold
	ILP	Has a mixed sheet B. B has three strands with topology 213
TIM barrel (1 003 001)	SCOP	Contains parallel β-sheet barrel, closed; $n = 8$, $S = 8$; strand order 12345678; the first six superfamilies have similar phosphate-binding sites
	ILP	Has between five and nine helices; Has a parallel sheet of eight strands
Rossmann-like (1 003 002)	SCOP	Core: three layers, a/b/a; parallel β-sheet of six 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 three and four helices; Has α-helix B as the second core element in the sequence; B contains a glycine residue 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 first and fourth strands in the sheet B, respectively. C and D are the end strands of B and are 4.360(±2.18) Å apart. D contains a proline residuein the C-terminal end.
Barrel-sandwich hybrid (1 002 079)	SCOP	Sandwich of half-barrel-shaped β-sheets
	ILP	Has an antiparallel sheet B. B has four strands with topology 3214. C and D are the first and fourth strands in B, respectively. C and D are in contact. C contains a glycine residue in the N-terminal end
Long α-hairpin (1 001 002)	SCOP	Two helices; antiparallel hairpin, left-handed twist
	ILP	Has a total of two helices. α-Helices B and C are the first and second 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°) to one another. C contains a glycine residue in the middle region

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.

Table 3. Rules learnt with ILP in the absence of multiple structure alignments

SCOP fold category	Rule
Immunoglobulin	There is at most one helix, the loop between the fifth and sixth 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 first strand is followed by a helix, the two elements are separated by a coil of about one residue. The sixth strand is followed by a helix
SH3	There are four to six strands, the loop between the third and fourth strand is one to three residues long.

The rule corresponding to each fold in Table 2 is taken directly from Turcotte *et al.*[16] (where available). Several folds in Table 2 were not considered in the previous study and are absent from this Table.

properties. Sequence properties that are peculiar to a given fold may give valuable insight into the relationship between sequence, structure and function. While this sequence motif is known to be associated with the Rossmann fold, ILP offers the potential to automatically identify key residues.

Immunoglobulin fold

For some folds, ILP rules describe the folds in more detail than the SCOP database. SCOP describes immunoglobulin folds as having seven strands in two 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, 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 folds. The ILP rule given in Table 2 describes an immunoglobulin fold as having two antiparallel sheets, one with four strands and the other with three strands, in agreement with SCOP. However, it includes the topology of those sheets (2134 and 123, respectively). This clearly distinguishes this fold from the prealbumin fold, which contains a sheet with three core strands that is mixed and has the first strand in the centre of the sheet (topology 213). The immunoglobulin fold 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†. 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 articulated them clearly 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.

† http://www.sbg.bio.ic.ac.uk/~cootes/rules.html

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 Tables 4 and 5) 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 eight core strands (data not shown) apart from the TIM-barrels. Hence, a rule describing a TIM-barrel as an eight-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 glycine and proline residues 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 described, or implied, by the corresponding SCOP description.

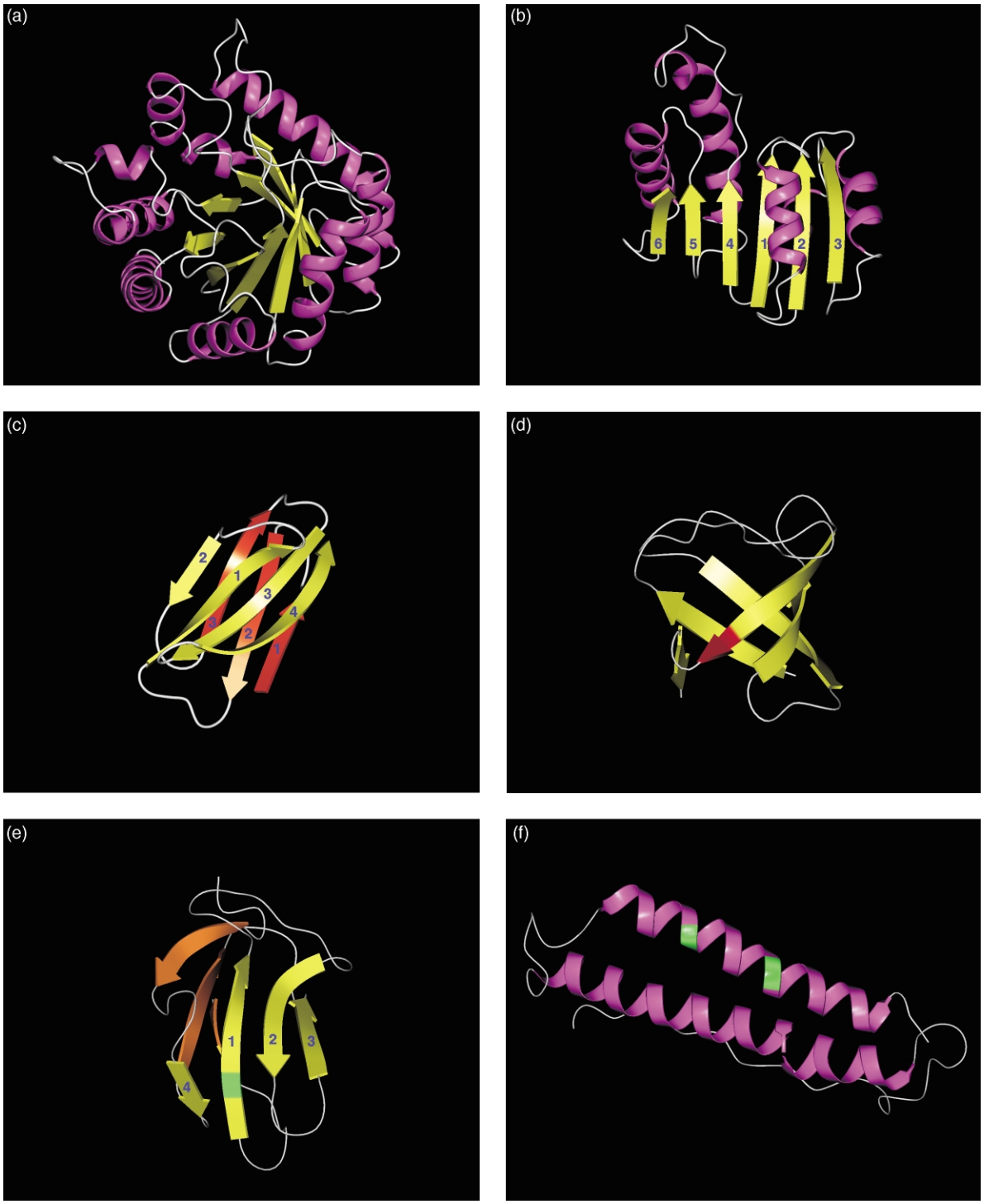

Figure 2. Structures demonstrating features learnt using ILP. The features highlighted correspond to the rules learnt using ILP in this study, given in Table 2. The numbering of strands shows sheet topology, relevant glycine residues are highlighted in green and proline residues in red. The structures 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 α-hairpin.

Interestingly, the Progol rule describes the presence of a proline residue at the end of the fourth strand (Figure 2(d)). This proline residue 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 residue, with its peculiar property of greatly restricting backbone conformational freedom, might be strongly preferred in order to form the break in the regular secondary structure of the

Table 4. Predicates describing protein fold properties

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 that 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 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

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

fourth strand. It might play a role in ensuring that the SH3 barrel is an open one, as a proline residue 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 residue may be required in order to give the 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 anti-parallel sheet with topology 3214, in which the N-terminal portion of the first strand contains a glycine residue. 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 well outside (by as much as 50° in several cases) of the expected range for a residue within a β-strand $(-60° < \text{phi} < -150°, 90° < \text{psi} < 180°)$. This indicates that a glycine residue would be very strongly 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 residue or indeed the topology of the sheet.

Table 5. Occurrence of predicates in rules

Predicate	No. 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

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 residue in the middle of the second of two helices is an important feature for this fold. Domains from four different superfamilies in this fold category exhibit this feature, implying that the glycine residue is important structurally. However, glycine residues have phi/psi angles within, or close to (within 15° of), the expected range for a residue in an α-helix $(-60° < \text{phi} < -150°, -45° < \text{psi} < -60°)$, indicating that backbone conformational freedom is not required in order to form this fold.

The rule learnt in this study describes the way in

which the α-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 by SCOP, as no attempt was made to learn this type of feature.

False positives

Despite considerable agreement between different structure classification schemes, differences do exist,[7] suggesting that misclassifications can occur. The rules that have been learnt automatically in this study may prove useful in finding where, if any, those misclassifications have occurred. False positives predicted by the rules learnt using ILP represent an automatically generated list of potential candidates for reclassification. This approach offers the advantage of having the reasons for the false positive being deemed to belong to the fold category of interest clearly articulated by the ILP rule.

For example, the two rules that were learnt for the Rossmann fold in this study match a small number of domains that are not classified as Rossmann folds within the SCOP hierarchy. The first of the two rules describing Rossmann folds requires that the structure contain between three and four core helices with two glycine residues in the helix at core position b. This rule matches false positives from five SCOP sequence families. In two of these families, these glycine residues correspond to a G-X-G-X-X-G sequence motif in the matched domains. These domains are d1b6ra2 (SCOP fold biotin carboxylase N-terminal domain-like) and d1uag_1 (SCOP fold N-terminal domain of MurD (UDP-*N*-acetylmuramoyl-L-alanine:D-glutamate ligase)). Apart from containing a Rossmann fold-like sequence motif, the parallel sheet topologies of these two domains (312 and 32145, respectively) are sub-topologies of the sheet topology described in the second Rossmann fold rule (321456). The text description accompanying the SCOP fold category of the first domain mentions that the fold is a "possible rudiment form of Rossmann-fold domain". The SCOP text description in the latter case makes no such reference in the version of SCOP used here, but in a later version (1.61) the fold is described as an "incomplete Rossmann fold". Interestingly, these domains are classified as belonging to the Rossmann fold category at the topology level of CATH v2.4.

The second rule describing Rossmann folds as those containing a parallel sheet with topology 321456 and helices at core positions g and i in contact and parallel also locates a false positive, the domain d1fsz_ in the SCOP fold category 'tubulin, GTPase domain'. Apart from matching this rule, the domain contains a G-X-G-X-X-G sequence motif, with two of the glycine residues in the helix at core position b, part of the first Rossmann fold rule discussed earlier. However, this domain does not match the first rule due to a different number of core helices. However, the presence of this motif may support the ILP rule asserting that this domain could be considered a Rossmann fold. This domain is classified as a Rossmann fold at the topology level in CATH v2.4.

The above examples indicate that the ILP rules sometimes locate domains that could be considered potential misclassifications. However, in other situations, false positives reveal limitations in the fold rules. Consider, for example, the ILP rule describing TIM barrels as containing eight core strands in a parallel sheet and between five and nine core helices. This rule matches two domains (d1c3pa_ and d1d0ba) from other fold categories in SCOP (arginase/deacetylase and leucine-rich repeat, LRR (right-handed beta-alpha superhelix), respectively). However, protein experts would not consider either of these structures to be TIM barrels, due to their sheets being relatively flat in space. The ILP rule for the TIM barrel sheet does not describe the curvature of the sheet as a consequence of achieving fold discrimination with as few, short rules as possible (as discussed earlier in the section TIM-barrel fold). However, this omission means that false positives found in this case are not the result of misclassification.

Outliers

In the process of constructing multiple structure alignments for each fold, domains were sometimes removed from consideration in order to ensure alignments were consistent and to avoid misalignments. These domains, which have been more difficult to align, may represent cases that are more difficult to classify (and the more interesting, in terms of searching for misclassifications). Overall, the results are still statistically significant if outliers are included in testing. Even if all outliers are included in testing as false negatives (that is, the outliers are all considered to be incorrect without any attempt to realign them and test them against the ILP rules) the overall accuracy is still 97% and the results remain statistically significant ($\chi^2 = 43.4$, $p \ll 0.01$).

If outliers that were removed from consideration originally are subsequently added to the multiple structure alignment without the original restrictions (as described in Methods), it is possible to then test these for consistency with ILP rules. In some cases, these outliers match the ILP rules for that fold. For the Rossmann fold, one SCOP sequence family did not have any example domains in the original multiple structure alignment. Upon subsequent alignment, several domains from this family matched the corresponding ILP rules. However, for other folds, this process tended to reveal reasons for which these outlier domains were originally removed. Many outlier structures failed to match their corresponding ILP rules upon subsequent alignment due to missing or poorly defined secondary structures or due to

misalignment with the domain in the multiple structure alignment. In some cases, the structures are only fragments of a given fold. However, some outliers did not match ILP rules despite realigning well. The ILP rule for the SH3-barrel fold describes a four-strand open barrel with a proline in the fourth strand. Outlier domains from one SH3-barrel sequence family had all four core strands when realigned but did not have a proline residue in the fourth strand.

Conclusion

The structural principles underpinning much of fold space can be described automatically using ILP. Such rules are objective and are discriminatory with respect to different fold categories by construction. Furthermore, the rules produced can be interpreted readily 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.

In this work, rules have been learnt automatically for 45 of the more common types of protein fold. In principle, the process outlined here could be applied to the remaining, relatively rare folds. However, as has been shown here, the accuracy of prediction decreases with the inclusion of rules learnt for folds that contain only a few independent examples. There are two main reasons for this. Firstly, it is more difficult to learn general principles from a small number of examples. Secondly, the definition of core elements is less reliable when using multiple structure alignments generated from only a few structures. In the most extreme case, a fold with only one known example, multiple structure alignments cannot be generated at all. However, future work may be able to overcome this by employing machine learning techniques to predict core secondary structure elements in the absence of structure alignments. That is, rules for core elements could be learnt from multiple structure alignments that are known to be reliable and then used to predict core elements in individual structures for which there is no reliable multiple structure alignment. It might be reasonable to expect that elements with particular physical properties are less likely to be conserved across a fold class. For example, secondary structure elements that are short or that make few physical contacts with the rest of the proteins structure might be less likely to be core. Such an approach may prove to be more reliable than using multiple structure alignments for rare folds. Having predicted core regions, one would then proceed using the strategy described here.

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 been shown to differ.[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, release number 1.50.[3] 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 database,[20] 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 four or more sequence families could be clustered in a multiple structure alignment (see the 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 pair-wise 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 pair-wise alignments in each fold category were then clustered with respect to their root mean square distance (RMSD), in a manner similar 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 pair-wise 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 pair-wise alignment with the master domain with RMSD >10 Å were firstly eliminated from further consideration. Also, in order to avoid corrupting the multiple alignment with misaligned pair-wise alignments, domains for which less than two-thirds 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 program.[24] Then a simple matching scheme was employed to match secondary structures units in the different domains on the basis of 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 was labelled a, the second group was labelled 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 the domains that were clustered to form the multiple structure alignment for that SCOP fold category. The negative examples were the 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 $\alpha+\beta$). 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

Fivefold 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 recalculated. One example per sequence family was selected randomly 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).

Subsequent alignment of outliers for testing

In order to evaluate rules learnt on those domains initially removed as outliers (as described above), outliers were aligned to the closest domain in the original multiple structure alignment calculated for that fold, irrespective of the conditions applied earlier. Core elements were then defined in the outlier domain by finding pair-wise matches between elements in the outlier and core elements in the closest domain already aligned. This was determined by the relative overlap of the elements in the pair-wise alignment in a similar fashion to the matching of elements in the definition of core elements performed earlier. Background knowledge was then determined for the outlier in terms of the defined core elements as before. Rules learnt for the corresponding fold were then tested for their consistency with the outlier domains background knowledge.

Acknowledgements

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

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. *Nucl. Acids Res.* **28**, 257–259.
4. Pearl, F. M. G., Lee, D., Bray, J. E., Sillitoe, I., Todd, A. E., Harrison, A. P. *et al.* (2000). Assigning genome sequences to CATH. *Nucl. Acids Res.* **28**, 277–282.
5. Holm, L. & Sander, C. (1998). Touring protein fold space with DALI/FSSP. *Nucl. Acids Res.* **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. *Struct. Fold. Des.* **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. *J. Logic Prog.* **19/20**, 629–679.
10. Muggleton, S. H. (1995). Inverse entailment and progol. *New Gen. Comput. J.* **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. *Proc. Natl Acad. Sci. USA*, **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. *J. Comput. Aided Mol. Des.* **8**, 405–420.
13. Muggleton, S. H., King, R. D. & Sternberg, M. J. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Eng.*, 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 Eng.* **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. *Proc. Natl Acad. Sci. USA*, **89**, 11322–11326.
16. Turcotte, M., Muggleton, S. H. & Sternberg, M. J. (2001). Automated discovery of structural signatures of protein fold and function. *J. Mol. Biol.* **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. *J. Mol. Biol.* **187**, 101–107.
18. Bork, P., Holm, L. & Sander, C. (1994). The immunoglobulin fold. *J. Mol. Biol.* **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 Sci.*, 8.
20. Brenner, S. E., Koehl, P. & Levitt, M. (2000). The ASTRAL compendium for protein structure and sequence analysis. *Nucl. Acids Res.* **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. *Electron. Trans. Artif. Intell.* **5**, 245–274.
22. Taylor, W. R. & Orengo, C. A. (1989). Protein structure alignment. *J. Mol. Biol.* **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. *J. Mol. Biol.* **299**, 499–520.
24. Hutchinson, E. G. & Thornton, J. M. (1996). PROMOTIF—a program to identify and analyze structural motifs in proteins. *Protein Sci.* **5**, 212–220.

Edited by J. Thornton

(Received 15 January 2003; received in revised form 6 May 2003; accepted 9 May 2003)