Machine Learning, 43, 81–95, 2001

The Effect of Relational Background Knowledge on Learning of Protein Three-Dimensional Fold Signatures

MARCEL TURCOTTE m.turcotte@icrf.icnet.uk
Biomolecular Modelling Laboratory, Imperial Cancer Research Fund, P.O. Box 123, London WC2A 3PX, UK

STEPHEN H. MUGGLETON stephen@cs.york.ac.uk
Department of Computer Science, University of York, Heslington, York, YO1 5DD, UK

MICHAEL J. E. STERNBERG m.sternberg@icrf.icnet.uk
Biomolecular Modelling Laboratory, Imperial Cancer Research Fund, P.O. Box 123, London WC2A 3PX, UK

Editors: Luc De Raedt, C. David Page and Stefan Wrobel

Abstract. As a form of Machine Learning the study of Inductive Logic Programming (ILP) is motivated by a central belief: relational description languages are better (in terms of accuracy and understandability) than propositional ones for certain real-world applications. This claim is investigated here for a particular application in structural molecular biology, that of constructing readable descriptions of the major protein folds. To the authors' knowledge Machine Learning has not previously been applied systematically to this task. In this application, the domain expert (third author) identified a natural divide between essentially propositional features and more structurally-oriented relational ones. The following null hypotheses are tested: 1) for a given ILP system (Progol) provision of relational background knowledge does not increase predictive accuracy, 2) a good propositional learning system (C5.0) without relational background knowledge will outperform Progol with relational background knowledge, 3) relational background knowledge does not produce improved explanatory insight. Null hypotheses 1) and 2) are both refuted on cross-validation results carried out over 20 of the most populated protein folds. Hypothesis 3 is refuted by demonstration of various insightful rules discovered only in the relationally-oriented learned rules.

Keywords: inductive logic programming, scientific discovery, protein fold

1. Introduction

Inductive Logic Programming (ILP) has been applied successfully in a large number of applications to structural biology (Muggleton, King & Sternberg, 1992; King et al., 1992; King et al., 1996; Srinivasan et al., 1996; Finn et al., 1998; Srinivasan et al., 1997; Sternberg et al., 1994). Underlying these investigations has been an attempt to test whether relational description languages are better (in terms of accuracy and understandability) than propositional ones for such applications. In general the advantages of relational representations seem to be born out in these investigations. However, it is always possible to choose propositional attributes which defeat such a conclusion. This can be demonstrated as

follows. Suppose investigator A applies ILP system B to problem C and shows that the rule $P(x, y) \leftarrow Q(x, z), R(z, y)$ has high accuracy and then submits a paper to journal D. Referee R might now respond that the solution could have been expressed in propositional form as $P \leftarrow S$ where S was defined (behind the scenes) as $Q(x, z), R(z, y)$. Such a response seems unreasonable since it was only possible for R to make such a suggestion after seeing the solution produced by system B. Alternatively R could suggest that a large set of propositions could have been introduced, which could then be combined to produce something equivalent to S. However, for relational representations (like the rule produced by B) in which new "existential" variables (like z) are introduced into the body it is far from clear how such propositional variables would be defined. Any such attempts at "propositionalisation" seem rather contrived and tend to detract from the readability of the resulting rules.

In this paper we take a different approach. The domain expert (third author) defined what appeared to him to be a "natural" representation for the application. One of these representations was more relationally-oriented than another. It was not possible in any obvious way to "propositionalise" the relationally-oriented representation. We used these representations to investigate the advantages and disadvantages of relationally-oriented representations.

This paper is structured as follows. In Section 2 the main null hypotheses to be empirically evaluated are laid out. The Molecular Biology motivation for the general application area is given in Section 3. In Section 4 background is given for both ILP (Section 4.1) and protein structure classification (Sections 4.2 and 4.3). The experiments are described in Section 5. Section 6 concludes and discusses the results.

2. Hypotheses to be tested

The null hypotheses to be empirically investigated in this paper are as follows.

1. For a given ILP system (Progol) provision of relational background knowledge does not increase predictive accuracy.
2. A good propositional learning system (C5.0) without relational background knowledge will outperform Progol with relational background knowledge.
3. Relational background knowledge does not produce improved explanatory insight.

3. Molecular biology motivation

The functional properties of proteins are determined by their three-dimensional structure. Therefore, to understand the function of proteins we need to unravel the principles that govern protein structure. Despite more than three decades of research, we cannot deduce the three-dimensional structure from the knowledge of its constituents (sequence) alone. However, vast amounts of data on protein structure have been accumulated, approximately 10,000 protein structures, and new projects, such as the Protein Structure Initiative, might

produce as much as 10,000 new structures over the next five years (Bourne, 1999). Furthermore, classification schemes, such as SCOP (Brenner et al., 1996), have been developed and can be used as a starting point for machine learning experiments. Here, we present an application of Inductive Logic Programming (ILP) to learn rules relating local structures to the concept of folds defined by SCOP. The objective is to automate the discovery of structural features, or signatures, of a fold that distinguish it from the rest. The three-dimensional structures of proteins are highly complex and the identification of rules explaining the observed fold remains a challenging area often involving the manual intervention of experts (Brenner et al., 1996; Branden & Tooze, 1999; Orengo Jones & Thornton, 1994). For several folds, these signatures are reported in the literature generally after extensive study. A few experts are familiar with many of these signatures, but the knowledge is not formalised with a common language, in a form suitable for automated testing as new structures are determined. Furthermore, automated methods could identify features missed by manual examination.

4. Background

4.1. Inductive logic programming

ILP is a logic-based approach to machine learning. Several features suggest it might be particularly well suited to study problems encountered in molecular biology. First, protein structures are the result of complex interactions between sub-structures and the ability of ILP algorithms to learn relations might prove to be a key feature. Second, 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. Third, logic programs are used as a common representation for examples, background knowledge and hypotheses, which provide a good integration for the development of applications together with the machine learning experiments. Finally, hypotheses can be made readable, by straightforward translation to natural languages, and integrated to the cycles of scientific debate.

Inductive Logic Programming is concerned with the induction of hypotheses from examples and background knowledge (Muggleton & Raedt, 1994). A restricted subset of first-order logic is used as a common representation for the examples, the background knowledge and also the generated hypotheses. In the case of the protein folds problem, a positive example might be that domain `d1hlb__` belongs to the Globin fold, represented as `fold('Globin-like', d1hlb__)`. The background knowledge might contain information such as the relationship between secondary structure and the presence of proline residues. The ILP algorithm then constructs a hypothesis which explains this example in terms of the background knowledge. The following rule was generated for the Globin-like fold.

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

According to this rule a domain X belongs to the Globin-like fold if its first helix is followed by another one that contains a proline. The results presented here were obtained with the ILP system Progol (Muggleton & Firth, 1999).

Progol is an ILP system which takes background knowledge, integrity constraints, and examples in the form of a logic program. It is also given the description of the hypothesis language in the form of "mode" declarations and "prune" statements. Progol then progresses using a covering algorithm by forming general rules from individual examples. For each example a "most specific" (or bottom) clause is constructed. A graph search is then carried out over the generalisations of the bottom clause. Individual clausal hypotheses in this search are evaluated by the "information compression" produced. Progol can be viewed as a modified Bayesian Maximum A Posteriori (MAP) learning algorithm.

4.2. Protein structure

Protein structures can be described at various levels of abstraction. In general, three levels are distinguished: primary, secondary and tertiary structure. Proteins are polymers, which means that they are made of smaller molecules, amino acids, assembled linearly. This level of abstraction is referred to as the primary structure or sequence. There are twenty naturally occurring amino acids and each one has a diverse set of chemical properties, for example hydrophobicity and size. Amino acids are represented with a standard one letter code and protein sequences are often written as strings of letters. The typical length of proteins varies from 100 to 500 amino acids.

A particular sequence of amino acids folds into a specific compact three-dimensional or tertiary structure from which the exact location of every atom can be deduced. The two predominant methods to structure determination are X-ray crystallography and NMR spectroscopy. Those techniques require sophisticated equipments—nowadays synchrotron facilities are often used as a source of radiation (Kim, 1998). Because of technological limitations, the sequences of amino acids are routinely determined in large quantities while the determination of the three-dimensional structure remains difficult. It is estimated that as part of the structural genomics projects there will be 10 large-scale initiatives and each of them will produce 200 structures per year (Bourne, 1999).

Early on it was predicted that segments of the primary sequence would adopt local regular structures (Pauling, Corey & Branson, 1951). The two main types being the α-helices and the β-strands, while the intervening regions are called loops or coils. Several computer programs exist to identify secondary structure elements from the three-dimensional structure.

The "Holy Grail" of molecular biology is to devise a method that would predict the three-dimensional structure, the exact location of every atoms, from the knowledge of the sequence alone. The problem is often broken down into two sub-problems; i) prediction of the secondary structure and ii) docking of the secondary structure elements to form the compact three-dimensional structure; for example, β-strands assemble together to form β-sheets. The problem of secondary structure prediction is to map each residue to one of the three types (H-helix, E-strand and C-coil).

4.3. Protein structure classification

There are several stages in the process of scientific discovery. One of the earliest is often the development of classification schemes (Langley, 1998). Once in place, qualitative and quantitative rules can be derived that relate the examples to each other. With approximately 10,000 known protein structures, over the last few years, classification schemes have been developed. In our study we have been using SCOP which is a classification done manually by a world-expert on protein structure (Brenner et al., 1996). These schemes facilitate our understanding of protein structure and here serves as a starting point for machine learning experiments, figure 1 illustrates the diversity of protein structures.

The basic unit of this classification is a domain, a structure or substructure that is considered to be folded independently, see figure 1. Small proteins have a single domain. For larger ones, a domain is a substructure.

Domains are grouped into families. Domains of the same family have evolved from a common ancestry. In most cases, the relationship can be identified by direct sequence comparison methods. The next level is called a super-family. The members of a super-family are believed to have evolved from a common ancestry, but often the relationship cannot be inferred by sequence comparison methods alone; the expert relies on other evidence, such as functional features.

The next level is a fold, proteins that share the same core secondary structures, and the same interconnections. The similarity is generally considered due to convergence towards a stable architecture. Finally, folds are conveniently grouped into classes based on the overall distribution of their secondary structure elements, see figure 1, the cytokines and globins are members of the all-α class, while the Rossmann fold belongs to the α/β class.

5. Experiments

The experiments in this section are aimed at evaluating the hypotheses in Section 2.

5.1. Materials

In order to allow reproducibility of the results, the algorithms and data have all been made available. The algorithms used in these experiments were Progol4.4[1] and C5.0.[2] The data sets, including algorithm settings, for the experiments have also been made available.[3]

We report on three experiments. In the first one, the background knowledge was limited and learning was essentially attribute-valued based. In the second one, the background knowledge was augmented with relational information. Finally, integrity constraints were used to express preferences formulated by the protein structure expert for certain forms of rules.

Our study is restricted to the five most populated folds of each of the four main classes, see Table 1. The justifications of this choice are as follows.

- Since these are the most populated folds they contain a relatively large number of examples, which means that learning is more robust and the results more meaningful. Many of the folds have only one known protein in them.

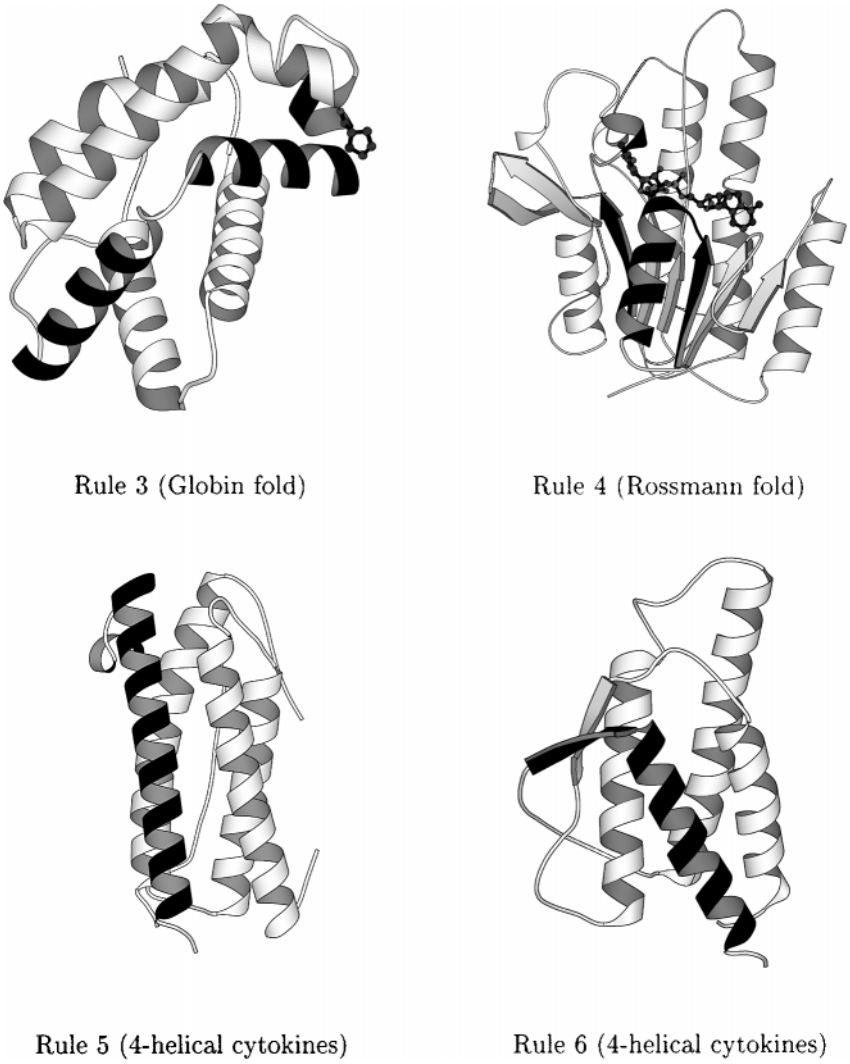

Figure 1. Schematic representation showing β-strands (arrows), α-helices (ribbons) and intervening loop regions. The figure shows the diversity of protein structures, for example the top two domains belong to two different folds while the bottom two domains belong to the same one. The secondary structure elements which are used in the description of a rule have been coloured in black, see text for details.

- Less populated folds are often ill-defined and contain multiple domains which coagulated together.
- The highly populated folds have been well studied and characterised in the literature, which means that the rules learned can be compared against what is already known.

Table 1. Selected folds. Dom is the number of domains, Fam the number of families and Super the number of super-families. The number of domains represents the number of entries after selection (`scoplib.pl`).

Fold	Dom	Fam	Super
All-α:			
DNA-binding 3-helical bundle	30	17	4
EF Hand-like	14	7	2
Globin-like	13	2	1
4-helical cytokines	10	3	1
lambda repressor-like DNA-binding domains	10	3	1
Other folds (92)	210	139	111
All-β:			
Immunoglobulin-like beta-sandwich	45	12	8
Trypsin-like serine proteases	21	4	1
OB-fold	20	11	4
SH3-like barrel	16	7	6
Lipocalins	14	2	1
Other folds (56)	220	123	90
α/β:			
beta/alpha (TIM)-barrel	55	28	17
NAD(P)-binding Rossmann-fold domains	21	7	1
P-loop	14	4	1
alpha/beta-Hydrolases	12	10	1
Periplasmic binding protein-like II	13	2	1
Other folds (70)	200	131	88
$\alpha+\beta$:			
Ferredoxin-like	26	21	17
Zincin-like	13	8	2
SH2-like	13	1	1
beta-Grasp	12	6	6
Interleukin 8-like chemokines	9	1	1
Other folds (96)	240	158	113

The four main classes of SCOP contain a total of 334 folds–representing 1251 domains, while the 20 folds we study in this paper contain 381 domains representing 30% of the total number of domains.

5.2. Method

Rules were generated for each fold as separate runs. In the case of the Globin-like fold the positive examples are the 13 domains classified as such in SCOP. Negative examples were

selected from all other folds of the same structural class, in the case of the Globin-like 26 negative examples were randomly selected from the 274 domains from the 96 other folds of the all-α class. The ratio of positive to negative examples was chosen to achieve rules which would have good coverage without having too many general clauses per fold. This was achieved by trying to maximise the number of rules plus the remaining number of uncovered examples. The best ratio of positives to negatives was found to be 1 : 2.

To reduce the redundancy in the data-set, one representative domain per protein was selected using `scoplib.pl` (Kelley et al. 2000). Prior to the cross-validation experiments the data was curated manually, when Progol was unable to find a rule for a given example. Visual inspection often revealed abnormalities in the data. The most common problem was the fusion of duplicated domains.

Secondary structure information for each domain was calculated from the three-dimensional structure using PROMOTIF (Hutchinson & Thornton, 1996).

Predictive accuracy was assessed by use of cross-validation.

5.3. Results

The cross-validation results are tabulated in Table 2. Weighted average accuracies, accompanied by standard errors based on summed contingency tables, are given in the last row of the table. The averaged accuracy differences between Progol II and all other systems are significant. No other differences are significant. We can thus refute null hypotheses 1 and 2 (see Section 2). Null hypothesis 3 requires a more in depth analysis of the contents of the rules in the experiments. This is provided below.

5.4. Attribute-values learning

For the first experiment, the background knowledge contains only predicates which encode global characteristics of protein folds, specifically, the total number of residues and the total number of secondary structures of both types, α and β.

This experiment shows that it is possible to construct good classifiers with a background knowledge which is essentially limited to attribute-values, see Table 2. The C5.0 algorithm, successor of C4.5 (Quinlan, 1993), gives greater accuracy than Progol I, though the difference between the two systems is not significant (Wilcoxon's test). Indeed, the two systems often produce similar rules, for example, Progol's rule for the Globin-like fold is:

Rule 1 (Globin-like) *X is a Globin-like if the length of the domain is between 135 and 166 residues long.*

```
fold('Globin-like', X) :-
    len_interval(135 =< A =< 166).
```

while C5.0 gives an interval of 135 to 163. This is perhaps not so surprising with such a restricted background knowledge.

Table 2. Cross-validation predictive accuracy. Columns labelled I, II and III refer to results obtained with Progol for three experiments.

Fold	C5.0	I	II	III
All-α:				
Globin-like	96.8	94.75	95.06	94.56
DNA-binding 3-helical bundle	84.6	65.36	82.97	81.92
4-helical cytokines	85.7	83.28	70.69	73.13
lambda repressor-like DNA-binding domains	73.7	49.95	73.43	63.37
EF Hand-like	78.5	66.64	77.57	68.48
All-β:				
Immunoglobulin-like beta-sandwich	77.4	81.41	76.29	71.07
SH3-like barrel	90.7	91.37	91.40	76.53
OB-fold	79.3	62.93	78.43	76.92
Trypsin-like serine proteases	94.7	93.56	93.13	81.47
Lipocalins	87.9	75.90	88.30	78.50
α/β:				
beta/alpha (TIM)-barrel	73.4	67.09	70.66	66.14
NAD(P)-binding Rossmann-fold domains	55.9	57.07	71.63	78.47
P-loop	56.7	67.29	76.02	81.21
alpha/beta-Hydrolases	52.4	66.89	72.18	75.08
Periplasmic binding protein-like II	58.0	66.42	68.91	62.94
$\alpha+\beta$:				
Interleukin 8-like chemokines	86.0	92.36	92.93	85.63
beta-Grasp	59.3	75.18	71.66	63.56
Ferredoxin-like	69.8	63.56	83.07	80.38
Zincin-like	69.2	67.01	64.30	56.30
SH2-like	66.7	69.45	76.81	79.38
Average:				
	74.8	72.48	78.28	74.35
	$\pm$1.30	$\pm$1.34	$\pm$1.23	$\pm$1.31

5.5. Relational learning

New predicates are added to the background knowledge which introduce relationships between secondary structure elements and their properties, see Appendix A.1 for the complete list of predicates.

The overall accuracy for this experiment, Progol II, is 78.8 %, which is significantly higher than the mean accuracy for Progol I experiment. More importantly, some of rules can now be related to published results in the relevant scientific literature. Consider the rule generated for the lambda repressor:

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 6 residues long.*

```
fold('lambda repressor', X) :-
    len_interval(53 =< X =< 88),
    adjacent(X, A, B, 3, h, h),
    coil(A, B, 6).
```

The particular coil region mentioned in the rule turns out to be important for the specific recognition of DNA (Branden & Tooze, 1999), see Section 6. When inspecting the rules, our protein structure expert (Sternberg) showed more interest in rules containing information about secondary structure elements. Although Progol had access to a richer background knowledge, including information about secondary structure, often Progol produced the same rule as previously, Progol I experiment, in particular this is seen for the Globin-like fold.

In the Progol III experiment, integrity constraints are introduced in the background knowledge to express the preference of the protein structure expert towards rules containing information about specific secondary structure elements. In effect this means that Progol now returns sub-optimal solutions. Indeed, the accuracy is reduced to 74.8%, but a larger fraction of the rules can now be interpreted in terms of previously published results in the relevant scientific literature.

5.6. Expert-type knowledge

In this section, we review four rules and present a possible biological interpretation. The complete set of rules is available from our Web site (`www.bmm.icnet.uk/ilp`).

Rule 3 (Globin fold) *Helix A at position 1 is followed by helix B. B contains a proline residue.*

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

The Globin-fold is a good example of divergent evolution. In SCOP, this fold comprises diverse sequences such as myoglobin, hemoglobin and phycocyanins. Yet the three-dimensional structure of these proteins is well preserved. One hallmark of this fold is the presence of a conserved proline residue in helix B, which causes a sharp bend in the main chain. This observation has been reported previously by Bashford, Chothia and Lesk (1987), see figure 1 where helices `A` and `B` are coloured black while the proline is represented as ball-and-stick.

Rule 4 (Rossmann fold) *Strand A at position 1 is followed by helix B. Strand C at position 6 is followed by helix D. The coil 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),
    coil(A, B, 1).
```

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. The region is embedded in a $\beta - \alpha - \beta$ motif which is highly conserved and contains the sequence motif G-X-G-X-X-G (Weirenga1 et al., 1986). The fifth and sixth secondary structures clamp the nicotinamide moiety of NAD, see figure 1, elements `A, B, C` and `D` are coloured black, while NAD is shown as ball-and-stick.

Similarly the rules generated for the lambda repressor and P-loop can be related to regions which are important for recognition and activity and have been documented in the literature.

Rule 5 (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),
    unit_len(A, hi).
```

Rule 6 (4-helical cytokines) *The second strand is immediately followed by a helix.*

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

Often, Progol produces more than one rule to cover all the positive examples of a fold. Similarly, SCOP classification has often more than one family and/or more than one superfamily per fold. Thus, sometimes the mapping of the rules onto the examples matches that of SCOP. This occurs for the 4-helical cytokines, which has two families, the long-chain and short-chain cytokines. Members of the long-chain cytokines family all start with a long helix, as observed by Progol, see Rule 5. While the distinctive feature of the short-chain cytokines is the absence of a coil between the last strand-helix pair, Rule 6. Although these proteins have been classified in the same family, their sequences are quite diverged (with pairwise distances within the so-called twilight zone) (Rozwarski et al., 1994). The fact that the second strand and last helix form a contiguous segment was observed by (Rozwarski et al., 1994) and used to tether their structural alignment. Further investigation reveals that the first residue of the helix also participates in the hydrogen bonds network of the sheet; except for one domain where the sheet is distorted.

The analysis above is sufficient to convince us that null hypothesis 3 (Section 2) can also be rejected.

6. Conclusion and discussion

We have presented three learning experiments using two background knowledge sets, attribute-valued and relational. All of the null hypotheses of Section 2 were rejected on the basis of the results. Overall we conclude that relational background knowledge has demonstrable advantages for learning in the construction of fold descriptions. The rules constructed in the experiments described in this paper represent the first systematic characterisation of the major protein folds.

In 1972, Irwin D. Kuntz wrote: "Although more than ten protein crystal structures have been determined, the principles by which these molecules develop secondary and tertiary structure are not yet well understood." (Kuntz, 1972) Twenty eight years later, approximately 10,000 protein structures are known yet our understanding of the principles governing protein folds is still not sufficient to provide accurate predictions.

In such a complex field as protein structure, it is unlikely that understanding will come from machine learning experiments alone. Rather the machine learning tools must be strongly integrated into the human process of scientific discovery. Inductive Logic Programming offers many distinct advantages with this respect. First, protein structures are the result of complex interactions between secondary structure elements and the ability of ILP algorithms to learn relations is a key feature. Second, ILP systems can make use of problem-specific background knowledge, allowing the expert to guide the search through the hypothesis space. Third, logic programs are used as a common representation for examples, background knowledge and hypotheses, which provides a good integration for the development of applications together with the machine learning experiments. Finally, hypotheses can be made readable, by straightforward translation to natural languages, and integrated to the cycles of scientific debates.

The two ways to describe protein folds have different biological implications. In the first paradigm, attribute values correspond to global properties, such as the number of residues of a domain or the number of secondary structure of a given. The rules produced in the context of the relational learning experiments, were found to be more informative, as judged by the protein structure expert. The rules can be explained in terms of structural and/or functional concepts, such as active site location. Progol, when constructing a rule, looks for motifs which are common to all the domains of a given fold but almost never encountered in others, except for a limited number of cases which is set by a user defined threshold (noise). Features which are important for structure and/or function tend to be conserved amongst members of the same fold, at least up to the super-family level. Hence the rules constructed by Progol can sometimes identify conserved functional motifs. Of the 59 rules generated for the experiment III, at least 5 can be related to previously published results. Unfortunately, no one seems to provide new insights. The current limitations of this application are concerned with the representation and we are currently investigating the possibility to introduce superposition information in the background knowledge as a mean to circumvent these problems.

Appendix

A.1. Protein folds background knowledge.

This appendix lists the predicates constituting the background knowledge.

A.1.1. Global information (attribute-valued).

`len_interval(Lo =< Dom =< Hi)` when `Lo` and `Hi` variables are both instantiated, `len_interval` is true if the length of the domain `Dom` is in the range `Lo` to `Hi`. Otherwise, `Lo` is bound to the length of the smallest positive example and `Hi` is bound to the length of the longest positive example.

`nb_alpha_interval(Lo =< Dom =< Hi)` similar to `len_interval` but process the number of alpha helices.

`nb_beta_interval(Lo =< Dom =< Hi)` similar to `len_interval` but process the number of beta strands.

A.1.2. Relational information.

`adjacent(Dom, S1, S2, Loop, TypeS1, TypeS2)` if `S1` and `S2` are both instantiated this predicate returns true if the type of secondary structure `S1` is `TypeS1` and `S2` is `TypeS2`, and the length of the loop separating `S1` and `S2` is `Loop` plus or minus an allowed delta. Otherwise, `S1` and `S2` are bound to two consecutive secondary structure elements, `Loop, TypeS1` and `TypeS2` are bound appropriately. Through backtracking all successive pairs are found.

A.1.3. Local information.

`unit_len(S,Cst)` is true if the length of the secondary structure `S` is `Cst`, the values for `Cst` are `very_lo`, `lo`, `hi` and `very_hi`.

`unit_aveh(S,Cst)` similar to `unit_len` but process the average hydrophobicity.

`unit_hmom(S,Cst)` similar to `unit_len` but process the hydrophobic moment.

`unit_pro(S)` is true if `S` contains a proline amino acid, the presence of a proline is known to disrupt secondary structure.

`coil(S1,S2,Len)` bounds `Len` to the length of the loop between secondary structures `S1` and `S2` or is true if the length of the loop is `Len` plus or minus 50%.

Acknowledgments

This work is supported by a BBSRC/EPSRC Bioinformatics grant. This work was supported also by the Esprit Long Term Research Action ILP II (project 20237), EPSRC grant GR/K57985 on Experiments with Distribution-based Machine Learning and an EPSRC Advanced Research Fellowship held by the second author.

Notes

1. Available from ftp://ftp.cs.york.ac.uk/pub/ML_GROUP/progol4.4 .
2. Available from http://www.rulequest.com/.
3. http://www.bmm.icnet.uk/ilp/ML2000.tar.gz .

References

Bashford, D., Chothia, C., & Lesk, A. M. (1987). Determinants of a protein fold. Unique features of the globin amino acid sequences. *Journal of Molecular Biology*, *196*(1), 199–216.

Bourne, P. E. (1998). Editorial. *Bioinformatics*, *15*(9), 715–716.

Branden, C. & Tooze, J. (1999). *Introduction to protein structure*. Garland.

Brenner, S. E., Chothia, C., Hubbard, T. J., & Murzin, A. G. (1996). Understanding protein structure: Using SCOP for fold interpretation. *Methods in Enzymology*, *266*, 635–643.

Finn, P., Muggleton, S., Page, D., & Srinivasan, A. (1998). Pharmacophore discovery using the inductive logic programming system Progol. *Machine Learning*, *30*, 241–271.

Hutchinson, E. G. & Thornton, J. M. (1996). PROMOTIF—a program to identify and analyze structural motifs in proteins. *Protein Science*, *5*(2), 212–220.

Kelley, L. A., MacCallum, R. M., & Sternberg, M. J. E. (2000). Enhanced genome annotation using structural profiles in the program 3D-pssm, *Journal of Molecular Biology*, *299*(2), 510–522.

Kim, S.-H. (1998). Shining a light on structural genomics. *Nature Structural Biology*, Synchrotron supplement: 643–645.

King, R., Muggleton, S., Lewis, R., & Sternberg, M. (1992). Drug design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proceedings of the National Academy of Sciences*, *89*(23), 11322–11326.

King, R., Muggleton, S., Srinivasan, A., & Sternberg, M. (1996). Structure-activity relationships derived by machine learning: The use of atoms and their bond connectives to predict mutagenicity by inductive logic programming. *Proceedings of the National Academy of Sciences*, *93*, 438–442.

Kuntz, I. D. (1972). Protein folding. *Journal of the American Chemical Society*, *94*(11), 4009–4012.

Langley, P. (1998). The computer-aided discovery of scientific knowledge. In *Proceedings of the First International Conference on Discovery Science*, Fukuoka, Japan: Springer-Verlag.

Muggleton, S. & Firth, J. (in press). CProgol4.4: Theory and use. In S. Džeroski & N. Lavrac (Eds.), *Inductive Logic Programing and Knowledge Discovery in Databases*.

Muggleton, S., King, R., & Sternberg, M. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Engineering*, *5*(7), 647–657.

Muggleton, S. & De Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, *19/20*, 629–679.

Orengo, C. A., Jones, D. T., & Thornton, J. M. (1994). Protein superfamilies and domain superfolds. *Nature*, *372*(6507), 631–634.

Pauling, L., Corey, R. B., & Branson, H. R. (1951). The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. USA*, *37*, 205–210.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann.

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.

Srinivasan, A., King, R. D., Muggleton, S. H., & Sternberg, M. (1997). Carcinogenesis predictions using ILP. In N. Lavrač & S. Džeroski (Eds.), *Proceedings of the Seventh International Workshop on Inductive Logic Programming* (pp. 273–287). Berlin: Springer-Verlag, LNAI 1297.

Srinivasan, A., Muggleton, S., King, R., & Sternberg, M. (1996). Theories for mutagenicity: A study of first-order and feature based induction. *Artificial Intelligence*, *85*(1/2), 277–299.

Sternberg, M., King, R., Lewis, R., & Muggleton, S. (1994). Application of machine learning to structural molecular biology. *Philosophical Transactions of the Royal Society B*, *344*, 365–371.

Wierenga, R. K., Terpstra, P., & Hol, W. G. J. (1986). Prediction of the occurence of the ADP-binding $\beta-\alpha-\beta$-fold in proteins, using and amino acid sequence fingerprint. *Journal of Molecular Biology*, *187*, 101–107.

Received March 31, 1999
Revised December 13, 1999 & April 12, 2000
Accepted June 1, 2000
Final manuscript June 1, 2000