Application of Inductive Logic Programming to Discover Rules Governing the Three-Dimensional Topology of Protein Structure

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
{M.Turcotte, M.Sternberg}@icrf.icnet.uk
[2] University of York, Department of Computer Science
Heslington, York, YO1 5DD, UK
stephen@cs.york.ac.uk

Abstract. Inductive Logic Programming (ILP) has been applied to discover rules governing the three-dimensional topology of protein structure. The data-set unifies two sources of information; SCOP and PROMOTIF. Cross-validation results for experiments using two background knowledge sets, global (attribute-valued) and constitutional (relational), are presented. The application makes use of a new feature of Progol4.4 for numeric parameter estimation. At this early stage of development, the rules produced can only be applied to proteins for which the secondary structure is known. However, since the rules are insightful, they should prove to be helpful in assisting the development of taxonomic schemes. The application of ILP to fold recognition represents a novel and promising approach to this problem.

1 Introduction

Classification is an important activity in all scientific areas. In the case of protein structures, the task is complex and at the moment is best performed by human experts. However, the number of known protein structures is increasing rapidly which creates a need for automatic methods of classification. The work presented here focuses on one level of the classification, fold recognition. Inductive Logic Programming (ILP) has been applied to derive new principles governing the formation of protein folds such as common substructures and the relationship between local sequence and tertiary structure.

The tertiary structure of proteins is itself arranged hierarchically. The building blocks, the amino acids, also termed residues, assemble linearly to form the primary structure or sequence. Sequence segments adopt regular conformations, helices and strands, collectively called secondary structures. ILP has previously been applied to prediction of protein secondary structure [1, 2]. Secondary structures form motifs called supersecondary structures. Finally, these interact together to form the tertiary structure.

Protein fold recognition involves finding the fold relationship of a protein sequence of unknown structure. Two proteins have a common fold if they share the same core secondary structures, and the same interconnections. Homologous proteins share the same or similar folds. Protein fold recognition focuses on analogous proteins and remote homologues. It allows the inference of a relationship between two proteins that could not be inferred by direct sequence comparison methods. Thus it allows structural and possibly functional information to inferred for new protein sequences.

Approaches to protein fold recognition can be classified into two broad classes. The first class of approaches considers sequential information, see [3, 4] for a review. Information, often called profile, is derived from the primary sequence, secondary structure and solvent accessibility. A database of target profiles is built for all known folds using experimental data. Dynamic programming algorithm is then used to align two profiles, probe and target. The information about the probe is derived from prediction methods. The second class of approaches uses pair potentials which sums all the propensity scores of residues pairs at a certain distance, see [5] for a review.

Machine learning techniques have been applied to the problem. Hubbard *et al.* [6] used Hidden Markov Models (HMM) to create a model of a multiple sequence alignment which is subsequently used to retrieve related sequences from the protein structure database. The HMM scores are combined with those obtained by comparing the predicted secondary structure, based on the multiple sequence alignment, to the experimental secondary structure. Di Francesco *et al.* [7] also used HMM but at a different stage of the recognition process. They used them to built a model of all observed secondary structures of a given fold. Secondary structure is predicted for the probe and evaluated using the models of all targets. Rost and Sander [8] used neural networks and information from predicted secondary structure plus predicted solvent accessibility. These methods are based on sequence alignment, either of the polypeptide chain, secondary structure or solvent accessibility, they resemble to the first approach mentioned above.

A different approach has been undertaken by Dubchak *et al.* [9], they used neural networks together with global descriptors. The global descriptors are composition, transition and distribution of physico-chemical properties such as hydrophobicity, polarity and predicted secondary structure.

Our approach combines both, sequential and global descriptors. This paper is organised as follows. Section 2 gives the details of the ILP system we used. Section 3 introduces the new data-set. Section 4 presents the results of the learning experiments. We conclude, Section 5, with a discussion of the advantages of ILP to resolve the problem at hand and discuss the anticipated future developments.

2 ILP System

The experimentation was carried out using Progol4.4 which is the latest of the family of Progol ILP systems [10]. Progol4.4 is distinguished from its predecessors

Progol4.1 and Progol4.2 by its use of constraints for numeric parameter estimation. This is an adaptation of the 'lazy-evaluation' mechanism, first proposed by Srinivasan and Camacho. For instance, suppose we want to find upper bounds for the predicate `lessthan/2`. First, use declarations such as the following.

```
:- modeb(1,interval(#float =< +float =< #float))?
:- constraint(interval/1)?
```

The constraint declaration says that any `interval/1` atom in the most-specific clause should have a Skolem constant in place of the upper and lower bound constants (#float's above). In the search, during any refinement which introduces a constraint atom, the flag solving is turned on, and the user-defined predicate is given all substitutions from positive and negative examples as a list of lists of lists (takes the form [P,N] where P is from the positive examples and N from the negatives, and P,N are lists of lists, each list giving all substitutions related to a particular example), and returns an appropriate substitution for the constant. This constant is used in place of the Skolem constant in subsequent testing of the refined clause and its refinements. Thus definitions for constraint predicates have to have at least two clauses, having the guards 'solving' and 'not(solving)' to define respectively the procedure for computing the parameter and the normal application of the predicate.

3 Data-Set

A Prolog database has been constructed by translating automatically the output of the computer program PROMOTIF [11] and the database SCOP [12].

The data-set is meant to be used throughout several projects. The translation retains most of the structure of the data. Program transformations are later applied to reformat the data for each specific project.

In principle, the data-set can be used to learn any feature implemented by SCOP and PROMOTIF. In practice, because learning experiments necessitate supervision, we are aiming at only learning Prolog definition for a limited subset of these features. However, the data are being made available in the hope that it will encourage further experimentation[1].

The data-set should be useful on its own. Prolog has already proven to be an excellent tool to manage protein structure databases [13, 14].

3.1 Structure Classification

The classification of protein structures is a complex task. The main classification schemes are SCOP [12] and CATH [15]. The former classification is performed manually while the second is semi-automated. For this work we refer to SCOP, it is used to relate structures and folds.

[1] See `http://www.icnet.uk/bmm/people/turcotte/ilp98/`

The basic unit in SCOP is a domain, a structure or substructure that is considered to be folded independently. Small proteins have a single domain, for larger ones, a domain is a substructure, also termed region, indicated by a chain id and a sequence interval range. Domains are grouped into families. Domains of the same family have evolved from a common ancestral sequence. In most cases, the relationship can be identified by direct sequence comparison methods. The next level is called a superfamily. Members of a superfamily are believed to have evolved from a common ancestry, but the relationship cannot always be inferred by sequence comparison methods alone; the expert relies on other evidences, functional features for example. The next level is a fold, proteins share the same core secondary structures, and the same interconnections. The resemblance can be attributed to convergence towards a stable architecture. Finally, folds are conveniently grouped into classes (such as all-α and all-β) based on the overall distribution of their secondary structure elements. Figure 1 shows the Prolog representation of a SCOP entry.

```
scop(Class, Fold, Super, Family, Protein, Species, PdbId, Region)
```

Fig. 1. A domain entry in the SCOP database.

A Perl program has been written which creates a Prolog representation from a SCOP HTML file. In this work we used SCOP database 1.35 generated by scopm 1.087. The four major classes contain 9153 domains, covering 630 families and 298 folds.

3.2 Structural Attributes

PROMOTIF is used to calculate the structural attributes used in this work. Given a set of coordinates, PROMOTIF generates a series of files, each containing a particular set of structural features. These features are secondary structure, β- and γ-turns, helical geometry and interactions, β strands and β-sheet topology, β-bulges, β-hairpins, β-α-β units, ψ-loops and main-chain hydrogen bonding patterns [11].

A Perl program has been written that translates these free format files to Prolog clauses, Fig. 2 shows a sub-set of PROMOTIF attributes.

3.3 Selection

Because crystallographers[2] do not select randomly the proteins they study, the databases are biased. Crystallographers select proteins because of their connection to a particular molecular pathway, a particular disease or their overall scientific interest in general.

[2] Crystallography, the study of X-ray diffraction patterns, is the main source of our knowledge of protein structure.

```
sst(Pdb, Chain, Pos, Aa, Structure).
helix(Pdb, Num, Chain, Pos, Chain, Pos, Len, Type).
helix_pair(Pdb, Num, Num, Dist, Angle, Region, Region, Num, Num).
strand(Pdb, Num, Sheet, Chain1, Lo, Chain2, Hi, Len).
hairpin(Pdb, Beta1, Beta2, Len1, Len2).
sheet(Pdb, Label, N, Type).
bturn(Pdb, Chain, Pos, Type).
```

Fig. 2. Examples of PROMOTIF attributes represented as Prolog clauses.

To remove redundancy a single representative domain has been selected per protein, as defined in SCOP. The procedure was carried out with the computer system Darwin [16]. All sequences of a protein are gathered and compared all against all. The sequence with maximum average similarity score to all other members of the set is selected as the representative element.

Next, to ensure enough diversity, folds having less than 5 families were removed. Table 1 lists all the selected folds and the cross-validation accuracy measures.

Negative examples were chosen randomly from proteins of the same class but having a different fold. The rational is that it is more difficult to distinguish between two folds of the same class than it is to distinguish between folds of different classes and is justified by the existence of accurate method for class prediction. Finally, in accord with previous experiments we selected the number of negative examples proportional to the number of positive examples.

Rules were learnt that discriminate between members and non-members of a fold. The expected accuracy of a random prediction should be 50%.

4 Learning Experiments

Progol was applied to all folds using two background knowledge sets. The first set involved global attributes of protein structure. The second included constitutional information as well. For each fold, rules were learnt that discriminate between members (positive examples) and non-members (negative examples).

4.1 Global Attributes

We first present a learning experiment that involves only global attributes. Learning here is essentially attribute-value based. We used the total number of residues, total number of secondary structures of both types, β and α, and three different constraints. Figure 3 lists them all. As we will see, it is possible to derive rules that are effective and in some cases provide interesting insights.

For 17 out of 23 folds, Progol produced a descriptive rule; indeed, in this experiment, Progol produced a single rule per fold, with overall cross-validation accuracy of 70.76%, see Table 1. One such rule is that of the Immunoglobulin-like β-sandwich fold. This single rule covers most of the positive examples and

Table 1. Cross-validation predictive accuracy measures for global and combined information for all folds.

Folds	Fam	Dom	Global Acc (%)	Combined Acc (%)
All-α:				
Four-helical bundle	7	12	95.83 ± 4.08	95.83 ± 4.08
EF Hand-like	7	14	78.57 ± 7.75	78.57 ± 7.75
Three-helical bundle	13	27(26)	90.57 ± 4.02	90.57 ± 4.02
All-β:				
Diphtheria toxin	5	6	50.00 ± 14.43	41.67 ± 14.23
Barrel-sandwich	4	8	- ± -	68.75 ± 11.59
beta-Trefoil	5	9	66.67 ± 11.11	66.67 ± 11.11
ConA-like	5	8	75.00 ± 10.83	50.00 ± 12.50
SH3-like barrel	5	13	84.62 ± 7.08	73.08 ± 8.70
OB-fold	9	18	- ± -	61.11 ± 8.12
Immunoglobulin†	13	41	78.75 ± 4.57	71.25 ± 5.06
α/β:				
Restriction endonucleases	5	5	20.00 ± 12.65	80.00 ± 12.65
alpha/beta-Hydrolases	8	9	- ± -	55.56 ± 11.71
Ribonuclease H-like motif	11	16	43.75 ± 8.77	75.00 ± 7.65
Flavodoxin-like	11	14	67.86 ± 8.83	60.71 ± 9.23
P-loop	4	15	- ± -	50.00 ± 9.13
Rossmann-fold	7	20	80.00 ± 6.32	72.50 ± 7.06
(TIM)-barrel†	24	49	80.00 ± 4.22	78.89 ± 4.30
$\alpha+\beta$:				
FAD-linked reductases	5	5	100.00 ± 0.00	90.00 ± 9.49
Lysozyme-like	6	7	92.86 ± 6.88	100.00 ± 0.00
Cystatin-like	5	7	35.71 ± 12.81	71.43 ± 12.07
Metzincin-like	6	11	77.27 ± 8.93	86.36 ± 7.32
beta-Grasp	6	13	- ± -	42.31 ± 9.69
Ferredoxin-like	17	21	- ± -	61.90 ± 7.49
Overall:			70.76 ± 1.79	71.53 ± 1.72

† 10-fold cross validation, other values were obtained by leave-one-out procedure.
Fam is total number of families.
Dom is total number of domains (positive examples), in the case of three-helical bundle, the number of negative examples is one less because of a shortage of data.
Acc is the cross-validation accuracy, defined as the sum of true positives and true negatives over the total. In some cases, Progol was unable to infer any rule, this is indicated with minus sign.
The overall cross-validation accuracy values are calculated from the sum of all the contingency tables, thus it also accounts for cases where Progol was not able to produce any rule. ± values are standard errors of cross-validation accuracy.

```
:- modeh(1,fold(#fold_t,+dom_t))?           % relates folds and domains
:- modeb(1,len(+dom_t,-nat))?               % total number of residues
:- modeb(1,nb_alpha(+dom_t,-nat))?          % total number of helices
:- modeb(1,nb_beta(+dom_t,-nat))?           % total number of strands
:- modeb(1,interval(#nat =< +nat =< #nat))?
:- modeb(1,interval_l(+nat =< #nat))?
:- modeb(1,interval_r(#nat =< +nat))?
```

Fig. 3. Mode declarations.

says that a domain adopts an Immunoglobulin-like β-sandwich fold if its length, measured in number of residues, is between 50 and 173, has one or no α-helix and seven to ten β-strands. The Prolog representation is shown in Fig. 4.

```
fold('Immunoglobulin-like beta-sandwich',A) :-
    len(A,B), interval(50=<B=<173),
    nb_alpha(A,C), interval(0=<C=<1),
    nb_beta(A,D), interval(7=<C=<10).
```

Fig. 4. The rule induced by Prolog for the Immunoglobulin-like β-sandwich fold.

Three rules were produced that are of a particular interest. They say that for these folds there is a significant number of cases where the number of helices and strands is the same. The relation is not trivial as the total number of secondary structures also varies (see Fig. 5). In the case of β/α (TIM)-barrel the rule says that the number of α-helices is the same as the number of β-strands and this number is between eight and sixteen. It suggests that these folds are made of repetitive motifs, this can be programmed in the background knowledge and will be tested in future experiments.

```
fold('Flavodoxin-like',A) :-
    nb_alpha(A,B), nb_beta(A,B), interval_l(B=<6).

fold('NAD(P)-binding Rossmann-fold domains',A) :-
    nb_alpha(A,B), nb_beta(A,B), interval(5=<B=<7).

fold('beta/alpha (TIM)-barrel',A) :-
    nb_alpha(A,B), nb_beta(A,B), interval(8=<B=<16).
```

Fig. 5. Same number of strands and helices. The top rule says that a domain `A` adopts a `Flavodoxin-like` fold if it has `B` helices, `B` strands and `B` is less than or equal to 6. All three folds belong to the same class, α/β.

However good these rules are to discriminate between folds, they do not provide much structural insights; although the three rules in Fig. 5 suggest an element of symmetry. We recall that our objective is to derive new principles governing the formation of protein folds and thus we now move on to a more complex representation which facilitates their discovery.

4.2 Combined Attributes

We now present the second learning experiment that incorporates constitutional (relational) as well as global information (attribute-value). New attributes are introduced. The predicate `adjacent/6` serves three purposes. First, the predicate is used to introduce two secondary structure identifiers in a rule. Second, the predicate tells us that the two units are consecutive. It gives the location of the first element, the location is allowed to vary slightly and this variation depends on its position in the sequence, the closer to the end the more variation is allowed. Finally, the predicate also indicates the secondary structure type of each unit.

Two consecutive secondary structures are separated by a coil, a sequence of amino acids which varies in length, this information is represented by the predicate `coil/3`.

Three properties of secondary structures have been considered here: average hydrophobicity, hydrophobic moment and length, respectively represented by `ave_h/2`, `h_mom/2` and `unit_len/2`. The numerical values of the parameters were substituted by symbolic constants. The constant `very_hi` was assigned if the value of the parameter was greater than or equal to the mean plus two standard deviations, `hi` if the value was greater than or equal to the mean plus one standard deviation, `very_lo` if the value was less than or equal to the mean minus two standard deviations and `lo` if the the value of the parameter was less than or equal to the mean minus one standard deviation. Values between the mean minus one standard deviation and the mean plus one standard deviation were omitted to speed up the calculations. Following the same line or reasoning as Section 3.3, the mean and standard deviation were calculated for proteins of the same class.

In the previous section, rules were obtained for 17 out of 23 folds. With the new attributes, we now have rules for all the folds. The overall accuracy is not significantly higher, see Table 1. Previously, one rule per fold was produced, we now have some folds having up to three rules.

Figures 7 and 6 illustrate the format of rules produced. We recall that the aim of the learning process is to discriminate between members and non-members of a fold where negative examples are selected from elements of the same class. In the four-helical up-and-down bundle, the distinctive feature is the presence of a rather long helix around position five, followed by another helix. In EF-hand, it is the strand/helix pair located at the start or at the end of the molecule which is the distinctive feature. Finally, in DNA-binding 3-helical bundle, two populations are represented, the largest one is distinguished by its length and the fact that it has exactly three helices. The second population is distinguished

by its pair of β-strands, connected by a short coil, located at the beginning of the molecule.

```
fold('Four-helical up-and-down bundle',A) :-
    adjacent(A,B,C,5,h,h), unit_len(B,hi).

fold('EF Hand-like',A) :-
    adjacent(A,B,C,1,h,e), nb_alpha(A,D), interval(4=<(D=<9)).

fold('EF Hand-like',A) :-
    adjacent(A,B,C,9,e,h).

fold('DNA-binding 3-helical bundle',A) :-
    len(A,B), interval(38=<(B=<111)),
    nb_alpha(A,C), interval(3=<(C=<3)).

fold('DNA-binding 3-helical bundle',A) :-
    adjacent(A,B,C,2,e,e), coil(B,C,4).
```

Fig. 6. Prolog representation of the rules of all-α class.

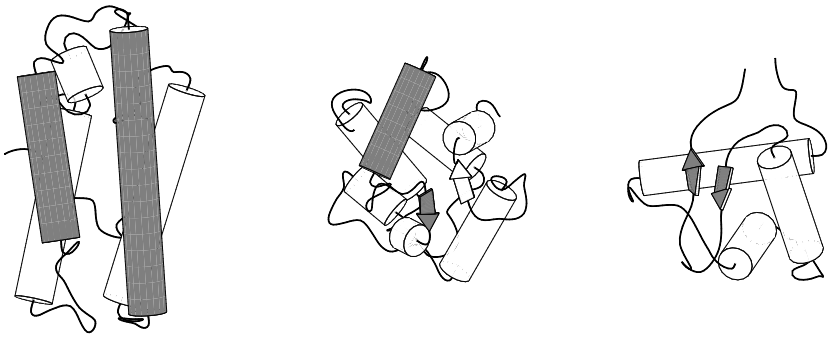
5 Conclusion

We have applied Inductive Logic Programming to the problem of fold recognition. This work is preliminary, but it shows that Progol is capable of producing rules that are both accurate and descriptive. The rules produced are non-trivial. Nevertheless they are easily interpretable by the expert in terms of structural concepts: edge strands, hairpins, etc.

The learning experiment was presented in two steps, global and constitutional attributes, two rather different ways to describe proteins. Attribute learners, such as C4.5 or CART, are suitable for use with global attributes, but it would be difficult to introduce concepts related to constitutional information. Other machine learning techniques address this problem. Hidden Markov Models, for example, are most suitable for this form of information. Here, we have shown an application that integrates both types of information transparently, and often this information has been used in a complementary way.

To tackle this problem a new database has been built that unifies multiple sources of information. The database is general and allows queries that involve structural features and taxonomic information. We hope that it will be useful both inside and outside the machine learning community.

The work presented here also raises interesting questions. It suggests that it is possible to distinguish between folds using small patterns of secondary structure. These patterns are present in most, or all, proteins of a fold but



(a) (b) (c)

Fig. 7. Schematic representation of the domains of the all-α class. The structural features used for the construction of the rules are shaded. (a) Four-helical up-and-down bundle, (b) EF Hand-like and (c) DNA-binding 3-helical bundle. The cylinders represent α-helices, the arrows represent β-strands and the coil regions are represented as a thin line.

not in others. It would be interesting to know if these also correspond to well defined and predictable segments. Since these patterns are conserved, it is sound to postulate that the multiple sequence alignment in these regions will be well defined as well and should be suitable for evolutionary based secondary structure prediction methods. Such method for fold recognition would depend less on the overall accuracy of the secondary structure prediction program used. The use of experimental secondary structure is justified in the context where it is used as an aid to develop taxonomic schemes. The evaluation of the method using predicted secondary structure is the next step in this project. In addition, for protein fold predictions, these rules could be used in conjunction with other fold recognition methods, based on profiles or pair potentials.

Rules have been learnt independently for each fold. As a result, the fold predictions from different sets of rules are overlapping. We need to quantify this overlap but most importantly we need to consider a resolution mechanism. First-order decision lists paradigm, as described by Mooney and Califf [17], is a good candidate. Rules are ordered in increasing level of coverage. Rules with low coverage are encountered first. They are considered as exception to more general rules. In [17], each rule ends with a cut, hence producing a single answer. In the field of protein recognition it is most common to return the list of all possible folds.

Several developments are planned. The relation between the number of strands and helices detected by Progol suggested that symmetry and segmentation should

be added to the background knowledge. Another improvement would be to make a more effective use of the structural information available. For example, we want to make use of structural alignments and equivalence between secondary structures. The richness of this paradigm allows to express hierarchical and non-local information. This is a direction of research that we intend to investigate further. We particularly want to explore the recognition of common substructures.

Acknowledgement

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. MT receives a fellowship from *Fonds pour la formation de chercheurs et l'aide à la recherche*, Québec, Canada. The authors wish to thank C. Bryant for his careful reading of the manuscript and comments.

References

[1] S. Muggleton, R. King, and M. J. E. Sternberg. Protein secondary structure prediction using logic-based machine learning. *Protein Engineering*, 5(7):647–657, 1992.

[2] M. J. E. Sternberg, R. D. King, R. A. Lewis, and S. Muggleton. Application of machine learning to structural molecular biology. *Philosophical Transactions of the Royal Society of London - Series B: Biological Sciences*, 344(1310):365–71, 1994.

[3] D. Fischer and D. Eisenberg. Protein fold recognition using sequence-derived predictions. *Protein Science*, 5:947–955, 1996.

[4] R. B. Russell, M. A. S. Saqi, P. A. Bates, R. A. Sayle, and M. J. E. Sternberg. Recognition of analogous and homologous protein folds - assessment of prediction success and associated alignment accuracy using empirical substitution matrices. *Protein Engineering*, 11(1):1–9, 1998.

[5] S. H. Bryant. Evaluation of threading specificity and accuracy. *Proteins*, 26(2):172–185, 1996.

[6] T. J. Hubbard and J. Park. Fold recognition and ab initio structure predictions using hidden markov models and beta-strand pair potentials. *Proteins Struct. Funct. Genet.*, 23(3):398–402, 1995.

[7] V. Francesco, Di, J. Garnier, and P. J. Munson. Protein topology recognition from secondary structure sequences: Application of the hidden markov models to the alpha class proteins. *Journal of Molecular Biology*, 267(2):446–463, 1997.

[8] B. Rost, R. Schneider, and C. Sander. Protein fold recognition by prediction-based threading. *Journal of Molecular Biology*, 270:471–480, 1997.

[9] I. Dubchak, I. Muchnik, and S.-H. Kim. Protein folding class predictor for SCOP: approach based on global descriptors. *ismb*, 5:104–107, 1997.

[10] S. Muggleton, editor. *Inductive Logic Programming*. Academic Press, 1992.

[11] E. G. Hutchinson and J. M. Thornton. PROMOTIF – a program to identify and analyze structural motifs in proteins. *Protein Science*, 5(2):212–20, 1996.

[12] S. E. Brenner, C. Chothia, T. J. Hubbard, and A. G. Murzin. Understanding protein structure: using SCOP for fold interpretation. *Methods in Enzymology*, 266:635–43, 1996.

[13] C. J. Rawlings, W. R. Taylor, J. Fox J. Nyakairu, and M. J. E. Sternberg. Using Prolog to represent and reason about protein structure. In Ehud Y. Shapiro, editor, *Third International Conference on Logic Programming*, volume 225 of *Lecture Notes in Computer Science*, pages 536–543. Springer, 1986.

[14] G. J. Barton and C. J. Rawlings. A Prolog approach to analysing protein structure. *Tetrahedron Computer Methodology*, 3(6C):739–756, 1990.

[15] C.A. Orengo, A.D. Michie, S. Jones, D.T. Jones, M.B. Swindells, and J.M. Thornton. CATH – a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108, 1997.

[16] G. H. Gonnet and S. A. Benner. Computational biochemistry research at ETH. Technical report, E.T.H. Department Informatik, March 1991.

[17] R.J. Mooney and M.E. Califf. Induction of first-order decision lists: Results on learning the past tense of english verbs. *Journal of Artificial Intelligence Research*, 3:1–24, 1995.