Multi-Class Protein Fold Recognition using Large Margin Logic based Divide and Conquer Learning

Huma Lodhi
Department of Computing
Imperial College London
London, UK
h.lodhi@imperial.ic.ac.uk

Stephen Muggleton
Department of Computing
Imperial College London
London, UK
s.muggleton@.imperial.ac.uk

Mike J. E. Sternberg
Division of Molecular Biosciences
Imperial College London
London, UK
m.sternberg@imperial.ac.uk

ABSTRACT

Inductive Logic Programming (ILP) systems have been successfully applied to solve complex problems in bioinformatics by viewing them as binary classification tasks. It remains an open question how an accurate solution to a multi-class problem can be obtained by using a logic based learning method. In this paper we present a novel logic based approach to solve complex and challenging multi-class classification problems by focusing on a key task, namely protein fold recognition. Our technique is based on the use of large margin methods in conjunction with the kernels constructed from first order rules induced by an ILP system. The proposed approach learns a multi-class classifier by using a divide and conquer reduction strategy that splits multi-classes into binary groups and solves each individual problem recursively hence generating an underlying decision list structure. The method is applied to assigning protein domains to folds. Experimental evaluation of the method demonstrates the efficacy of the proposed approach to solving multi-class classification problems in bioinformatics.

1. INTRODUCTION

The underlying aim of a multi-class approach is to learn a highly accurate function that categorizes examples into predefined classes. Effective multi-class techniques are crucial to solving the challenging and complex problems in bioinformatics such as multi-class protein fold recognition.

The two areas of machine learning, namely Inductive Logic Programming (ILP) and Kernel based methods (KMs) are well known for their distinguishing features: ILP techniques are characterized by their use of background knowledge and expressive language formalism whereas strong mathematical foundations and high generalization ability are remarkable characteristics of KMs. Recently some useful techniques (Support Vector Inductive Logic Programming (SVILP) [8], kFOIL [6] and RUMBLE [10]) have been designed by exploiting the characteristics of KMs and ILP to solving binary classification problems and performing real-valued predictions. In this paper we study multi-class classification in the combined ILP and kernel based learning scenario by extending SVILP for bioinformatics tasks.

SVILP solves binary classification problems in a multi-stage learning process. In the first stage, a set of first order Horn clauses (rules) are obtained from an ILP system. In the next stages similarity between examples is computed by the use of a novel kernel function that captures semantic and structural commonalities between the examples. The computed relational and logic based kernel is used in conjunction with a large margin learning algorithm to induce a binary classifier. In this way, SVILP performs classification task by training a large margin first order classifier.

In order to solve multi-class problems we propose a simple but accurate approach. The method is designed by reducing multi-class classification task to binary problems. However our approach is different from the existing reduction techniques as it learns hidden structure and characteristics from data and hence improves the performance of the classifier. The proposed method is based on a divide and conquer strategy and it discriminates different classes by using an underlying structure based on decision lists. The multi-class problem is reduced by recursively breaking it down into binary problems where each binary task is solved by invoking an SVILP machine. At each node of the decision list the algorithm induces a classifier and updates the training set by removing the examples of the class chosen at the previous node. A label is assigned to a new example by traversing the list.

The recognition of proteins having similar structure is a challenging and complex task in bioinformatics. It has key importance in studying protein structure and function and can provide answers to biological problems. In fold recognition, labels are assigned to proteins from a set of predefined annotations (labels, folds). In this way protein fold recognition can be viewed as the multi-class classification task. The aim of a protein fold classification system is to assign proteins to one of many folds with high accuracy. Machine learning methods have been applied to investigate the problem. The studies reported in [11; 3]) applied Support Vector Machines (SVMs) to solving multi-class protein fold classification problem. Chen and Kurgan [1] and Shen and Chou [12] studied ensemble methods to assign proteins to 27 folds from SCOP [9]. In this paper we present a novel logic based approach to solving protein fold recognition problem. We also compare the proposed approach with standard multi-class logic based method and multi-class SVMs. The experimental results demonstrate the efficacy of the proposed technique in assigning protein folds.

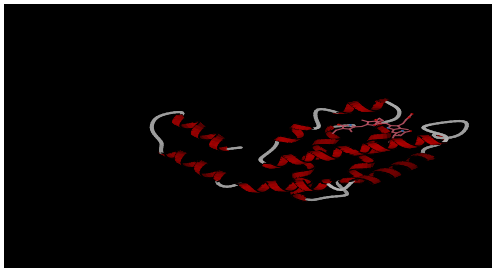

Figure 1: Protein domain 'd1all_'

dom_t(d1alla_).
len(d1alla_, 161). nb_alpha(d1alla_,7).
nb_beta(d1alla_,0). has_pro(d1alla_h1).
sec_struc(d1alla_, d1alla_h3). unit_t(d1alla_h3).
sst(d1alla_h3,4,4,a,104,9,h,0.443,3.003,116.199,
[v,t,p,i,e,e,i,g,v]). unit_hmom(d1alla_h2, hi). ···

Figure 2: Relationally encoded features of protein domain. 'd1all_'.

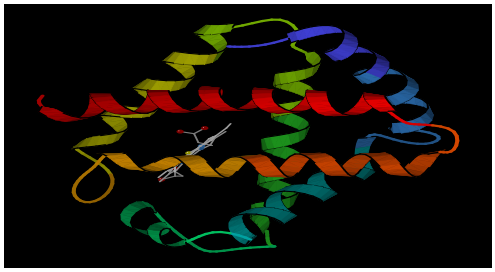

Figure 3: Protein domain 'd2hbg__'

dom_t(d2hbg__).
len(d2hbg__, 147). nb_alpha(d2hbg__,6).
nb_beta(d2hbg__,0). has_pro(d2hbg__h5).
sec_struc(d2hbg__, d2hbg__h2). unit_t(d2hbg__h2).
sst(d2hbg__h2,3,3,blank,40,7,h,0.540,1.812,213.564,
[q,m,a,a,v,f,g]). ·····

Figure 4: Relational encoded features of protein domain 'd2hbg__'.

2. MULTI-CLASS INDUCTIVE LOGIC PROGRAMMING (MC_ILP)

ILP systems have been successfully applied to binary classification tasks in bioinformatics. There are few ILP systems that can perform multi-class classification tasks [5]. The standard multi-class logic based method, described below, is biased towards the majority class. The method is based on learning theories H_j(first order horn clauses) for each class j. The obtained theories for r classes are merged into a multi-theory H. For each class the number of correctly classified training examples are recorded. A class is assigned to a new example if the example satisfies the conditions of the rules. In the case that an example is predicted to have multiple classes, then the class with the maximum number of predicted training examples is assigned to the example. If an example fails to satisfy the conditions of all the rules in H, a default class (majority class) is assigned to it. The method is termed as multi-class ILP (MC_ILP).

3. SUPPORT VECTOR INDUCTIVE LOGIC PROGRAMMING

Support Vector Inductive Logic Programming [8] is a new machine learning technique that integrates Inductive Logic Programming and Support Vector Machines. SVILP learning can be viewed as a multi-stage induction process. The four stages that comprise SVILP learning are described as follows.

In the first stage a set of rules $\mathcal{H}$ is obtained from an ILP system that takes relationally encoded examples (positive, negative) and background knowledge as input. This stage maps the examples into a logic based relational space. A first order rule, $h \in \mathcal{H}$, can be viewed as a boolean function of the form, $h : D \rightarrow \{0, 1\}$.

In the next stage a subset $H \in \mathcal{H}$ is selected using an information theoretic measure, namely compression, described below. The subset of rules, H, is selected by thresholding the compression value. This stage maps the examples into another lower dimensional space containing the information relevant to the task at hand. The compression value of a rule is computed by the expression, $C = \frac{PT*(ps-(ng+c))}{ps}$, where ps is the number of positive examples correctly deducible from the rule, ng is the number of negative examples that satisfy the conditions of the rules, c is the length of the rule and PT is the total number of positive examples.

In the third stage a kernel function is defined on the selected set of rules that can be weighted/unweighted. The kernel is based on the idea of comparing two examples by means of structural and relational features they contain; the more features in common the more similar they are. The function is given by the inner product between the mapped examples where the mapping ϕ is implied by the set of rules H. The mapping ϕ for an example d is given by, [1]. $\phi : d \rightarrow \left(\sqrt{\pi(h_1(d))}, \sqrt{\pi(h_2(d))}, \ldots, \sqrt{\pi(h_t(d))}\right)'$, where $h_1, \ldots, h_t$ are rules and π is the weight assigned to each rule h_i. The construction embeds the data into a feature space, where dimensionality of the space is the same as the cardinality of the set of rules. In this way, an example is viewed as a column vector where each entry of the vector is indexed by a specific rule. The kernel for examples d_i and d_j is given by, $k(d_i, d_j) = \langle\phi(d_i), \phi(d_j)\rangle = \sum_{l=1}^{t} \sqrt{\pi(h_l(d_i))}\sqrt{\pi(h_l(d_j))}$. The kernel specified by an inner product between two mapped examples is a sum over all the common hypothesized rules. Given that ϕ maps the data into feature space spanned by ILP rules, we can construct Gaussian RBF kernels, $k_{RBF}(d_i, d_j) = \exp\left(\frac{-\|(\phi(d_i)-\phi(d_j)\|^2}{2\sigma^2}\right)$, where $\|(\phi(d_i)-\phi(d_j)\| = \sqrt{k(d_i, d_i) - 2k(d_i, d_j) + k(d_j, d_j)}$. In the final stage learning is performed by using an SVM in conjunction with ILP kernel. SVILP is flexible to construct any kernel in the space spanned by the rules. However, in the present work we used k_{RBF}.

[1] $'$ specifies column vector

fold(Globinlike,A)	←
	adjacent(A,B,C,1,h,h), adjacent(A,C,D,2,h,h), coil(B,C,4).
	/*A domain is classified 1 (belongs to Fold 'Globinlike') if helices B (at position 1) and C are adjacent, C (at position 2) and D are adjacent and length of loop connecting B and C is 4.*/
fold(Globinlike,A)	←
	adjacent(A,B,C,1,h,h), has_pro(C).
	/*A domain is classified 1 if helices B(at position 1) and C are adjacent and C has proline.*/
fold('Globinlike',A)	←
	adjacent(A,B,C,1,h,h), coil(B,C,4), nb_α_interval(4=<(A=<8)).
	/*A domain is classified 1 if helices B (at position 1) and C are adjacent, number of α helices are in range [4,8] and length of loop connecting B and C is 4*/.

Figure 5: Rules followed by English conversion for Protein domains in Globin-like fold.

We now show how ILP kernel measures similarity in logic and relational space by considering a pair of protein domains, '2hbg_' and '1alla__'. The domains belong to α structural class and 'Globin-like' fold (SCOP classification scheme). Figures 1, 2, 3 and 4 show the two domains and their relationally encoded features. Here predicates 'len', 'nb_alpha', and 'nb_beta' denote the length of the polypeptide chain, number of α-helices and β strands respectively. The other predicates represent the relationship between the secondary structure elements and their properties (hydrophobicity, the hydrophobic moment, the length of proline and etc.).
Figure 5 shows a set of induced rules together with their English conversion. A rule classifies an example positive (1) if it fulfils the conditions of the rule while an example that fails to satisfy the conditions is classified negative (0). The set of equally weighted rules maps the two examples as follows: $\phi(d1alla\_) = \phi(d1) = \begin{pmatrix} 1\times1 & 1\times1 & 1\times1 \end{pmatrix}'$ and $\phi(d2hbg\_\_) = \phi(d2) = \begin{pmatrix} 1\times1 & 0\times1 & 1\times1 \end{pmatrix}'$. Given that the rules are equally weighted, each entry of the vector is multiplied by 1. The kernel values between the examples are as follows: $k(d1,d2) = k(d2,d1) = 2$, $k(d1,d1) = 3$ and $k(d2,d2) = 2$.

4. SUPPORT VECTOR INDUCTIVE LOGIC PROGRAMMING BASED MULTI-CLASS CLASSIFICATION

We now propose a novel logic based method to solving multi-class classification problems like protein fold recognition. We apply inductive learning in which a learning algorithm is provided with a set of examples, D, of the form $D = \{(d_1,c_1),(d_2,c2),\ldots,(d_n,c_n)\}$ where d_i are training examples and $c_i \in \{1,2,\ldots,r\}$ are classes (labels). The goal of the classification algorithm is to generate a function $f : d \rightarrow \{1,2,\ldots,r\}$ that assigns a new example d to the class with low error probability.
In order to solve multi-class problems we apply powerful but simple divide and conquer strategy. The complex multi-class classification task is divided into binary problems and each problem is solved recursively. The method constructs a decision list as shown in figure 6. Here each non-leaf node has two children. Classes are represented by non leaf nodes where edges are labeled by the binary classifier's output. We term the divide and conquer technique as decision list based SVILP (DL_SVILP). The method is shown as Algorithm 1. The technique reduces multi-class classification problem to $r-1$ binary problems, where r is the total number of classes.

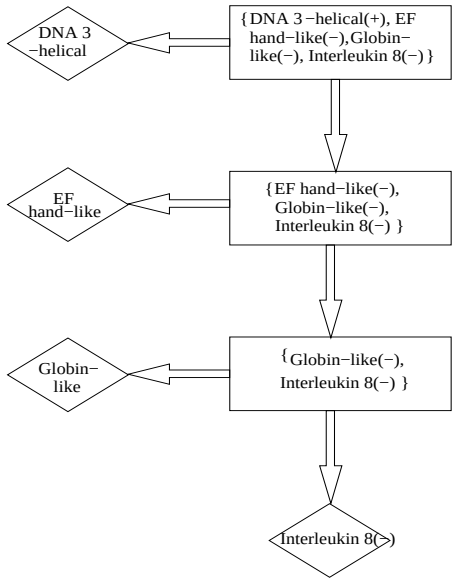

Figure 6: A decision list, learned by the large margin first order rule learner, for multi-class protein fold recognition.

The algorithm can be viewed as comprising $r-1$ iterations. In each iteration a class is selected as the positive class and the remaining classes are reduced to the negative class. The binary problem is solved by using a large margin first order rule learner. The training set is updated by removing the examples of the chosen class. In this way the root node contains all the classes whereas the node at depth $r-1$ contains two classes. The size of the training set used at depth $r-1$ is (much) smaller than the size of the training set for the root node. DL_SVILP assigns a class j to a new example d as follows:

1. Begin at the root node
2. Apply the classifier associated with the node to example d
3. Travel down the edge labeled by the classifier's output
4. If the edge is labeled positive output the class associated with the leaf. If the edge is labeled negative repeat steps 2 and 3 until the last positive edge is reached. Output the label given by the node.

Algorithm 1 Support Vector Inductive Logic Programming (DL_SVILP) for multi-class classification

Input: A set of training examples $\{(d_1, c_1), (d_2, c_2), \ldots, (d_n, c_n)\}$, where $d_i \in D$ and $c_i \in \{1, 2, \ldots, r\}$ and a vector *index* that represents learned structure of the list.
for $j = 1$ to $r - 1$ **do**
/* Select a class p from r classes */
$p = index[j]$
/* Formulate the binary class problem by assigning label '1' to examples of class p and '-1' to examples of remaining classes */
$D_i = \{(d_1, c_1), (d_2, c_2), \ldots, (d_n, c_n)\}$, where $d_i \in D$ and $c_i \in \{1, -1\}$
/* Induce a binary classification function f_i by applying SVILP to set D_i */
$f_i : D_i \rightarrow \{1, -1\}$
/* Reduce the size of set D_i by removing the examples belonging to class p */
$D_{i+1} = D_i \setminus D_p$
end for
return f_i for $i = 1, \ldots, r - 1$

We now describe how the underlying structure of the list is constructed. The method is dynamic and adaptive to learning process. At each node the selection of the positive class is made in such a way so as the classifier can have high generalization ability. The method is presented as Algorithm 2. For each class j a binary class problem is formulated by assigning label '1' to examples of the class j and '-1' to examples of remaining classes. The classifier, induced from the dataset, is evaluated on a validation set and its performance is measured and stored in a list. The process of inducing the classifiers and recording their performances in a list is repeated for all the r classes. Finally the list is sorted and this ranked list defines the underlying structure. In order to measure the performance of the underlying binary classifier we define the expression given by,

$$W_P * P^- + W_N * N^+$$

Here P denotes the number of positive example, and N represent number of negative examples. Similarly, the number of positive examples that are misclassified are represented by P^-, where N^+ shows the number of negative examples that are classified positive. W_P and W_N are the weights assigned to P^-, and N^+ respectively. The weights are assigned to give equal importance to all the classes in a dataset that is characterized by uneven class distribution. We select the weights by using a heuristic and set W_P to $\frac{N}{P}$ where W_N is set to 1.

5. EXPERIMENTS AND RESULTS

We conducted experiments to evaluate the performance of the proposed method to solving multi-class protein fold recognition problem.

We used accuracy as evaluation measures. Let P_j denote the number of examples belonging to class j, $P = \sum_{j=1}^{j=k} P_j$ represent total number of examples belonging to k classes, and TP_j denote the number of correctly classified examples belonging to class j. The accuracy for each class j is given by $\frac{TP_j}{P_j}$ whereas the overall accuracy is given by the expression $\frac{\sum_{j=1}^{j=k} TP_j}{P}$.

Algorithm 2 Learning underlying structure for DL_SVILP

Input: Training set, $d_1, d_2, \ldots, d_n$, validation set, $d'_1, d'_2, \ldots, d'_s$, r classes and a large margin first order rule learner (for example SVILP)
for $j = 1$ to r **do**
/* Formulate the binary class problem by assigning label '1' to examples of class j and '-1' to examples of remaining classes */
/* Induce a binary classification function by applying SVILP to training data, $d_1, d_2, \ldots, d_n$ */
/* Apply the learned function to validation set, $d'_1, d'_2, \ldots, d'_s$ */
/* Measure performance of classifier using expression given below */
$S[j]' = W_P * P^- + W_N * N^+$
where P = total number of positive example, N = total number of negative examples, P^- = number of misclassified positive examples, N^- = number of misclassified negative examples, $W_P = \frac{N}{P}$ and $W_N = 1$
$index[j]' = j$
end for
/* Sort list S' in ascending order and reorder list $index'$ accordingly */
$S = sort(S')$
$index = reorder(index')$
return *index* and S

Table 1: Class distribution for 20 protein folds.

Fold	#Exm	Fold	#Exm
α		α/β	
1	30	11	55
2	14	12	21
3	13	13	14
4	10	14	13
5	10	15	12
β		$\alpha + \beta$	
6	45	16	26
7	21	17	13
8	20	18	13
9	16	19	12
10	14	20	9

We solved protein fold classification problem by applying the proposed method to the dataset presented in [13]. In order to compare the performance of SVILP based multi-class classification scheme with non-SVILP based methods we used multi-class SVM (MC_SVM) and MC_ILP. MC_SVM was trained by using SVM^{light} [4] where the method was presented in [2]. For MC_SVM, we represented protein domains by using non-relational features namely, total number of residues, α-helices and β-strands. Previous research demonstrated the effectiveness of these features for protein fold classification task. For MC_ILP and SVILP based techniques we used relational fold discriminatory features described in [13]. These features are polypeptide chain length, number of α-helices and β-strands, adjacent secondary structure elements, properties of the secondary structure such as the hydrophobicity, the hydrophobic moment, the length of

Table 2: 5-fold cross-validated over all accuracy (OA) ± standard deviation for protein fold dataset for MC_ILP, DL_SVILP and MC_SVM. We also report cross-validated accuracy ± standard deviation for 20 folds.

Fold	MC_ILP	DL_SVILP	MC_SVM
α			
1	43.3 ± 9.0	**73.3 ± 8.1**	43.3 ± 9.0
2	**28.6 ± 12.1**	21.4 ± 11.0	14.3 ± 9.4
3	46.2 ± 13.8	**61.5 ± 13.5**	53.8 ± 13.8
4	10.0 ± 9.5	**40.0 ± 15.5**	0.0 ± 0.0
5	40.00 ± 15.5	40.00 ± 15.5	20.0 ± 12.6
OA	36.4 ± 5.5	**53.3 ± 5.7**	31.2 ± 5.3
β			
6	73.3 ± 6.6	**91.1 ± 4.2**	71.1 ± 6.8
7	57.1 ± 10.8	**95.2 ± 4.7**	66.7 ± 10.3
8	0.0 ± 0.0	15.00 ± 8.0	15.0 ± 8.0
9	43.8 ± 12.4	**75.0 ± 10.8**	68.8 ± 12.8
10	64.3 ± 12.8	**71.4 ± 12.1**	64.3 ± 12.8
OA	52.6 ± 4.6	**74.1 ± 4.1**	59.5 ± 4.6
α/β			
11	**85.5 ± 4.8**	67.3 ± 6.3	58.2 ± 6.7
12	52.4 ± 10.9	**76.2 ± 9.3**	28.6 ± 9.9
13	28.6 ± 12.1	**50.0 ± 13.4**	7.1 ± 6.9
14	7.7 ± 7.4	**30.8 ± 12.8**	0.0 ± 0.0
15	0.0 ± 0.0	8.3 ± 8.0	**16.7 ± 10.8**
OA	54.8 ± 4.6	**56.5 ± 4.6**	35.7 ± 4.5
$\alpha+\beta$			
16	53.8 ± 9.8	**69.2 ± 9.1**	23.1 ± 8.3
17	15.4 ± 10.0	**53.9 ± 13.8**	30.8 ± 12.8
18	7.7 ± 7.4	**46.2 ± 13.8**	30.8 ± 12.8
19	0.0 ± 0.0	8.3 ± 8.0	**25.0 ± 12.5**
20	**77.8 ± 13.9**	66.7 ± 15.7	22.2 ± 13.9
OA	32.9 ± 5.8	**52.1 ± 5.8**	26.0 ± 5.6
OA	46.2 ± 2.6	**60.4 ± 2.5**	40.2 ± 2.5

proline (number of proline residues) and the length of the loop. In order to construct underlying binary SVILP classifiers we used CProgol5 (PROGOL) [7] and SVM^{light}. For MC_ILP theories for were obtained by using CProgol5.

The dataset comprises 381 protein domains. They belong to 20 folds of SCOP that have been categorized into 4 structural classes, namely α, β, α/β and $\alpha+\beta$. Table 1 shows the class distribution for 20 protein folds. The indices 1 to 20 shown in Tables 2 and 1 represent SCOP folds DNA 3-helical, EF hand-like, Globin-like, 4-Helical cytokines, Lambda repressor, Ig beta-sandwich, Tryp ser proteases, OB-fold, SH3-like barrel, Lipocalins, α/β (TIM)-barrel, Rossmann-fold, P-loop, Periplasmic II, α/β-Hydrolases, Ferredoxin-like, Zincin-like, SH2-like, β-Grasp, and Interleukin respectively. The dataset is characterized by uneven class distribution as shown in table 1.

We randomly divided the dataset into 5 equal-sized folds and applied the experimental methodology as follows. At each cross-validation round 3-folds were used for training the classifiers where the remaining two folds were used as validation set and test set. The free parameter of SVM_MC (C, width of the Gaussian kernel) and SVILP_DL (C, width of the Gaussian kernel) were tuned by using the validation set.

Table 2 lists the cross-validated accuracy for each protein fold for multi-class classification methods. Overall accuracy over 20 folds is also given. Table 2 shows that the accuracy values of DL_SVILP are significantly higher than the other methods and it outperforms MC_SVM and MC_ILP.

6. CONCLUSION

In this paper we presented a novel logic based multi-class classification method, DL_SVILP. It produced an accurate solution to a complex bioinformatics problem, namely multi-class protein fold recognition. Experimental results showed that DL_SVILP captured structural and relational similarities between proteins. It accurately assigned protein domains to folds and outperformed all the other methods in the study.

Acknowledgements

The research presented in the paper is supported by Biotechnology and Biological Sciences Research Council (BBSRC) grant with reference number BB/E000940/1.

7. REFERENCES

[1] K. Chen and L. Kurgan. PFRES: Protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics*, 23:2843–2850, 2007.

[2] K. Crammer and Y. Singer. On the algorithmic implementations of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, (2):265–292, 2001.

[3] C. H. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17:349–358, 2001.

[4] T. Joachims. Making large–scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 169–184, Cambridge, MA, 1999. MIT Press.

[5] W. V. Laer, L. de Raedt, and S. Dzeroski. On multi-class problems and discretization in Iductive Logic Programming. In *Proceedings of the 10th International Symposium on Foundations of Intelligent Systems*, pages 277–286, 1997.

[6] N. Landwehr, A. Passerini, L. Raedt, and P. Frasconi. kFOIL: Learning simple relational kernels. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 389–394, 2006.

[7] S. Muggleton. Inverse entailment and progol. *New Generation Computing*, 13:245–286, 1995.

[8] S. Muggleton, H. Lodhi, A. Amini, and M. J. E. Sternberg. Support Vector Inductive Logic Programming. In *Proceedings of the Eighth International Conference on Discovery Science*, volume 3735 of *LNCS(LNAI)*, pages 163–175. Springer Heidelberg, 2005.

[9] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, 247(536–540), 1995.

[10] U. Ruckert and S. Kramer. Margin-base first-order rule learning. *Machine Learning*, 70(2-3):189–206, 2008.

[11] M. Shamim, M. Anwaruddin, and H. A. N. J. Nagarajaram. Support Vector Machine based classification of protein folds using the structural properties of amino acid residue pairs. *Bioinformatocs*, 23(24):3320–3327, 2007.

[12] H. B. Shen and C. K. Chou. Ensemble classifier for protein fold recognition. *Bioinformatics*, 22(14):1717–1722, 2006.

[13] M. Turcotte, S. Muggleton, and J. E. Sternberg. Automated discovery of structural signatures of protein fold and function. *J. Mol. Biol.*, 306:591–605, 2001.