Using Inductive Logic Programming for Predicting Protein-Protein Interactions from Multiple Genomic Data

Tuan Nam Tran, Kenji Satou, and Tu Bao Ho

School of Knowledge Science,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai Nomi Ishikawa 923-1292, Japan
{tt-nam, ken, bao}@jaist.ac.jp

Abstract. Protein-protein interactions play an important role in many fundamental biological processes. Computational approaches for predicting protein-protein interactions are essential to infer the functions of unknown proteins, and to validate the results obtained of experimental methods on protein-protein interactions. We have developed an approach using Inductive Logic Programming (ILP) for protein-protein interaction prediction by exploiting multiple genomic data including protein-protein interaction data, SWISS-PROT database, cell cycle expression data, Gene Ontology, and InterPro database. The proposed approach demonstrates a promising result in terms of obtaining high sensitivity/specificity and comprehensible rules that are useful for predicting novel protein-protein interactions. We have also applied our method to a number of protein-protein interaction data, demonstrating an improvement on the expression profile reliability (EPR) index.

1 Introduction

The interaction between proteins is fundamental to a broad spectrum of biological functions, including regulation of metabolic pathways, immunologic recognition, DNA replication, progression through the cell cycle, and protein synthesis. Therefore, mapping the organism-wide protein-protein interaction network plays an important role in functional inference of the unknown proteins. With the development of genomic technology, new experimental methods have vastly increased the number of protein-protein interactions for various organisms. An enormous amount of protein-protein interaction data have been obtained recently for yeast and other organisms using high-throughput experimental approaches such as yeast two-hybrid [12], affinity purification and mass spectrometry [2], phage display [22]. However, a potential difficulty with these kinds of data is a prevalence of false positive (interactions that are seen in an experiment but never occur in the cell or are not physiologically relevant) and false negatives (interactions that are not detected but do occur in the cell). As such, the prediction of protein-protein interactions using computational approaches can

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 321–330, 2005.

be used to validate the results of high-throughput interaction screens and used to complement the experimental approaches.

There have been a number of studies using computational approaches applied to predicting interactions. Bock and Gough [3] applied a Support Vector Machine learning system to predict directly protein-protein interactions from primary structure and associated data. Jansen *et al.* [13] used a Bayesian networks approach for integrating weakly predictive genomic features into reliable predictions of protein-protein interactions. A different approach is based on interacting domain pairs, attempting to understand protein-protein interactions at the domain level. Sprinzak and Margalit [23] proposed the AM (Association Method) for computing the score for each domain pair. Deng *et al.* [9] estimated the probabilities of interactions between every pair of domains using an EM algorithm, using the inferred domain-domain interactions to predict interactions between proteins. The major drawback of this approach is that there are currently no efficient experimental methods for detecting domain-domain interactions. Also, in [11], Grigoriev demonstrated that there is a significant relationship between gene expression and protein interactions on the proteome scale, finding that the mean correlation coefficients of gene expression profiles between interacting proteins are higher than those between random protein pairs.

In this paper, we present an approach for predicting genome-wide protein-protein interactions in yeast using the ILP system Aleph [1], a successor to Progol [16]. Unlike the other work, our approach is able to exploit the relationships among features of multiple genomic data, and to induce rules that give possible insight into the binding mechanism of the protein-protein interactions. Concerning rule-based methods using protein-protein interaction data, Oyama *et al.* [21] applied Association Rule Mining to extracting rules from protein-protein interaction data, however, the goal of this work is descriptive while our aim is to generate rules for predictive purposes.

2 ILP and Bioinformatics

Inductive Logic Programming (ILP) is the area of AI which is built on a foundation laid by research in machine learning and computational logic. ILP deals with the induction of hypothesized predicate definitions from examples and background knowledge. Logic programs are used as a single representation for examples, background knowledge and hypotheses. ILP is differentiated from most other forms of Machine Learning (ML) both by its use of an expressive representation language and its ability to make use of logically encoded background knowledge. This has allowed successful applications of ILP in areas such as molecular biology and natural language which both have rich sources of background knowledge and both benefit from the use of an expressive concept representation languages [17].

It is considered that one of the most important application domains for machine learning in general is bioinformatics. There have been many ILP systems that are successfully applied to various problems in bioinformatics. ILP is partic-

ular suitable for bioinformatics tasks because of its ability to take into account background knowledge and work directly with structured data. The ILP system GOLEM [18] was used to model the structure activity relationships of trimethoprim analogues binding to dihydrofolate reductase [14]. A study of discriminating molecules with positive mutagenicity from those with negative mutagenicity [15] has been conducted using Progol [16], another ILP system. ILP has also been applied to many other tasks in bioinformatics, such as protein secondary structure prediction [19] and protein fold recognition [26].

3 Using ILP for Predicting Protein-Protein Interactions

In this section, we present an algorithm for discovering rules using ILP. We use a multi-relational data mining approach to discover rules from multiple genomic data concerning protein-protein interactions. At present, we are using five kinds of genomic data:

1. **SWISS-PROT** [5], containing description of the function of a protein, its domains structure, post-translational modifications, variants, and so on.
2. **MIPS** [4], containing highly accurate protein interaction data for yeast.

Algorithm 1 Discovering rules for protein-protein interactions

Require:

Set of protein interacting pairs $I = \{(p_i, p_j)\}$, $p_i \in P$, $p_j \in P$, where P is the set of proteins occurred

Number of negative examples N

Multiple genomic data used for extracting background knowledge ($S^{SWISS-PROT}, S^{MIPS}, S^{expression}, S^{GO}, S^{InterPro}$)

Ensure: Set of rules R for protein-protein interaction prediction

1: $R := \emptyset$, $S_{pos} := I$
2: Extract protein annotation information concerning each p of P from $S^{SWISS-PROT}$
3: Extract protein information concerning each p of P from S^{MIPS}
4: Call GENERATE-NEGATIVES for artificially generating N negative examples
5: Extract the expression correlation coefficients from $S^{expression}$ for every protein pairs (p_k, p_l), where $p_k \in P, p_l \in P$.
6: Extract all `is_a` and `part_of` relations (g_1, g_2), $g_1 \in G_P, g_2 \in G_P$, where G_P is the set of GO terms associated with P
7: Extract all relations between InterPro domains and GO terms $(d_{InterPro}, g)$ from $S^{InterPro}$, $d_{InterPro} \in D_P^{InterPro}, g \in G_P$, where $D_P^{InterPro}$ is the set of InterPro domains associated with P
8: Select a positive example at random
9: Saturate it to find the most specific clause that entails this example
10: Do top-down search for selecting the best clause c and add c to R
11: Remove covered positive examples
12: If there remain positive examples, go to step 8
13: **return** R

3. **Gene expression data** [24], containing the correlation of mRNA amounts with temporal profiles during the cell cycle.
4. **Gene Ontology (GO)** [20], containing the relations between GO terms.
5. **InterPro** [7], containing the relations between InterPro domains and their corresponding GO terms.

Our algorithm 1 consists of two main parts. The first part (step 1 to 7) is concerned with generating negative examples and extracting background knowledge from multiple genomic data. The second part (step 8 to 12) deals with inducing rules given the lists of positive, negative examples and background knowledge using Aleph [1]. Aleph is an ILP system that uses a top-down ILP covering algorithm, taking as input background information in the form of predicates, a list of modes declaring how these predicates can be chained together, and a designation of one predicate as the head predicate to be learned. Aleph is able to use a variety of search methods to find good clauses, such as the standard methods of breadth-first search, depth-first search, iterative beam search, as well as heuristic methods requiring an evaluation function. We use the default evaluation function *coverage* (the number of positive and negative examples covered by the clause) in our work.

Algorithm 2 GENERATE-NEGATIVES

Require:
Number of negative examples N and S^{MIPS}
Ensure: Set of negative examples S_{neg} consisting of N protein pairs

1: $n := 0$, $S_{neg} := \emptyset$
2: **repeat**
3: Select an arbitrary pair (p_k, p_l), where $p_k \in P, p_l \in P$
4: Find the sets of subcellular location L_k and L_l of p_k and p_l from S^{MIPS}
5: **if** $L_k \cap L_l = \emptyset$ **then**
6: Add (p_k, p_l) to S_{neg}
7: $n := n + 1$
8: **endif**
9: **until** $n = N$
10: **return** S_{neg}

In this paper, we want to learn the following target predicate

`interact(Protein, Protein)`: the instances of this relation represent the interaction between two proteins.

For background knowledge, we shortly denote all predicates used by each genomic data. Note that Aleph uses *mode declarations* to build the bottom clause, and there are three types of variables: (1) an input variable (+), (2) an output variable (−), and (3) a constant term (#). Table 1 shows the list of predicates used as background knowledge for each genomic data.

4 Experiments

4.1 Data Preparation

We used the core data of the Yeast Interacting Proteins Database provided by Ito [6] as positive examples. Ito *et al.* [12] conducted comprehensive analysis using their system to examine two-hybrid interactions in all possible combinations between the 6000 proteins of the budding yeast *Saccharomyces cerevisiae*. Among 4,549 interactions detected using yeast-hybrid analysis, the "core" data consist of 841 interactions with more than two IST hits[1], accounting for 18.6% of the whole data. Note that the core data used in this paper is a subset of protein-protein interactions of MIPS [4] database, which is considered as the gold-standard for positive examples in [13]. A negatives gold-standard is defined similar to [13] in which negative examples are synthesized from lists of proteins in separate subcellular compartments.

We employ our approach to predict protein-protein interactions. We used the core data of Ito data set [6] mentioned above as positive examples, selecting at random 1000 protein pairs whose elements are in separate subcellular compartments as negative examples. Each interaction in the interaction data originally shows a pair of bait and prey ORF (Open Reading Frame)[2] some of which are not found in SWISS-PROT database. After removing all interactions in which either bait ORF or prey ORF is not found in SWISS-PROT, we obtained 602 interacting pairs from the original 841 pairs.

4.2 Analysis of Sensitivity/Specificity

To validate our proposed method, we conducted a 10-fold cross-validation test, comparing cross-validated sensitivity and specificity with those obtained by using AM [23] and SVM method. The AM method calculates a score d_{kl} to each domain pair (D_k, D_l) as the number of interacting protein pairs containing (D_k, D_l) divided by the number of protein pairs containing (D_k, D_l).

In the approach of predicting protein-protein interactions based on domain-domain interactions, it can be assumed that domain-domain interactions are independent and two proteins interact if at least one domain pairs of these two proteins interact. Therefore, the probability p_{ij} that two proteins P_i and P_j interact can be calculated as

$$p_{ij} = 1 - \prod_{D_k \in P_i, D_l \in P_j} (1 - d_{kl})$$

We implemented the AM and SVM methods in order to compare with our proposed method. We used the PFAM domains extracted from SWISS-PROT and superdomains, i.e. proteins without any domain information. The probability threshold is set to 0.05 for the simplicity of comparison. For SVM method, we

[1] IST hit means how many times the corresponding interaction was observed. The higher IST number, the much more reliable the corresponding interaction is.

[2] ORF is a series of codons which can be translated into a protein.

Table 1. Predicates used as background knowledge in various genomic data

Genomic data	Background Knowledge
SWISS-PROT	`haskw(+Protein,#Keyword)`: A protein contains a keyword
	`hasft(+Protein,#Feature)`: A protein contains a feature
	`ec(+Protein,#EC)`: An enzyme code for a protein
	`pfam(+Protein,-PFAM_Domain)` A protein contains a Pfam domain
	`interpro(+Protein,-InterPro_Domain)` A protein contains a InterPro domain
	`pir(+Protein,-PIR_Domain)` A protein contains a Pir domain
	`prosite(+Protein,-PROSITE_Domain)` A protein contains a Prosite domain
	`go(+Protein,-GO_Term)` A protein contains a GO term
MIPS	`subcellular_location(+Protein,#Subcellular_Structure)` Relation between proteins and the subcellular structures in which they are found.
	`function_category(+Protein,#Function_Category)` A protein which is categorized to a certain function category
	`protein_category(+Protein,#Protein_Category)` A protein which is categorized to a certain protein category
	`phenotype_category(+Protein,#Phenotype_Category)` A protein which is categorized to a certain phenotype category
	`complex_category(+Protein,#Complex_Category)` A protein which is categorized to a certain complex category
Gene expression	`correlation(+Protein,+Protein,-Expression)` Expression correlation coefficient between two proteins
GO	`is_a(+GO_Term,-GO_Term)` `is_a` relation between two GO terms
	`part_of(+GO_Term,-GO_Term)` `part_of` relation between two GO terms
InterPro	`interpro2go(+InterPro_Domain,-GO_Term)` Mapping of InterPro entries to GO

used SVM^{light} [25] for learning, and used the same set of PFAM domains and superdomains as used in AM method. The linear kernel with default value of the parameters was used. For Aleph, we selected $minpos = 2$ and $noise = 0$, i.e. the lower bound on the number of positive examples to be covered by an acceptable clause is 2, and there are no negative examples allowed to be covered by an acceptable clause. We also used the default evaluation function *coverage* which is defined as $P - N$, where P, N are the number of positive and negative examples covered by the clause.

Table 2 shows the performance of Aleph compared with AM and SVM methods. The sensitivity of a test is described as the proportion of true positives it detects of all the positives, measuring how accurately it identifies positives. On the other hand, the specificity of a test is the proportion of true negatives it detects of all the negatives, thus is a measure of how accurately it identifies negatives. It can be seen from this Table that the proposed method showed a considerably high sensitivity and specificity given a certain number of negative examples. The number of negative examples should be chosen neither too large nor too small to avoid the imbalanced learning problem. At present, we did not compare our approach with EM method [9] in which they obtained 42.5% specificity and 77.6% sensitivity using the combined Uetz and Ito protein-protein interaction data.

Table 2. Performance of Aleph compared with AM and SVM methods. The sensitivity and specificity are obtained for each randomly chosen set of negative examples. The last column demonstrates the number of rules obtained using our proposed method with the minimum positive cover is set to 2.

# Neg	Sensitivity			Specificity			# Rules
	AM	SVM	Aleph	AM	SVM	Aleph	
100	0.70	**0.99**	0.90	**0.46**	0.01	0.44	27
500	0.68	0.54	**0.79**	0.42	0.61	**0.84**	63
1000	0.71	0.32	**0.73**	0.39	0.88	**0.93**	62
2000	**0.69**	0.26	**0.69**	0.38	0.95	**0.96**	58
4000	**0.69**	0.15	0.68	0.39	0.98	**0.99**	68

4.3 Rule Analysis

Figure 1 demonstrates a number of selective rules obtained when providing 602 positive examples and 1000 randomly chosen negative examples. Those rules are manually ranked using the difference between positive and negative coverages. It can be seen that although some of rules can be obtained using other propositional learning methods, some other rules can only be obtained using ILP. Rule 1 supports the approach using domain-domain interactions, demonstrating that two proteins interact if they share a common PFAM domain (81 cases covered among a total of 602 positive examples). Some rules obtained also match the result reported in [11] that the mean correlation coefficients of gene expression profiles between interacting proteins are higher than those between random protein pairs.

Using the Gene Ontology Term Finder tool [10], we also searched for significant GO terms, or parents of the GO terms used to describe the pair of protein interaction of each positive example covered by those rules in Figure 1. As a result, it can be found that rule 5, 6, 10, 12, 13, 14 are relevant with very high confidence, rule 7, 8, 9, 11 are relevant with lower confidence, and rule 15 is irrelevant.

4.4 Assessment of the Reliability Using EPR Index

Since high-throughput experimental methods may produce false positives, it is essential to assess the reliability of protein-protein interaction data obtained. Deane et al. [8] proposed the expression profile reliability (EPR) index to assess the reliability of measurement of protein interaction. The EPR index estimates the biologically relevant fraction of protein interactions detected in a high throughput screen. For each given data, we retrieved all protein pairs that classified as positive. Table 3 shows the EPR index calculated using the original and our proposed method for a number of well-known protein-protein interaction data. It can be seen that the EPR index of our method is higher than the original one, demonstrating the validity of the proposed method.

Rule 1 [Pos cover = 81 Neg cover = 0]
$interact(A, B) :- \quad pfam(B, C), pfam(A, C).$
Rule 2 [Pos cover = 61 Neg cover = 0]
$interact(A, B) :- \quad go(B, C), go(A, C), is\_a(C, D).$
Rule 3 [Pos cover = 51 Neg cover = 0]
$interact(A, B) :- \quad interpro(B, C), interpro(A, C), interpro2go(C, D).$
Rule 4 [Pos cover = 15 Neg cover = 0]
$interact(A, B) :- \quad go(B, C), go(A, C),$
$hasft(A, domain\_coiled\_coil\_potential).$
Rule 5 [Pos cover = 8 Neg cover = 0]
$interact(A, B) :- \quad go(B, C), go(A, C),$
$complex\_category(A, intracellular\_transport\_complexes).$
Rule 6 [Pos cover = 6 Neg cover = 0]
$interact(A, B) :- \quad subcellular\_location(B, nucleus),$
$function\_category(A, cell\_cycle\_and\_dna\_processing),$
$phenotype\_category(B, cell\_morphology\_and\_organelle\_mutants).$
Rule 7 [Pos cover = 6 Neg cover = 0]
$interact(A, B) :- \quad pfam(A, C), subcellular\_location(B, er),$
$haskw(B, autophagy).$
Rule 8 [Pos cover = 5 Neg cover = 0]
$interact(A, B) :- \quad phenotype\_category(B, conditional\_phenotypes),$
$hasft(A, domain\_rna\_binding\_rrm).$
Rule 9 [Pos cover = 5 Neg cover = 0]
$interact(A, B) :- \quad correlation(B, A, C), gteq(C, 0.241974),$
$hasft(A, domain\_rna\_binding\_rrm).$
Rule 10 [Pos cover = 4 Neg cover = 0]
$interact(A, B) :- \quad pfam(A, C), haskw(B, direct\_protein\_sequencing),$
$hasft(B, domain\_histone\_fold).$
Rule 11 [Pos cover = 4 Neg cover = 0]
$interact(A, B) :- \quad correlation(A, B, C), gteq(C, 0.236007),$
$hasft(A, domain\_poly\_gln).$
Rule 12 [Pos cover = 4 Neg cover = 0]
$interact(A, B) :- \quad protein\_category(A, gtp - binding\_proteins),$
$correlation(A, B, C), gteq(C, 0.144137).$
Rule 13 [Pos cover = 4 Neg cover = 0]
$interact(A, B) :- \quad function\_category(B, cell\_fate),$
$hasft(B, transmem\_potential), hasft(A, transmem\_potential).$
Rule 14 [Pos cover = 3 Neg cover = 0]
$interact(A, B) :- \quad subcellular\_location(B, integral\_membrane),$
$correlation(A, B, C), gteq(C, 0.46332).$
Rule 15 [Pos cover = 2 Neg cover = 0]
$interact(A, B) :- \quad correlation(B, A, C), gteq(C, 0.599716),$
$haskw(A, cell\_division).$

Fig. 1. Some rules obtained with $minpos = 2$. For example, rule 14 means that protein A will interact with protein B if protein B is located in the integral membrane of the cell, and the expression correlation coefficient between protein A and protein B is greater than 0.46332.

Table 3. Evaluated the proposed method using EPR index. The number of interactions after preprocessing means the number of interactions obtained after removing all interactions in which either bait ORF or prey ORF it not found in SWISS-PROT.

Data	Number of interactions			EPR index	
	Original	After preprocessing	Proposed	Original	Proposed
Ito	4549	3174	1925	0.1910 ± 0.0306	**0.2900** ± 0.0481
Uetz	1474	1109	738	0.4450 ± 0.0588	**0.5290** ± 0.0860
Ito+Uetz	5827	4126	2567	0.2380 ± 0.0287	**0.3170** ± 0.0431
MIPS	14146	10894	7080	0.5950 ± 0.0337	**0.6870** ± 0.0420
DIP	15409	12152	8674	0.4180 ± 0.0260	**0.5830** ± 0.0374

5 Conclusions and Future Work

We have presented an approach using ILP to predict protein-protein interactions. The experimental results demonstrate that our proposed method can produce comprehensible rules, and at the same time, showing a considerably high performance compared with other work on protein-protein interaction prediction. In future work, we would like to investigate further about the biological significance of novel protein-protein interactions obtained by our method, and apply the ILP approach to other important tasks, such as predicting protein functions and subcellular locations using protein-protein interaction data. We are also investigating to exploit the GO structures as background knowledge, rather than using the occurence of a single GO term as in the current work.

Acknowledgements

This research is supported by the grant-in-aid for scientific research on priority areas (C) "Genome Information Science" from the Japanese Ministry of Education, Culture, Sports, Science and Technology. The authors would like to thank JST BIRD (Institute for Bioinformatics Research and Development) for all the support during the period of this work.

References

1. Aleph A. Srinivasan. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph_toc.html.
2. A. Bauer and B. Kuster. Affinity purification-mass spectrometry: Powerful tools for the characterization of protein complexes. *Eur. J. Biochem.*, 270(4):570–578, 2003.
3. J. R. Bock and D. A. Gough. Predicting protein-protein interactions from primary structure. *Bioinformatics*, 17(5):455–460, 2001.
4. Comprehensive Yeast Genome Database. http://mips.gsf.de/genre/proj/yeast/index.jsp.

5. SWISS-PROT database. http://www.expasy.ch/sprot.
6. Yeast Interacting Proteins Database. http://genome.c.kanazawa-u.ac.jp/Y2H/.
7. InterPro database concerning protein families and domains. http://www.ebi.ac.uk/interpro/.
8. C. M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg. Protein interactions: Two methods for assessment of the reliability of high-throughput observations. *Mol. Cell. Prot.*, 1:349–356, 2002.
9. M. Deng, S. Mehta, F. Sun, and T. Chen. Inferring domain-domain interactions from protein-protein interactions. *Genome Res.*, 12(10):1540–1548, 2002.
10. SGD Gene Ontology Term Finder. http://db.yeastgenome.org/cgi-bin/GO/goTermFinder.
11. A. Grigoriev. A relationship between gene expression and protein interactions on the proteome scale: analysis of the bacteriophage t7 and the yeast *saccharomyces cerevisiae*. *Nucleic Acids Res.*, 29(17):3513–3519, 2001.
12. T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. In *Proc. Natl. Acad. Sci. USA 98*, pages 4569–4574, 2001.
13. R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, 2003.
14. R. King, S. Muggleton, R. A. Lewis, and M. J. Sternberg. Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. In *Proc. Natl. Acad. Sci.*, pages 11322–11326, 1992.
15. R. King, S. Muggleton, A. Srinivasan, and M. J. Sternberg. Structure-activity relationships derived by machine learning: the use of atoms and their bond connectives to predict mutagenicity by inductive logic programming. In *Proc. Natl. Acad. Sci.*, pages 438–442, 1996.
16. S. Muggleton. Inverse entailment and progol. *New Generation Computing*, 13:245–286, 1995.
17. S. Muggleton. Inductive logic programming: Issues, results and the challenge of learning language in logic. *Artificial Intelligence*, 114:283–296, 1999.
18. S. Muggleton and C. Feng. Efficient induction of logic programs. In *Proceedings of the First Conference on Algorithmic Learning Theory*, 1990.
19. S. Muggleton, R. King, and M. Sternberg. Protein secondary structure prediction using logic-based machine learning. *Protein Engineering*, 5(7):647–657, 1992.
20. Gene Ontology. http://www.geneontology.org/.
21. T. Oyama, K. Kitano, K. Satou, and T. Ito. Extracting of knowledge on protein-protein interaction by association rule discovery. *Bioinformatics*, 18(5):705–714, 2002.
22. G. P. Smith. Filamentous fusion phage: Novel expression vectors that display cloned antigens on the vision surface. *Science*, 228(4705):1315–1317, 1985.
23. E. Sprinzak and H. Margalit. Correlated sequence-signatures as markets of protein-protein interaction. *J. Mol. Biol.*, 311:681–692, 2001.
24. Yale Gerstein Lab Supplementary data. http://networks.gersteinlab.org/genome/intint/supplementary.htm.
25. SVM^{light} T. Joachim. http://svmlight.joachims.org.
26. M. Turcotte, S. Muggleton, and M. J. Sternberg. Protein fold recognition. In *International Workshop on Inductive Logic Programming (ILP-98), C. D. Page (Ed.)*, 1998.