Prediction of domain-domain interactions using inductive logic programming from multiple genome databases

Thanh Phuong Nguyen and Tu Bao Ho

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

Abstract. Protein domains are the building blocks of proteins, and their interactions are crucial in forming stable protein-protein interactions (PPI) and take part in many cellular processes and biochemical events. Prediction of protein domain-domain interactions (DDI) is an emerging problem in computational biology. Different from early works on DDI prediction, which exploit only a single protein database, we introduce in this paper an integrative approach to DDI prediction that exploits multiple genome databases using inductive logic programming (ILP). The main contribution to biomedical knowledge discovery of this work are a newly generated database of more than 100,000 ground facts of the twenty predicates on protein domains, and various DDI findings that are evaluated to be significant. Experimental results show that ILP is more appropriate to this learning problem than several other methods. Also, many predictive rules associated with domain sites, conserved motifs, protein functions and biological pathways were found.

1 Introduction

Understanding functions of proteins is a main task in molecular biology. Early work in computational biology has focused on finding protein functions via prediction of protein structures, e.g., [13]. Recently, detecting protein functions via prediction of protein-protein interactions (PPI) has emerged as a new trend in computational biology, e.g., [2], [10], [24].

Within a protein, a domain is a fundamental structural unit that is self-stabilizing and often folds independently of the rest of the protein chain. Domains often are named and singled out because they figure prominently in the biological function of the protein they belong to; for example, the *calcium-bindingdomain* of *calmodulin*. The domains form the structural or functional units of proteins that partake in intermolecular interactions. Therefore, domain-domain interaction (DDI) problem has biological significance in understanding protein-protein interactions in deepth.

Concerning protein domains, a number of domain-based approaches to predict PPIs have recently been proposed. One of the pioneering works is an association method developed by Sprinzak and Margalit [23]. Kim *et al.* improved

the association method by considering the number of domains in each protein [10]. Han *et al.* proposed a domain combination-based method by considering the possibility of domain combinations appearing in both interacting and non-interacting sets of protein pairs [8]. A graph-oriented approach is proposed by Wojcik and Schachter called the 'interacting domain profile pairs' (IDPP) approach [26]. That method uses a combination of sequence similarity search and clustering based on interaction patterns. Therefore, the only purpose of the above mentioned work was to predict and/or to validate protein interactions. They all confirmed the biological role of DDIs in PPIs, however, they did not much take domain-domain interactions into account.

Recently, there are several works that not only use protein domains to predict protein interactions, but also attempt to discover DDIs. An integrative approach is proposed by Ng *et al.* to infer putative domain-domain interactions from three data sources, including experimentally derived protein interactions, protein complexes and Rosetta stone sequences [15]. The interaction scores for domain pairs in these data sources were obtained with a calculation scheme similar to the association method by considering frequency of each domain among the interacting protein pairs. The maximum likelihood estimation (MLE) is applied to infer domain interactions by maximizing the likelihood of the observed protein interaction data [5]. The probabilities of interaction between two domains (only single-domain pairs are considered) are optimized using the expectation maximization (EM) algorithm. Chen et al. used domain-based random forest framework to predict PPIs [2]. In fact, they used the PPI data from DIP and a random forest of decision trees to classify protein pairs into sets of interacting and non-interacting pairs. Following the branches of trees, they found a number of DDIs. Riley et al. proposed a domain pair exclusion analysis (DPEA) for inferring DDIs from databases of protein interactions [22]. DPEA features a log odds score, E_{ij}, reflecting confidence that domains i and j interact.

The above mentioned works mostly use protein interaction data to infer DDIs, and all of them have two limitations. First, they used only the protein information (particularly protein-protein interaction data) or the co-occurrence of domains in proteins, and ignored other domain-domain interaction information between the protein pairs. However, DDIs also depend on other features of proteins and domains as well–not only protein interactions [11], [25]. Second, each of them usually exploited only a single protein database and none of the single protein databases can provide all information needed to do better DDI prediction.

In this paper, we present an approach using ILP and multiple genome databases to predict domain-domain interactions. The key idea of our computational method of DDI prediction is to exploit as much as possible background knowledge from various databases of proteins and domains for inferring DDIs. Sharing some common points in ILP framework in bioinformatics with [24], this paper concentrates on discovering knowledge of domain-domain interactions. To this end, we first examine seven most informative genome databases, and extract more than a hundred thousand possible and necessary ground facts on protein domains. We

then employ inductive logic programming (ILP) to infer efficiently DDIs. We carry out a comparative evaluation of findings for DDIs and learning methods in terms of sensitivity and specificity. By analyzing various produced rules, we found many interesting relations between DDIs and protein functions, biological pathways, conserved motifs and pattern sites.

The remainder of the paper is organized as follows. In Section 2, we present our proposed methods to predict DDIs using ILP and multiple genome databases. Then the evaluation is given in Section 3. Finally, some concluding remarks are given in Section 4.

2 Method

In this section, we describe our proposed method to predict domain-domain interactions from multiple genome databases. Two main tasks of the method are: (1) Generating background knowledge[1] from multiple genome databases and (2) Learning DDI predictive rules by ILP from generated domain and protein data.

We first describe these tasks in the next section, Section 2.1, then present our proposed framework using ILP to exploit extracted background knowledge for DDI prediction (Section 2.2).

2.1 Generating background knowledge from multiple genome databases

Unlike previous work mentioned in Section 1, we chose and extracted data from seven genome databases to generate background knowledge with an abundant number of ground facts and used them to predict DDI. Figure 1 briefly presents these seven databases.

Fig. 1 Description of genome databases used

1. **Pfam** [6]: Pfam contains a large collection of multiple sequence alignments and profile hidden Markov models (HMM) covering the majority of protein domains.
2. **PRINTS** [7]: A compendium of protein fingerprints database. Its diagnostic power is refined by iterative scanning of a SWISS-PROT/TrEMBL composite.
3. **PROSITE** [18]: Database of protein families and domains. It consists of biologically significant sites, patterns and profiles.
4. **InterPro** [4]: InterPro is a database of protein families, domains and functional sites in which identifiable features found in known proteins can be applied to unknown protein sequences.
5. **Uniprot** [21]: UniProt (Universal Protein Resource) is the world's most comprehensive catalog of information on proteins which, consists of protein sequence and function data created by combining the information in Swiss-Prot, TrEMBL, and PIR.
6. **MIPS** [3]: The MIPS Mammalian Protein-Protein Interaction Database is a collection of manually curated high-quality PPI data collected from the scientific literature by expert curators.
7. **Gene Ontology (GO)** [19]: The three organizing principles of GO are molecular function, biological process and cellular component. This database contains the relations between GO terms.

[1] the term 'background knowledge' is used here in terms of language of inductive logic programming.

We integrated domain data and protein data from seven genome databases: four domain databases (Pfam database, PROSITE database, PRINTS database, and InterPro database) and three protein databases from UniProt database, MIPS database, Gene Ontology.

Extract domain and protein data from multiple genome databases. The first issue faced is, what kinds of genome databases are suitable for DDI prediction. When choosing data, we are concerned on two points. First is biological role of that data in domain-domain interaction, and second is the availability of that data.

Denote by D the set of all considered protein domains, d_i a domain in D, p_k a protein that consists of some domains d_is, and P the set of such proteins. A domain pair (d_i, d_j) that interacts with each other is denoted by d_{ij}, otherwise by $\neg d_{ij}$. In fact, whether two domains d_i and d_j interact depends on: (i) the domain features of d_i and d_j and, (ii) the protein features of some proteins p_ks consisting of d_i and d_j [25]. Denote by df_t^m a domain feature tth extracted from the domain database M. With different domains, one feature df_t^m may have different values. For example, the domain site and pattern feature extracted from PROSITE database have some values like *Casein kinase II phosphorylation site* or *Anaphylatoxin domain signature*. Denote by pf_r^l a protein feature rth extracted from protein database L. Also in different domains, one protein feature pf_r^l may have different values. For example, GO term feature extracted from GO database have some values like *go0006470* or *go0006412*. The extracted domain/protein features are mentioned as biologically significant factors in domain-domain interactions [25], [11], [20], etc. The combination of both domain features and protein features constructed the considerable background knowledge associated with DDIs.

Algorithm 1 shows how to extract data (values) of domain features df_t^ms and protein features pf_r^ls for all domains $d_i s \in D$ from multiple data sources. Pfam domain accessions are domain identifiers and ORF (open reading frame) names are protein identifiers. We know that one protein can have many domains and one domain can belong to many proteins. Then, each protein identifier is mapped with the identifiers of its own domains. As the result, protein feature values are assigned to domains.

This paper concentrates on predicting DDIs for *Saccharomyces cerevisiae* – a budding yeast, as the *Saccharomyces cerevisiae* database is available. To map proteins and their own domains, the interacting proteins in DIP database [17], well-known yeast PPI database, are selected. If one protein has no domain, features of that protein are not predictive for domain-domain interactions. If one domain does not belong to any interacting proteins in DIP database, it seems not to have any chance to interact with others. Thus, we excluded all proteins and domains which did not have matching partners (Step 5, Step 6). Having extracted interacting proteins from DIP database, mapping data are more reliable and meaningful. After mapping proteins and their domains, the values of all domain/protein features are extracted (from Step 8 to Step 12).

Algorithm 1 Extracting protein and domain data from multiple sources

Input:

Set of domains $D \supset \{d_i\}$.

Multiple genomic data used for extracting background knowledge $(S^{Pfam}, S^{InterPro}, S^{PROSITE}, S^{PRINTS} S^{Uniprot}, S^{MIPS}, S^{GO})$.

Output:

Set of domain feature values $Feature^{domain}$.

Set of protein features values $Feature^{protein}$.

1: $Feature^{domain} := \emptyset$; $Feature^{protein} := \emptyset$; $P := \emptyset$.
2: Extract all interacting proteins p_ks from DIP database; $P := P \cup \{p_k\}$.
3: **for all** proteins p_ks $\in P$ and domains d_is $\in D$.
4: Mapping proteins p_ks with their own domains d_is by the protein identifiers and the domain identifiers.
5: **if** a domain d_i does not belong to any protein p_k **then** $D := D \backslash \{d_i\}$.
6: **if** a protein p_k does not consist of any domain d_i **then** $P := P \backslash \{p_k\}$.
7: **for each** $d_i \in D$
8: Extract all values $df_t^m.value$s for domain feature d_t^m from domain database M $(\forall M \in (S^{Pfam}, S^{InterPro}, S^{PROSITE}, S^{PRINTS}))$.
9: **if** $df_i^m.value \notin Feature^{domain}$ **then** $Feature^{domain} := Feature^{domain} \cup \{d_t^m\}.value$.
10: Extract all values $pf_r^m.value$ of protein feature pf_r^l from protein database L $(\forall L \in S^{Uniprot}, S^{MIPS}, S^{GO}))$.
11: **if** $pf_r^m.value \notin Feature^{domain}$ **then** $Feature^{protein} := Feature^{protein} \cup \{pf_r^l\}.value$.
12: **return** $Feature^{domain}, Feature^{protein}$.

Generating background knowledge. The data which we extracted from seven databases have different structures: numerical data (for example, the number of motif), text data (for example, protein function category), mixture of numerical and text data (for example, protein keywords, domain sites). The extracted data (the values of all domain/protein features) are represented in form of predicates.

Aleph system [1] is applied to induce rules. Note that Aleph uses *mode declarations* to build the bottom clauses, and a simple mode type is one of : (1) the input variable (+), (2) the output variable (−), and (3) the constant term (#). In this paper, target predicate is `interact_domain(domain, domain)`. The instances of this relation represent the interaction between two domains. For background knowledge, all domain/protein data are shortly denoted in form of different predicates. Table 1 shows the list of predicates used as background knowledge for each genomic data. With the twenty background predicates, we obtained totally 100,421 ground facts associated with DDI prediction.

Extracted domain features (from four databases: Pfam, InterPro, PROSITE and PRINTS) are represented in the form of predicates. These predicates describe domain structures, domain characteristics, domain functions and protein-domain relations. Among them, there are some predicates which are the relations between accession numbers of two databases, for example, *prints(pf00393,pr00076)*.

Data from different databases are bound by these predicates. In PRINTS database,

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

Genomic data	Background knowledge predicates	#Ground fact
Pfam	`prosite(+Domain,-PROSITE_Domain)` A domain has a PROSITE annotation number	1804
	`interpro(+Domain,-InterPro_Domain)` A domain has an InterPro annotation number	2804
	`prints(+Domain,-PRINTS_Domain)` A domain has a PRINTS annotation number	1698
	`go(+Domain,-GO_Term)` A domain has a GO term	2540
InterPro	`interpro2go(+InterPro_Domain,-GO_Term)` Mapping of InterPro entries to GO	2378
PROSITE	`prosite_site(+Domain,#prosite_site)` A domain contains PROSITE significant sites or motifs	2804
PRINTS	`motif_compound(+Domain,#motif_compound)` A domain is compounded by number of conserved motifs	3080
Uniprot	`haskw(+Domain,#Keyword)` A domain has proteins keywords	13164
	`hasft(+Domain,#Feature)` A domain has protein features	8271
	`ec(+Domain,#EC)` A domain has coded enzyme of its protein	2759
	`pir(+Domain,-PIR_Domain)` A domain has a Pir annotation number	3699
MIPS	`subcellular_location(+Protein,#Subcellular_Structure)` A domain has subcellular structures in which its protein is found.	10638
	`function_category(+Domain,#Function_Category)` A domain has the protein categorized to a certain function category	11975
	`domain_category(+Domain,#Domain_Category)` A domain has proteins categorized to a certain protein category	5323
	`phenotype_category(+Domain,#Phenotype_Category)` A domain has proteins categorized to a certain phenotype category	8066
	`complex_category(+Domain,#Complex_Category)` A domain has proteins categorized to a certain complex category	7432
GO	`is_a(+GO_Term,-GO_Term)` `is_a` relation between two GO terms	1009
	`part_of(+GO_Term,-GO_Term)` `part_of` relation between two GO terms	1207
Others	`num_int(+Domain,#num_int)` A domain has a number of domain-domain interactions	804
	`ig(+Domain,+ Domain, #ig)` Interaction generality is the number of domains that interact with just two considered domains	8246
Totals		**100,421**

motif_compound information gives the number of conserved motifs found in proteins and domains. The number of motifs is important in understanding the conservation of protein/domain structures in the evolutionary process [7]. We generated predicate `motif_compound(+Domain,#motif_compound)`. This predicate is predictive for DDI prediction and gives information about the stability of DDIs (example rules are shown and analyzed in Section 3.2). For example: $motif\_compound(pr00517, compound(8))$, where *pr00517* is the accession numbers in PRINTS database and *compound(8)* is the number of motifs.

Protein domains are the basic elements of proteins. Protein features have a significant effect on domain-domain interactions. These protein features (extracted from three databases Uniprot, MIPS and GO) are showed in the form of predicates. These predicates describe function categories, subcellular locations,

GO terms, etc. They give the relations between DDIs and promising protein features. For example, most interacting proteins are in the same complexes [14]. The domains of these interacting proteins can interact with each other. As a result, if some domains belong to the some proteins categorized in the same complex, they can be predicted to have some domain-domain interactions. The predicate `complex_category(+Domain,#Complex_Category)` means that a domain has proteins categorized to a certain complex category. For example, $complex\_category(pf00400, transcription\_complexes)$, where *pf00400* is Pfam accession number and *transcription_complexes* is complex category name.

2.2 Learning DDI predictive rules by ILP from generated domain and protein data

There have been many ILP systems that are successfully applied to various problems in bioinformatics, such as protein secondary structure prediction [13], protein fold recognition [12], and protein-protein interaction prediction [24]. The proposed ILP framework for predicting DDIs from multiple genome databases is described in Algorithm 2.

Algorithm 2 Discovering rules for domain-domain interactions

Input:

The domain-domain interactions database *InterDom* Number of negative examples ($\neg d_{ij}$) N

Multiple genomic data used for extracting background knowledge ($S^{Pfam}, S^{InterPro}, S^{PROSITE}, S^{PRINTS} S^{Uniprot}, S^{MIPS}, S^{GO}$)

Output: Set of rules R for domain-domain interaction prediction.

1: $R := \emptyset$.
2: Extract positive examples set $S_{interact}$ from *InterDom*.
3: Generate negative examples $\neg d_{ij}$s by selecting randomly N domain pairs from D where $\neg d_{ij} \notin S_{interact}$.
4: **for each** domain $d_i \in D$
5: **call** Algorithm 1 to generate values for features d_i^ms from domain database M ($\forall M \in (S^{Pfam}, S^{InterPro}, S^{PROSITE}, S^{PRINTS})$) and protein features pf_i^ls from protein database L ($\forall L \in (S^{Uniprot}, S^{MIPS}, S^{GO})$).
6: Integrate all domain features df_i^m and protein features pf_i^l for generating background knowledge
7: Run Aleph to induce rules r.
8: $R := R \cup \{r\}$.
9: **return** R.

In the framework, the common procedure of ILP method is presented. Step 2 and Step 3 are for generating positive and negative examples (see Section 3). In Steps 4 to 7, we extracted background knowledge including both domain features and protein features (see Section 2.1). Aleph system [1] is applied to induce rules in Step 8 . 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.

3 Evaluation

3.1 Experiment design

In this paper, we used 3000 positive examples from InterDom database. InterDom database consists of DDIs of multiple organisms [16]. Positive examples are domain-domain interactions in InterDom database which have score threshold over 100 and no false positives. The set of interacting pairs $S_{interact}$ in Algorithm 2 consists of these domain-domain interactions. Because there is no database for non domain-domain interaction, the negative examples $\neg d_{ij}$s are randomly generated. A domain pair $(d_i, d_j) \in D$ is considered to be a negative example, if the pair does not exist in the interaction set. In this paper, we chose different numbers of negatives (500, 1000, 2000, 3000 negative examples). To validate our proposed method, we conducted a 10-fold cross-validation test, comparing cross-validated sensitivity and specificity with results obtained by using AM [23] and SVM method. The AM method calculates a score d_{kl} for 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 them with our proposed method. We use the same database applying ILP to input AM and SVM. The probability threshold is set to 0.05 for the simplicity of comparison. For SVM method, we used SVM^{light} [9]. The linear kernel with default values of the parameters was used. For Aleph, we selected $minpos = 3$ and $noise = 0$, i.e. the lower bound on the number of positive examples to be covered by an acceptable clause is 3, and there are no negative examples allowed to be covered by an acceptable clause. These parameters are the smallest that allow us to induce rules with biological meaning. 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.

3.2 Analysis of experimental results

Table 2 shows the performance of Aleph compared with AM and SVM methods. Most of our experimental results had higher sensitivity and specificity compared with AM and SVM. 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, and thus is a measure of how accurately it identifies negatives. It can be seen from Table 2 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 an imbalanced learning problem.

The performance of method in terms of specificity and sensitivity are also statistically tested in terms of confidence intervals. Confidence intervals give us an estimate of the amount of error involved in our data. To estimate 95% confidence interval for each calculated specificity and sensitivity, we used t distribution. The 95% confidence intervals are shown in Table 2.

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 set to 3.

# Neg	Sensitivity			Specificity			# Rules
	AM	SVM	Aleph	AM	SVM	Aleph	
500	0.49±.027	**0.86**±.010	0.83±.016	0.54±.074	0.24±.004	**0.61**±.075	127
1000	0.57±.018	0.63±.074	**0.78**±.042	0.44±.033	0.49±.009	**0.68**±.042	173
2000	0.50±.015	0.32±.014	**0.69**±.027	0.50±.021	0.73±.015	**0.80**±.018	196
3000	0.49±.021	0.22±.017	**0.62**±.027	0.53±.022	0.81±.013	**0.84**±.010	235
Avg.	0.51±.020	0.51±.029	**0.73**±.028	0.50±.038	0.57±.010	**0.73**±.036	

Besides comparing cross-validated sensitivity and specificity, cross-validated accuracy and precision are considered. The average accuracy (0.76) and precision (0.80) of Aleph are higher than both AM method (0.51 and 0.56 respectively) and SVM method (0.66 and 0.72 respectively).

The experimental results have shown that ILP approach potentially predicts DDIs with high sensitivity and specificity. Further more, the inductive rules of ILP encouraged us to discover lots of comprehensive relations between DDIs and domain/protein features. Analysing our results in comparison with information in biological literatures and books, we found that ILP induced rules could be applied to the further related studies in biology.

The simplest rule covering many examples of positives is the self-interact rule. Many domains tend to interact with themselves (86 domain-domain interactions among positive examples). This phenomenon is reasonable because indeed lots of

proteins interact with themselves, and they consist of many of the same domains. Figure 2 shows some other induced rules.

Fig. 2 Some induced rules obtained with $minpos = 3$.

Rule 1 [Pos cover = 15 Neg cover = 0]
$interact\_domain(A, B) : -ig(A, B, C), C = 5,$
$function\_category(B, transcription),$
$protein\_category(A, transcription\_factors).$

Rule 2 [Pos cover = 20 Neg cover = 0]
$interact\_domain(A, B) : - \quad num\_int(B, C), gteq(C, 20),$
$complex\_category(A, scf\_comlexes).$

Rule 3 [Pos cover = 51 Neg cover = 0]
$interact\_domain(A, B) : - \quad interpro(B, C), interpro(A, C), interpro2go(C, D).$

Rule 4 [Pos cover = 23 Neg cover = 0]
$interact\_domain(A, B) : - \quad prints(B, C), motif\_compound(C, compound(8)),$
$function\_category(A, protein\_synthesis).$

Rule 5 [Pos cover = 31 Neg cover = 0]
$interact\_domain(A, B) : - \quad prints(B, C),$
$motif\_compound(C, compound(13)), haskw(A, cell\_cycle).$

Rule 6 [Pos cover = 29 Neg cover = 0]
$interact\_domain(A, B) : - \quad num\_int(A, C), C = 7,$
$function\_category(B, metabolism), haskw(B, thread\_structure).$

Rule 7 [Pos cover = 32 Neg cover = 0]
$interact\_domain(A, B) : - \quad ig(A, A, C), C = 3,$
$function\_category(B, cell\_type\_differentiation),$
$phenotype\_category(A, nucleic\_acid\_metabolism\_defects).$

Rule 8 [Pos cover = 15 Neg cover = 0]
$interact\_domain(A, B) : - \quad phenotype\_category(B, conditional\_phenotypes)$
$hasft(A, domain\_rna\_binding\_rrm).$

Rule 9 [Pos cover = 16 Neg cover = 0]
$interact\_domain(A, B) : - \quad prosite(B, C),$
$prosite\_site(C, tubulin\_subunits\_alpha\_beta\_and\_gamma\_signature).$

Rule 10 [Pos cover = 37 Neg cover = 0]
$interact\_domain(A, B) : - \quad go(B, C), is\_a(C, D), hasft(A, chain\_bud\_site\_$
$selection\_protein\_bud5).$

In the set of induced rules, there are (1) rules of only domain features (*i.e.* Rule 9), rules of only protein features (*i.e.* Rule 8) and especially rules of mixture of both domain features and protein features (*i.e.* Rule 4, Rule 5). In rules, the coverage values presented are the average predictive coverage on the 10 folds.

Related to *motif compound* feature in domain, we found that the more motifs a domain has, the more interactions the domain has with other domains. This

means that domains which have many conserved motifs tend to interact with others. And the interactions of these domains play an important role in forming stable domain-domain interactions in particular and protein-protein interactions in general [11]. Rule 4 shows that if we have two domains - one of them with eight motifs, and the other one belonging to proteins categorized in *protein_synthesis* function category, then the two domains interact.

Discovering the rules related to domain sites and domain signatures with predicate `prosite_site(domain,#prosite_site)`, we found some significant sites in domain joining in the domain-domain interactions. Rule 9 shows the relation between the accession numbers in Pfam database and PROSITE database, and then the signature information of domain in PROSITE database. This rule means that if one domain belongs to both Pfam database and PROSITE database and has *tubulin_subunits_alpha_beta_and_gamma_signature*, then it can interact with others. The rules like Rule 9 can be applied to understand protein-protein interaction interfaces and protein structures [20].

Rule 6 is an example which infers the relation between DDIs and biological pathways. From this rule, if we have an interacting domain pair, one of them has seven domain-domain interactions, and the other domain belongs to one protein which has keyword *thread_struture*, we can say that that protein functions in a certain metabolic pathway.

Thanks to inductive rules of ILP, we found a lot of relations between DDIs and different domain and protein features. We expect that the combination of these rules will be very useful for understanding DDIs in particular and protein structures, protein functions and protein-protein interactions in general.

4 Conclusion

We have presented an approach using ILP and multiple genome databases to predict domain-domain interactions. The experimental results demonstrated that our proposed method could produce comprehensible rules, and at the same time, performed well compared with other work on domain-domain interaction prediction. In future work, we would like to investigate further the biological significance of novel domain-domain interactions obtained by our method, and apply the ILP approach to other important tasks, such as determining protein functions, protein-protein interactions, and the sites, and interfaces of these interactions using domain-domain interaction data.

References

1. A.Srinivasan. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/.
2. X.W. Chen and M. Liu. Prediction of protein-protein interactions using random decision forest framework. *Bioinformatics*, 21(24):4394–4400, 2005.
3. Comprehensive Yeast Genome Database. http://mips.gsf.de/genre/proj/yeast/.
4. InterPro database concerning protein families and domains. http://www.ebi.ac.uk/interpro/.
5. M. Deng, S. Mehta, F. Sun, and T. Chen. Inferring domain-domain interactions from protein-protein interactions. *Genome Res.*, 12(10):1540–1548, 2002.

6. Protein families database of alignments and HMMs. http://www.sanger.ac.uk/Software/Pfam/.
7. Protein figerprint. http://umber.sbs.man.ac.uk/dbbrowser/PRINTS/.
8. D. Han, H.S.Kim, J.Seo, and W.Jang. A domain combination based probabilistic framework for protein protein interaction prediction. In *Genome Inform. Ser. Workshop Genome Inform*, page 250259, 2003.
9. Thorsten Joachims. http://svmlight.joachims.org/.
10. R.M. Kim, J. Park, and J.K. Suh. Large scale statistical prediction of protein - protein interaction by potentially interacting domain (PID) pair. In *Genome Inform. Ser. Workshop Genome Inform*, pages 48–50, 2002.
11. H. S. Moon, J. Bhak, K.H. Lee, and D. Lee. Architecture of basic building blocks in protein and domain structural interaction networks. *Bioinformatics*, 21(8):1479–1486, 2005.
12. M.Turcotte, S.H.Muggleton, and M.J.E.Sternberg. Protein fold recognition. In *Proc. of the 8th International Workshop on Inductive Logic Programming (ILP-98)*, pages 53–64, 1998.
13. S. Muggleton, R.D. King, and M.J.E. Sternberg. Protein secondary structure prediction using logic-based machine learning. *Protein Eng.*, 6(5):549–, 1993.
14. S. K. Ng and S. H. Tan. Discovering protein-protein interactions. *Journal of Bioinformatics and Computational Biology*, 1(4):711–741, 2003.
15. S.K. Ng, Z. Zhang, and S.H Tan. Integrative approach for computationally inferring protein domain interactions. *Bioinformatics*, 19(8):923–929, 2003.
16. S.K Ng, Z Zhang, S.H Tan, and K. Lin. InterDom: a database of putative interacting protein domains for validating predicted protein interactions and complexes. *Nucleic Acids Res*, 31(1):251–254, 2003.
17. Database of Interacting Proteins. http://dip.doe-mbi.ucla.edu/.
18. PROSITE: Database of protein families and domains. http://kr.expasy.org/prosite/.
19. Gene Ontology. http://www.geneontology.org/.
20. D. Reichmann, O. Rahat, S. Albeck, R. Meged, O. Dym, and G. Schreiber. From The Cover: The modular architecture of protein-protein binding interfaces. *PNAS*, 102(1):57–62, 2005.
21. Universal Protein Resource. http://www.pir.uniprot.org/.
22. R. Riley, C. Lee, C. Sabatti, and D. Eisenberg. Inferring protein domain interactions from databases of interacting proteins . *Genome Biology*, 6(10):R89, 2005.
23. E. Sprinzak and H. Margalit. Correlated sequence-signatures as markers of protein-protein interaction. *Journal of Molecular Biology*, 311(4):681–692, 2001.
24. T.N. Tran, K.Satou, and T.B.Ho. Using inductive logic programming for predicting protein-protein interactions from multiple genomic data. In *PKDD*, pages 321–330, 2005.
25. K. Wilson and J.Walker. *Principle and Techniques of Biochemistry and Molecular Biology*. Cambridge University Press, 6 edition, 2005.
26. J. Wojcik and V. Schachter. Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics*, 17(suppl-1):S296–305, 2001.