BIOINFORMATICS ORIGINAL PAPER Vol. 21 no. 24 2005, pages 4394–4400
doi:10.1093/bioinformatics/bti721

Systems Biology

Prediction of protein–protein interactions using random decision forest framework

Xue-Wen Chen* and Mei Liu

Bioinformatics and Computational Life-Sciences Laboratory, ITTC, Department of Electrical Engineering and Computer Science, The University of Kansas, 1520 West 15th Street, Lawrence, KS 66045, USA

Received on August 14, 2005; revised on October 6, 2005; accepted on October 14, 2005
Advance Access publication October 18, 2005

ABSTRACT

Motivation: Protein interactions are of biological interest because they orchestrate a number of cellular processes such as metabolic pathways and immunological recognition. Domains are the building blocks of proteins; therefore, proteins are assumed to interact as a result of their interacting domains. Many domain-based models for protein interaction prediction have been developed, and preliminary results have demonstrated their feasibility. Most of the existing domain-based methods, however, consider only single-domain pairs (one domain from one protein) and assume independence between domain–domain interactions.

Results: In this paper, we introduce a domain-based random forest of decision trees to infer protein interactions. Our proposed method is capable of exploring all possible domain interactions and making predictions based on all the protein domains. Experimental results on *Saccharomyces cerevisiae* dataset demonstrate that our approach can predict protein–protein interactions with higher sensitivity (79.78%) and specificity (64.38%) compared with that of the maximum likelihood approach. Furthermore, our model can be used to infer interactions not only for single-domain pairs but also for multiple domain pairs.

Contact: xwchen@ku.edu

Availability: Source code is written in Java and is available upon request from the authors.

Supplementary information: http://www.ittc.ku.edu/~xwchen/PPI/random_forest_PPI

1 INTRODUCTION

Proteins play an essential role in nearly all cell functions such as composing cellular structure and promoting chemical reactions. The multiplicity of functions that proteins execute in most cellular processes and biochemical events is attributed to their interactions with other proteins. It is thus critical to understand protein–protein interactions (PPIs).

In recent years, high throughput technologies have provided experimental tools to identify PPIs systematically and have generated tremendous amount of protein interaction data. However, the high throughput experiments are often associated with high false positives and false negatives (Mrowka *et al*., 2001). The experiments are also tedious and labor-intensive. In addition, the number of possible protein interactions within one cell is enormous, which makes experimental verification of each individual interaction impractical. The need arises in seeking complementary *in silico* methods that are capable of accurately predicting interactions.

A number of computational approaches for protein interaction discovery have been developed over the years. These methods differ in feature information used for protein interaction prediction. Earlier methodologies focus on estimating the interaction sites by recognizing specific residue motifs (Kini and Evans, 1996) or using features and properties related to interface topology, solvent accessible surface area and hydrophobicity (Jones and Thornton, 1997). Some other computational techniques are based on genomic sequence analysis, e.g. analyzing correlated mutations in amino acid sequences between interacting proteins (Pazos *et al*., 1997), searching for conservation of gene neighborhoods and gene order (Dandekar *et al*., 1998), using the gene fusion method or 'Rosetta stone' (Enright *et al*., 1999; Marcotte *et al*., 1999), employing genomic context to infer functional protein interactions (Huynen *et al*., 2000) and exploring the principle on similarity of phylogenetic trees for protein interaction prediction (Goh *et al*., 2000; Pazos and Valencia, 2001). Several papers propose to predict protein interaction sites based on profiles of a residue and its neighbors (Fariselli *et al*., 2002; Zhou and Shan, 2001). There is also a method to predict PPIs based on the primary structure and associated physicochemical properties (Bock and Gough, 2001).

Recently, there is a growing interest in domain-based protein interactions prediction. Preliminary results have demonstrated their feasibility. Protein domains are considered to be the building blocks of proteins. Domains are structural and/or functional units of proteins that are conserved through evolution to represent protein functions or structures. The assumption that proteins interact with each other through their domains is widely accepted.

A number of domain-based approaches have been proposed. One of the pioneering works is an association method proposed by Sprinzak and Margalit (2001). The association method defines a simple measure of interaction between two domains as the fraction of interacting protein pairs among all protein pairs containing the domain pair. It may assign high association scores to domain pairs with low frequency, which may not correspond well to the interaction probability. Kim *et al*. (2002) improved the association method by considering the number of domains in each protein. An integrative approach is proposed by Ng *et al*. (2003a) to infer putative domain–domain interactions from three data sources, including experimentally derived protein interactions, protein complexes and Rosetta stone sequences. The interaction scores for domain pairs in the data sources, protein interactions and protein

*To whom correspondence should be addressed.

complexes are obtained with a calculation scheme similar to the association method by considering frequency of each domain in the interacting protein pairs. The aforementioned approaches do not consider the fact that multiple domains in a protein can interact with multiple domains in another and the possibility of a domain pair appearing in both interacting and non-interacting protein pairs. Han *et al*. (2003, 2004) proposed a domain combination-based method. It considers all possible domain combinations as the basic units of each protein. The domain combination interaction probability is also based on the number of interacting protein pairs containing the domain combination pair and the number of domain combinations in each protein. The method considers the possibility of domain combinations appearing in both interacting and non-interacting sets of protein pairs. However all of these methods suffer from a general limitation of the association method, which is ignoring other domain–domain interaction information between the protein pairs.

A graph-oriented approach is proposed in Wojcik and Schachter (2001) called the 'interacting domain profile pairs' (IDPP) approach. The method uses a combination of sequence similarity search and clustering based on interaction patterns and domain information. Their use of domain profile pairs has provided better predictions than methods solely based on sequence information. The aim of the method is to infer protein interaction map of a target organism from a large-scale interaction map of a source organism, which can be very expensive to obtain. An optimization approach, maximum likelihood estimation (MLE), is applied in Deng *et al*. (2002). It infers domain interactions by maximizing the likelihood of the observed protein interaction data. The probabilities of interaction between two domains (only single-domain pair is considered) are optimized using the expectation-maximization (EM) algorithm.

Most of the abovementioned methods assume independency of domain–domain interactions and single-domain interaction. In this paper, we propose a novel domain-based random forest framework to predict PPIs. Standard random forest approach (Ho, 1998; Breiman, 2001) was used by Qi *et al*. (2005) in predicting PPIs. However, their method uses other protein properties instead of protein domains as the feature vector. In their method, the standard random forest was simply applied to determine similarity between protein pairs, and then this similarity is used by a *k*-nearest neighbor (kNN) algorithm to classify protein pairs.

In our application, the PPI prediction is formulated as a binary classification problem with novel feature representation. Due to the features' unique characteristics, the standard random forest algorithm cannot be directly applied to the protein-interaction inference problem. Instead, a new framework based on random forest is proposed here for PPI prediction. In the proposed method, rather than considering single-domain pair as the basic unit of protein interactions, we explore contributions of all the possible domain combinations to protein interactions. In addition, the proposed model does not assume that domain pairs are independent of each other. Our method is compared with the MLE method, and better results (in terms of the specificity and sensitivity) are obtained. Furthermore, the decision tree-based model can be used to infer domain–domain interactions for each predicted interacting protein pair.

The paper is organized into four sections. Section 2 introduces the standard random forest algorithm and our novel algorithm. The experimental results are presented in Section 3. Finally, conclusions are drawn in Section 4.

2 METHODS

2.1 Feature representation

We formulate the PPI prediction problem as a two-class classification problem: each protein pair is a sample belonging to either 'interaction' class (the two proteins interact with each other) or 'non-interaction' class (the two proteins do not interact with each other). In our application, a protein pair is characterized by the domains existing in each protein. Among all proteins in our dataset (both training and testing) there are 4293 unique Pfam domains. Thus, each protein pair is represented by a vector of 4293 features where each feature corresponds to a domain. Let $D = [X_1, X_2, \ldots, X_n]$ represent the n training samples and $X_i = [x_1^{(i)}, x_2^{(i)}, \ldots, x_{4293}^{(i)}, y_i]$ represent the i-th sample with 4293 feature attributes x_j belonging to the class y_i. In our problem formulation, $y_i = 1$ stands for the 'interaction' class and 0 for the 'non-interaction' class. Each feature x_j has a discrete value of 0, 1 or 2. If the sample protein pair does not contain the domain, then the associated feature value is 0. If one of the proteins contains the domain, then the value is 1. Finally, if both proteins have the domain, then it is 2. This ternary-valued feature representation is different from other domain-based methods. It allows us to represent each protein pair by a feature vector with protein domain information. The ternary-valued representation is necessary for our applications as domains may interact with themselves. While training decision trees with a binary-valued representation is faster than that with a ternary-valued representation, a binary-valued representation may not be able to distinguish between protein pairs with a domain existing in one protein and those with the domain existing in both proteins. A potential problem associated with the current representation is that it cannot tell whether two domains are from the same protein or from two different proteins. To handle this situation, in decision-making procedures, we consider the domains as an interacting domain pair only if these domains (two or more) are from different proteins and they lead to an interaction classification.

2.2 Random decision forest

In our application, the number of features for each sample is large. When the input space is extraordinarily large, random subspace (RS) feature selection introduced by Ho (1998) can potentially improve classifier diversity. Breiman (2001) proposes 'random forests of classifiers', which involves in developing an ensemble of decision trees from randomly sampled subspaces of the input features, and final classification is obtained by combining results from the trees via voting. It is shown in Ho *et al*. (1994) and Ho (1995) that combining multiple trees produced in randomly selected subspaces can improve the generalization accuracy. It is crucial to produce a large number of sufficiently different trees when using the combined power of multiple trees for increase in accuracy. The use of randomization in feature selection is a way to explore various possibilities of subspaces. While most classification methods suffer from the curse of dimensionality, the RS feature selection method can take advantage of the high dimensionality. In contrast to the Occam's Razor, the method improves accuracy as it grows in complexity as shown in Ho (1998).

The random decision forest constructs many decision trees and each is grown from a different set of training data. To construct individual decision trees, training samples are randomly selected with replacement from the original training dataset. If the number

of samples in the original training set is N, then N samples are randomly drawn with replacement. At each splitting or decision node, it determines the best splitting feature from a randomly selected subspace of m features where m is much smaller than M total number of features. Each tree in the forest is grown to the largest extent possible without pruning. To classify a new object, each tree in the forest gives a classification, which is interpreted as the tree 'voting' for that class. The final classification of the object is determined by majority votes among the classes decided by the forest of trees.

2.3 Domain-based random decision forest framework

Similar to the standard random decision forest algorithm, our individual decision tree in the forest is also built from different set of training data. For each tree, positive and negative samples from the original training dataset are selected randomly with replacement. To preserve the same proportion of positive samples among all samples, we drew samples from positive and negative sets separately. While building a decision tree, the standard random forest randomly selects a subspace of features to focus on at each splitting node. However, our application is unique and this randomness introduced may not work as well as in other applications. In other applications, all features contain information for classification no matter what the values are. In our application, a feature with a value 0 does not give us any information about the protein pair's interaction status. Consider the following example. A protein pair (P_1, P_2) has domains $\{a, b, c\}$ and $\{d, e\}$, respectively. Assume that the true domain interacting pair is (a, d) and it has appeared frequently in many different protein pairs. With random selection, domain $\{a\}$ or $\{d\}$ may not be selected properly although they are the domains that exist in many proteins. For example, we could randomly select $\{a\}$ and $\{e\}$ as the splitting attributes to classify the protein pair as interacting. Even though the classification is correct, we have the wrong information on the domain interacting pair.

In order to address this problem, we introduce probability selection for the feature subspace. Each feature in the entire feature space is assigned with a selection probability. The probability is calculated based on the number of interacting protein pairs in the original training dataset that include such domain feature (i.e. at least one protein of the pair contains the domain). A Roulette wheel representing the feature set is then created. Each domain feature is assigned to a real number in range [0.0, 1.0], which represents a section on the wheel. The range is calculated based on the probabilities. Thus, if a domain is common among large number of proteins, it will be selected with large chance.

Each decision tree is built level by level from the bootstrapped training dataset starting at the root. At each splitting node of a decision tree, in order to form the feature subspace, we spin the Roulette wheel by generating random numbers. If the random number generated falls in between a feature's range, then the feature is added to the subset unless it is already used as a splitting attribute by one of the parent nodes in the same branch up to the root. This process continues until $\log_2 M + 2$ (M is the total number of features) features are selected for the feature subspace. We then pick the best splitting attribute from the subspace based on a measure called 'goodness of split', which assesses how well the attributes discriminate between classes.

In this study, the information gain splitting criteria by Quinlan (1979, 1983) is used as the 'goodness of split' measure, which is based on the classic formula from information theory. The information gain measures the theoretical information content of a code by $\sum_i p_i \log(p_i)$, where p_i is the probability of the i-th message.

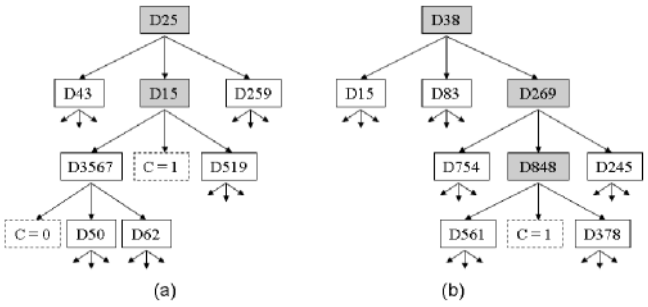

Fig. 1. For illustration, assume (**a**) and (**b**) are two decision trees in the forest. The leftmost branch is taken if feature value is 0, middle branch if 1 and rightmost branch if 2. The solid boxes represent decision nodes with selected domain features. Shaded boxes represent domains that form interacting domain or domain combination pairs. The dashed boxes represent classification.

In traditional random forest, individual trees are completely built without pruning. Due to the high dimensionality of our data, each tree is expected to be extremely large if grown completely. Thus in addition to stop splitting a node only when all samples are well classified, we applied some early stopping criterions as a forward pruning technique that stops pursuing branches with little statistical significance. A node in a decision tree also stops splitting when any of the following conditions is met: node is at the maximum level, node impurity is smaller or equal to a certain threshold, or minimum number of samples is remained at the node. Samples at a node are considered to be well classified when all belong to the same class. Node impurity is defined as the proportion of samples that are in the minority class.

$$\text{Node impurity} = \frac{\text{No. of samples at the node in the minority class}}{\text{Total no. of samples at the node}}. \quad (1)$$

A forest forms with construction of many decision trees. To classify a protein pair, rather than using all trees in the forest to vote as in the standard random forest algorithm, votes are obtained only from the trees that contain at least one domain feature from each protein of the pair as a splitting attribute. This is necessary, as a decision tree without domains from a protein pair does not indicate that the proteins are not interacting with each other. Therefore, a sufficient number of decision trees need to be built to cover all domains existed in the training samples. We determine this number by making sure that each domain feature in the training samples is covered by at least a certain number of trees. If no tree is able to vote for a protein pair, we assume the protein pair to be non-interacting. Otherwise, protein pairs are classified based on the majority votes. A tree casts vote of value 1 for interacting pairs, and vote of value 0 for non-interacting pairs.

After training, the classifier can also be used to determine domain–domain interactions. Since each splitting attribute represents a single domain, if an existence of two or more such attributes from different proteins leads to an interaction classification, then we can interpret the domains as forming an interacting domain pair or domain combination pair (Fig. 1). For example, assuming that

(a) and (b) in Figure 1 are decision trees in the forest, we can infer interacting domain pairs from the shaded decision nodes when a test protein pair follows through the path and reaches prediction 1 (represented as dashed boxes) and if the domains are from different proteins. A test protein pair follows the leftmost branch of a node if it does not contain the domain feature, thus feature value is 0 (more details in Section 2.1). The middle branch is followed if feature value is 1, and otherwise the rightmost branch is chosen. From tree (a), it classifies a protein pair to be interacting if one protein in the pair contains domain 25 and the other protein contains domain 15. Based on the interaction prediction, one can also conclude that domains 15 and 25 may interact with each other because they contribute to an interaction prediction. Similarly in tree (b), domains 38, 269 or 848 may form interacting domain combination pairs.

3 EXPERIMENTAL RESULTS

3.1 Data sources

PPI data for the yeast organism were collected from the DIP (Database of Interacting Proteins) (Salwinski *et al*., 2004; Deng *et al*., 2002; Schwikowski *et al*., 2000, Xenarios *et al*., 2001, http://dip.doe-mbi.ucla.edu). The dataset used by Deng *et al*. (2002) is a combined interaction data experimentally obtained through two-hybrid assays on *Saccharomyces cerevisiae* by Uetz *et al*. (2000) and Ito *et al*. (2000). Schwikowski *et al*. (2000) gathered their data from yeast two-hybrid, biochemical and genetic data.

Initially, we obtained 15 409 interacting protein pairs in the yeast organism from DIP, 5719 pairs from Deng *et al*. (2002) and 2238 pairs from Schwikowski *et al*. (2000). The datasets are then combined by removing the overlapping interaction pairs. Because domains are the basic units of protein interactions, proteins without domain information cannot provide any useful information for our prediction. Therefore, we excluded the pairs where at least one of the proteins has no domain information. Finally, 9834 protein interaction pairs remained among 3713 proteins, and it is separated evenly (4917 pairs each) into training and testing datasets. Since non-interacting protein data are not available, the negative samples are randomly generated. A protein pair is considered to be a negative sample if the pair does not exist in the interaction set. Total of 8000 negative samples were generated and also separated into two halves. Both final training and testing datasets contain 8917 samples, 4917 positive and 4000 negative samples.

The protein domain information is gathered from Pfam (Bateman *et al*., 2004), which is a protein domain family database that contains multiple sequence alignments of common domain families. In Pfam, hidden Markov model profiles were used to find domains in new proteins. The Pfam database consists of two parts: Pfam-A and Pfam-B. Pfam-A is manually curated, and Pfam-B is automatically generated. Both Pfam-A and Pfam-B families are used here. In total, there are 4293 Pfam domains defined by the set of proteins.

3.2 Evaluation criteria

To evaluate the methods for predicting PPIs, we use both specificity and sensitivity. The specificity SP is defined as the percentage of matched non-interactions between the predicted set and the observed set over the total number of observed non-interactions. The sensitivity, denoted by SN, is defined as the percentage of matched interactions over the total number of observed interactions.

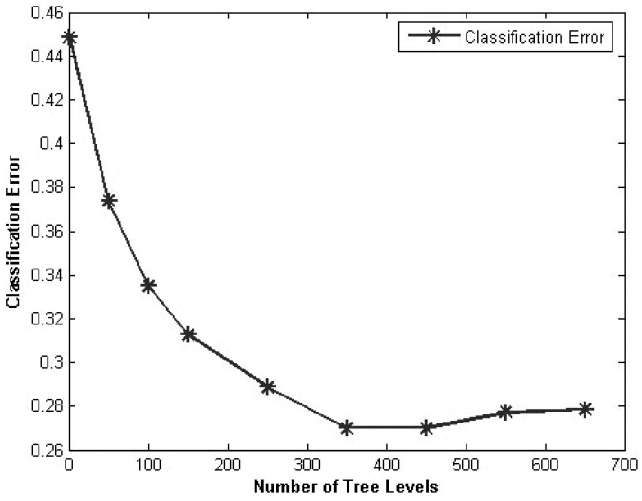

Fig. 2. Classification error comparison of different tree sizes in 5-fold cross-validation.

3.3 Predicting PPI

The accuracy of a random forest depends on the strength of individual tree classifier, which may be affected by tree size. In our implementation, we have set three stopping criteria to limit the tree size, and they are maximum tree level, impurity and minimum node size thresholds. Minimum node size defines the minimum number of samples to be classified by each node. In our forest, each decision tree is constructed with node impurity threshold of 0.01, and the minimum number of samples at a node is three. Among those early stopped nodes, <10% reached impurity and minimum node size thresholds. The maximum tree level criterion has the most impact in restricting the tree size. Thus, we grow forests of trees with different heights to make an appropriate parameter choice on maximum tree level threshold. We used 5-fold cross-validation and found that classification error rates over the validation sets decrease first as the tree levels increase. This is due to the increased performance of each individual tree. Because of majority voting, if each individual tree in a forest performs better, then the entire forest will also perform better. As shown in Figure 2, the forest classification error rate reaches the minimum for the heights of 350 and 450 and increases slightly after 450. We will select the maximum tree size at 450 levels.

To determine an appropriate number of trees in a forest, we set a limit on minimum coverage of each domain feature at 30 trees. In other words, it makes sure that each domain feature appeared in the training dataset is one of splitting attributes in at least 30 trees. In this way, we guarantee that at least a certain number of trees will vote to classify each protein pair. From the experiment, we found that with 100 trees in the forest, each domain feature is covered by at least 74 trees.

Training the forest of decision trees is the most computationally intensive part of the entire prediction process. Running on a 3.2 GHz Xeon computer, it takes ∼2–3 min to construct a decision tree. After the model is trained, predictions can be carried out very quickly.

The result of our method is compared with the MLE results (Deng *et al*., 2002). In their paper, different values of the false positive and false negative rates are evaluated. Their results show no significant accuracy change among the various values. Therefore, we picked

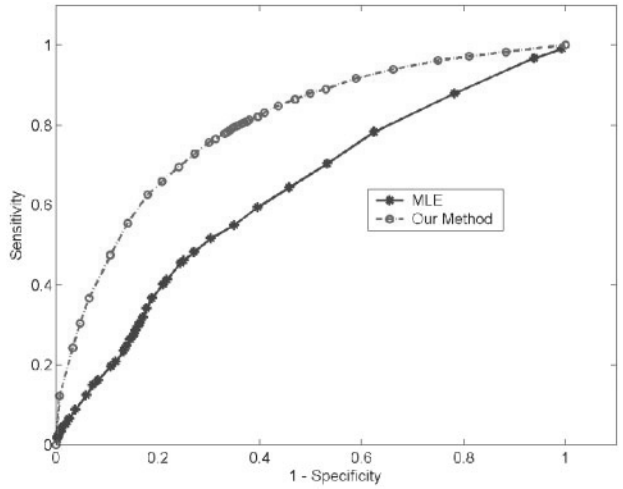

Fig. 3. ROC curves of our method and the MLE method.

Table 1. Accuracy comparison

	Our Method	MLE
True positive (TP)	3923	3850
False positive (FP)	1425	2499
True negative (TN)	2575	1501
False negative (FN)	994	1067
Sensitivity (SN)	79.78%	78.30%
Specificity (SP)	64.38%	37.53%

one of the values to train the MLE method over our training dataset, where the false positive rate $f_p = 1.0E-5$ and the false negative rate $f_n = 0.85$. For our application, we chose a forest of 150 trees and each with maximum height equal to 450. Impurity and minimum node size are set to 0.01 and 3, respectively. Figure 3 compares the ROC curves of our method and the MLE on the test dataset. The ROC curve of our model is constructed by changing threshold we place on interaction prediction. Typically, majority votes win where the threshold is 0; however, the threshold can be changed. For example, a threshold at 5 implies that at least five more interaction votes than non-interaction votes are necessary to classify a protein pair as interacting. Otherwise, the pair is classified as non-interacting. Therefore, with different thresholds, our model would perform differently in terms of specificity and sensitivity. It is clearly shown in Figure 3 that our method outperforms the MLE method in prediction accuracy. Table 1 shows the results of our method and the MLE over the test dataset. With comparable sensitivities fixed at approximately the same level 79%, our method can achieve 64.38% in specificity and the MLE can only reach 37.53% in specificity.

3.4 Inferring domain–domain interactions

For each correctly predicted PPI pair, we can derive domains involved in the decision process by tracing the branch or path the protein pair took to reach the prediction from trees that made the correct classification. The branch or path contains domains from the protein pair that contribute to the correct classification. Thus, these domains are said to be interacting with each other.

Table 2. Examples of inferred single-domain pairs also identified by Pfam

Domain A Pfam id	Name	Domain B Pfam id	Name
PF00069	Pkinase	PF00023	Ank
PF00069	Pkinase	PF02984	Cyclin_C
PF00069	Pkinase	PF00134	Cyclin_N
PF00364	Biotin_lipoyl	PF02852	Pyr_redox_dim
PF00117	GATase	PF02786	CPSase_L_D2
PF00117	GATase	PF02787	CPSase_L_D3
PF00117	GATase	PF00289	CPSase_L_Chain
PF00071	Ras	PF00996	GDI
PF00560	LRR_1	PF00076	RRM_1
PF00183	HSP90	PF00515	TPR_1

The total number of single-domain pairs (one domain from one protein) that are predicted to form interactions is 4366. Among them, 1891 pairs are found with Pfam-B domains for which interaction information is not available on Pfam or other sources. The remaining 2475 single-domain pairs contain Pfam-A domains. Among the 2475 predicted single Pfam-A domain pairs, 95 of them are found in the iPfam database (Finn *et al*., 2005) and 2239 of them are found in the InterDom database (a database of putative interacting domains, developed by Ng *et al*., 2003b). Table 2 lists some of the single-domain interacting pairs identified by our method. Those domain pairs are also identified by the iPfam as interacting domain pairs. The iPfam contains domain–domain interactions observed in PDB entries by mapping Pfam domains onto the PDB structures. For example, the domain biotin_lipoyl (PF00364) in Table 2 is annotated as biotin-requiring enzyme and it has a conserved lysine residue, which binds to biotin or lipoicacid. Biotin performs catalysis in some carboxyl transfer reactions and is covalently attached to a lysine residue via an amide bond. The pyr_redox_dim (PR02852) domain is annotated as pyridine nucleotide-disulphide oxidoreductase, dimerization domain and determined to involve in oxidation–reduction reaction. Table 3 lists some identified single-domain interaction pairs that are not found in iPfam. Those domain pairs are found to be interacting pairs with a high confidence by the InterDom (Ng *et al*., 2003b). For example, SH3 (PF00018) and Pkinase (PF00069) in Table 3 are derived from a PPI only involving single-domain proteins. A protein is considered as a single-domain protein if it has only one domain and the domain accounts for at least 50% of the protein length (Ng *et al*., 2003b). The domain interactions derived from single-domain protein interactions are usually considered to be highly likely. The SH3 domain is also found to interact with Pkinase_Tyr (PF07714) by iPfam (Finn *et al*., 2005). Pkinase and Pkinase_Tyr are both members of the protein kinase superfamily clan. A complete list of domain interaction pairs is available on the Supplementary page.

While most of the existing domain-based methods can only infer the interaction for single-domain pairs, our method is capable of retrieving more than two domains for each protein from a branch. This is attractive, as in some PPI it is highly probable that more than two domains form a combination in a protein to interact

Table 3. Examples of inferred single-domain pair also identified by InterDom

Domain A			Domain B		
Pfam id	Name	Description	Pfam id	Name	Description
PF00153	Mito_carr	Mitochondrial carrier protein	PF01423	LSM	LSM domain
PF00248	Aldo_ket_red	Aldo/keto reductase family	PF00106	adh_short	short chain dehydrogenase
PF00155	Aminotran_1_2	Aminotransferase class I and II	PF00735	GTP_CDC	Cell division protein
PF00018	SH3_1	SH3 domain	PF00069	Pkinase	Protein kinase domain
PF00241	Cofilin_ADF	Cofilin/tropomyosin-type actin-binding protein	PF00400	WD40	WD domain, G-beta repeat
PF00694	Aconitase_C	Aconitase C-terminal domain	PF01028	Topoisom_I	Eukaryotic DNA topoisomerase I, catalytic core
PF00330	Aconitase	Aconitase family (aconitate hydratase)	PF01336	tRNA_anti	OB-fold nucleic acid binding domain
PF00501	AMP-binding	AMP-binding enzyme	PF01253	SUI1	Translation initiation factor SUI1
PF00022	Actin	Actin	PF01853	MOZ_SAS	MOZ/SAS family
PF00249	Myb_DNA-binding	Myb-like DNA-bindingdomain	PF00098	zf-CCHC	Zinc knuckle

Table 4. Examples of interacting domain combination pairs discovered

Domain Combination in protein A	Domain Combination in protein B
PF00083	PF00397; PF00168
PF00676	PF02779; PF02780
PF00036	PF00612; PF02736; PF00063
PF00009	PF02798; PF00043; PF00647
PF00459	PF00627; PF00442; PF00443
PF00026	PF00176; PF00271; PF00097; PB019909
PF01412	PF02826; PF00389; PF01842; PB042699
PF00076; PF00806	PF00248
PF00249; PF00569	PF00628
PF00004; PB030344; PF01426	PF02178
PF00006; PF02874; PF00306	PF00231
PF00169; PF00620; PF00617	PF00252

with another domain or domain combination in another protein. A domain combination is defined as two or more domains functioning as a whole during interaction. Some of our identified domain combinations are listed in Table 4. Domains in combination demonstrate strong association. For example, domains in the combination {PF00006, PF02874, PF00306} listed in Table 4 (row #11) are annotated by Pfam as ATP synthase alpha/beta family, nucleotide-binding domain; ATP synthase alpha/beta family, beta-barrel domain; and ATP synthase alpha/beta chain, C-terminal domain, respectively. Identified in iPfam, the three domains cooperate with each other to bind to the ATP synthase (PF00231). Another domain combination {PF00612, PF02736, PF00063} in Table 4 (row #3) is annotated as IQ calmodulin-binding motif; Myosin N-terminal SH3-like domain; and Myosin head (motor domain), respectively. The iPfam found that the domains work together to form bonds with the EF hand (PF00036). The domains of the combination {PF02779, PF02780} in the second row are Transketolase, pyridine binding domain, and Transketolase, C-terminal domain, respectively. The two domains are identified by iPfam to be binding together in proteins to interact with the dehydrogenase E1 component (PF00676). The total number of 867 domain combination pairs is identified and a complete list can be found on the Supplementary page. Verifying those predictions is a challenging task because currently there are not enough resources available on domain combination pairs.

With the putative domain–domain interactions, we can also predict PPIs. To exemplify this, we select some predicted domain–domain interactions and then find proteins that contain these domains to see if these proteins interact with each other or not. For example, identified by Pfam (Bateman *et al*., 2004), cell division control protein 7 (CDC7) contains protein kinase domain (PF00069). Both our model and Pfam identify the domain to be interacting with ankyrin repeat (PF00023) domain. Regulatory protein SWI6 is known to contain the Ankyrin repeat. Our model predicts the proteins, CDC7 and SWI6, to be interacting. Indeed, the protein CDC7 is a conserved Dbf4-dependent protein kinase (DDK). Bailis *et al*. (2003) has demonstrated that *Schizosac charomyces prombe* Hsk1 (CDC7) regulates replication initiation, interacts and phosphorylates the heterochromatin protein 1 (HP1), which is equivalent of SWI6. Another example, cell cycle protein kinase DBF2 contains the protein kinase domain (PF00069), and protein G2/mitotic-specific cyclin 2 (CLB2) contains Cyclin, N-terminal domain (PF00134). The PF00069 and PF00134 domain pair is inferred by our model and verified by Pfam as an interacting domain pair. The cell cycle protein kinase DBF2 and G2/mitotoic-specific cyclin 2 protein is predicted by our model to be an interacting protein pair. The DBF2 protein kinase is found to control the inactivation of the CLB2 (G2/mitotic-specific cyclin 2) kinase in late mitosis (Lee *et al*., 2001). This clearly demonstrates the potential of the proposed method for PPI prediction.

4 CONCLUSION

Proteins perform biological functions by interacting with other molecules. It is hypothesized that proteins interact with each other through specific intermolecular interactions that are localized to specific structural domains within each protein. Often, protein domains are structurally conserved among different families of proteins. Thus, understanding protein interactions at the domain level gives detailed functional insights into proteins. Most of the existing domain-based computational approaches for predicting protein interaction assume that domain pairs are independent of each other and consider the interactions between two domains only. In this paper, we introduce a new method, domain-based random decision forest framework, to predict PPIs. Major advantages of

this method are no assumption of domain independence is made and it can infer interacting single-domain and domain combination pairs. The system is capable of utilizing all the possible interactions between domains. We compared our results with the MLE method (Deng *et al*., 2002). The experimental results have shown that our method can predict protein–protein interactions with higher specificity and sensitivity than the MLE method. In addition, our method is particularly useful because domain–domain interactions can be inferred from the domains involved in predicting protein interactions, especially, this method allows for discovering interactions of domain combinations.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for their valuable comments. This publication was made possible partly by the National Science Foundation under Grant no. EPS-0236913 and matching support from the State of Kansas through Kansas Technology Enterprise Corporation and by NIH Grant P20 RR17708 from the Institutional Development Award (IDeA) Program of the National Center for Research Resources.

Conflict of Interest: none declared.

REFERENCES

Bailis,J.M. *et al*. (2003) Hsk1-Dfp1 is required for heterochromatin-mediated cohesion at centromeres. *Nat. Cell Biol*., **5**, 1111–1116.

Bateman,A. *et al*. (2004) The Pfam protein families database. *Nucleic Acids Res*., **32** (Database Issue), D138–D141.

Bock,J.R. and Gough,D.A. (2001) Predicting protein–protein interactions from primary structure. *Bioinformatics*, **17**, 455–460.

Breiman,L. (2001) Random forests. *Mach. Learning*, **45**, 5–32.

Dandekar,T. *et al*. (1998) Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci*., **23**, 324–328.

Deng,M. *et al*. (2002) Inferring domain–domain interactions from protein–protein interactions. *Genome Res*., **12**, 1540–1548.

Enright,A.J. *et al*. (1999) Protein interaction maps for complete genomes based on gene fusion events. *Nature*, **402**, 25–26.

Fariselli,P. *et al*. (2002) Prediction of protein–protein interaction sites in heterocomplexes with neural networks. *Eur. J. Biochem*., **269**, 1356–1361.

Finn,R.D. *et al*. (2005) iPfam: visualization of protein–protein interactions in PDB at domain and amino acid resolutions. *Bioinformatics*, **21**, 410–412.

Goh,C.S. *et al*. (2000) Co-evolution of proteins with their interaction partners. *J. Mol. Biol*., **299**, 283–293.

Han,D. *et al*. (2003) A domain combination based probabilistic framework for protein–protein interaction prediction. *Genome Inform. Ser. Workshop Genome Inform*., **14**, 250–259.

Han,D. *et al*. (2004) PreSPI: design and implementation of protein–protein interaction prediction service system. *Genome Inform*., **15**, 171–180.

Ho,T.K. *et al*. (1994) Decision combination in multiple classifier systems. *IEEE Trans. Pattern Anal. Mach. Intell*., **16**, 66–75.

Ho,T.K. (1995) Random decision forests. Montreal, In *Proceedings of the Third International Conference on Document Analysis and Recognition,* August 14–15, Montreal, Canada, pp. 278–282.

Ho,T.K. (1998) The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell*., **20**, 832–844.

Huynen,M. *et al*. (2000) Predicting protein function by genomic context: quantitative evaluation and qualitative inferences. *Genome Res*., **10**, 1204–1210.

Ito,T. *et al*. (2000) Toward a protein–protein interaction map of the budding yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between yeast proteins. *Proc. Natl Acad. Sci. USA*, **97**, 1143–1147.

Jones,S. and Thornton,J.M. (1997) Prediction of protein–protein interaction sites using patch analysis. *J. Mol. Biol*., **272**, 133–143.

Kim,W.K. *et al*. (2002) Large scale statistical prediction of protein–protein interaction by potentially interacting domain (PID) pair. *Genome Inform. Ser. Workshop Genome Inform*., **13**, 42–50.

Kini,R.M. and Evans,J.H. (1996) Prediction of potential protein–protein interaction sites from amino acid sequence. Identification of a fibrin polymerization site. *FEBS Lett*., **385**, 81–86.

Lee,S.E. *et al*. (2001) Order of function of the budding-yeast mitotic exit-network proteins Tem1, Cdc15, Mob1, Dbf2, and Cdc5. *Curr. Biol*., **11**, 784–788.

Marcotte,E.M. *et al*. (1999) Detecting protein function and protein–protein interactions from genome sequences. *Science*, **285**, 751–753.

Mrowka,R. *et al*. (2001) Is there a bias in proteome research? *Genome Res*., **11**, 1971–1973.

Ng,S. *et al*. (2003a) Integrative approach for computationally inferring protein domain interactions. *Bioinformatics*, **10**, 359–365.

Ng,S. *et al*. (2003b) InterDom: a database of putative interacting protein domains for validating predicted protein interactions and complexes. *Nucleic Acids Res*., **31**, 251–254.

Pazos,F. *et al*. (1997) Correlated mutation contain information about protein–protein interaction. *J. Mol. Biol*., **1**, 511–523.

Pazos,F. and Valencia,A. (2001) Similarity of phylogenetic trees as indicator of protein–protein interaction. *Protein Eng*., **14**, 609–614.

Qi,Y. *et al*. (2005) Random forest similarity for protein–protein interaction prediction from multiple sources. *Pac. Symp. Biocomput.,* 531–542.

Quinlan,J.R. (1979) Discovering rules from large collections of examples: a case study. In Michie,D. (ed.), *Expert Systems in the Micro Electronic Age*. Edinburgh University Press, Edinburgh. pp. 168–201.

Quinlan,J.R. (1983) Learning efficient classification procedures and their application to chess end games. In Michalski,R.S., Carbonell,J.G. and Mitchell,T.M. (eds), *Machine Learning: An Artificial Intelligence Approach*. Morgan Kaufmann, Los Altos. pp. 463–482.

Salwinski,L. *et al*. (2004) The Database of Interacting Proteins: 2004 update. *Nucleic Acids Res*., **32** (Database issue), D449–D451.

Schwikowski,B. *et al*. (2000) A network of protein–protein interactions in yeast. *Nat. Biotechnol*., **18**, 1257–1261.

Sprinzak,E. and Margalit,H. (2001) Correlated sequence-signatures as markers of protein–protein interactions. *J. Mol. Biol*., **311**, 681–692.

Uetz,P. *et al*. (2000) A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature*, **403**, 623–627.

Wojcik,J. and Schachter,V. (2001) Protein–protein interaction map inference using interacting domain profile pairs. *Bioinformatics*, **17**, S296–S305.

Xenarios,I. *et al*. (2001) DIP: the Database of Interacting Proteins: 2001 update. *Nucleic Acids Res*., **29**, 239–241.

Zhou,H. and Shan,Y. (2001) Prediction of protein interaction sites from sequence profile and residue neighbor list. *Proteins*, **44**, 336–343.