INCORPORATING GRAPH FEATURES FOR PREDICTING PROTEIN-PROTEIN INTERACTIONS

Martin S. R. Paradesi
Department of Computing and Information Sciences, Kansas State University
234 Nichols Hall, Manhattan, KS 66506-2302
p.martin.samuel.rao@gmail.com
Phone: +1 785 215 0876
Fax: +1 785 532 7353

Doina Caragea
Department of Computing and Information Sciences, Kansas State University
234 Nichols Hall, Manhattan, KS 66506-2302
dcaragea@ksu.edu
Phone: +1 785 532 7908
Fax: +1 785 532 7353

William H. Hsu
Department of Computing and Information Sciences, Kansas State University
234 Nichols Hall, Manhattan, KS 66506-2302
bhsu@cis.ksu.edu
Phone: +1 785 532 7905
Fax: +1 785 532 7353

INCORPORATING GRAPH FEATURES FOR PREDICTING PROTEIN-PROTEIN INTERACTIONS

ABSTRACT

This chapter presents applications of machine learning to predicting protein-protein interactions (PPI) in *Saccharomyces cerevisiae*. Several supervised inductive learning methods have been developed that treat this task as a classification problem over candidate links in a PPI network – a graph whose nodes represent proteins and whose arcs represent interactions. Most such methods use feature extraction from protein sequences (e.g., amino acid composition) or associated with protein sequences directly (e.g., GO annotation). Others use relational and structural features extracted from the PPI network, along with the features related to the protein sequence. Topological features of nodes and node pairs can be extracted directly from the underlying graph. This chapter presents two approaches from the literature (Qi *et al.*, 2006; Licamele & Getoor, 2006) that construct features on the basis of background knowledge, an approach that extracts purely topological graph features (Paradesi *et al.*, 2007), and one that combines knowledge-based and topological features (Paradesi, 2008). Specific graph features that help in predicting protein interactions are reviewed. This study uses two previously published datasets (Chen & Liu, 2005; Qi *et al.*, 2006) and a third dataset (Paradesi, 2008) that was created by combining and augmenting three existing PPI databases. The chapter includes a comparative study of the impact of each type of feature (topological, protein sequence-based, etc.) on the sensitivity and specificity of classifiers trained using specific types of features. The results indicate gains in the area under the sensitivity-specificity curve for certain algorithms when topological graph features are combined with other biological features such as protein sequence-based features.

Keywords: protein-protein interaction networks, link mining, feature construction, supervised learning

1. Introduction:

1.1 *Protein-protein interaction prediction problem*:

The term *protein-protein interaction (PPI)* refers to associations between proteins as manifested through biochemical processes such as formation of structures, signal transduction, transport, and phosphorylation. PPI plays an important role in the study of biological processes. Many PPIs have been discovered over the years and several databases have been created to store the information about these interactions such as BIND (Bader *et al.*, 2003), DIP (Salwinski *et al.*, 2004), MIPS (Mewes *et al.*, 2002), IntAct (Kerrien *et al.*, 2007) and MINT (Chatr-aryamontri *et al.*, 2007). In particular, more than 80,000 interactions between yeast proteins are available from various high-throughput interaction detection methods (von Mering *et al.,* 2002). These methods can detect if the interaction is either a physical binding between proteins or a functional association between proteins. Often, the functional association between two proteins leads to physical binding among them. Determining PPI using high-throughput methods is expensive and time-consuming. Furthermore, a high number of false positives and false negatives can be generated. Therefore, there is a need for computational approaches that can help in the process of identifying real protein-protein interactions.

Several methods have been designed to address the task of predicting protein-protein interactions using machine learning. Most of them use features from protein sequences (e.g., amino acids composition) or associated with protein sequences directly (e.g., GO annotation). However, the PPI network can be used to design node and topological features from the associated graph. Several methods use such relational and structural features extracted from the PPI network, along with the features related to the protein sequence. This chapter provides an overview of several machine learning methods for predicting PPI using the graph information extracted from a PPI network along with other available biological features of the proteins and their interactions, and shows the importance of the graph features for accurate predictions.

1.2 *Overview of PPI databases:*

Several PPI databases have been used to extract examples of PPIs for machine learning algorithms. We review the main PPI databases in what follows.

1.2.1 The Biomolecular Interaction Network Database (BIND):

BIND (Bader *et al.*, 2003) stores information about interactions, complexes and pathways. It also contains a number of large scale interaction and complex mapping experiments using yeast two-hybrid, mass spectrometry, genetic interactions and phage display. The group that maintains BIND has also developed a graphical analysis tool that provides users an understanding of functional domains in protein interactions. They have also developed a clustering tool that allows users to divide the protein interaction network into specific regions of interest. BIND assumes that interactions can occur between two biological 'objects', which could be proteins, RNA or DNA sequences, genes, molecular complexes, small molecules, or photons (light).

1.2.2 The Database of Interacting Proteins (DIP):

DIP (Salwinski *et al.*, 2004) is a database containing 18,343 interactions between 4,923 proteins validated from 23,366 experiments of the *Saccharomyces cerevisiae* organism. A few of the experiments from which they validate protein interactions are co-immunoprecipitation, yeast two-hybrid and in vitro binding assays. The group that maintains DIP has developed several quality assessment methods and uses them to identify the most reliable subset of the interactions that are inferred from high-throughput experiments. They also provide an online implementation of their evaluation methods that can be used to evaluate the reliability of new experimental and predicted interactions.

1.2.3 IntAct:

IntAct (Kerrien *et al.*, 2007) contains data such as experimental methods, conditions and interacting domains that is extracted entirely from publications and is manually annotated by curators. It also

formalizes the data by using a comprehensive set of controlled vocabularies in order to ensure data integrity. It is thus far the only published database that contains negative examples of protein interactions, *i.e.* explicitly identifies pairs of proteins that do not interact. The database contains 169,792 interactions between 63,427 proteins. These interactions were obtained from 8,477 experiments that were performed on several organisms. The web site provides tools allowing users to search, visualize and download data from the repository.

1.2.4 A Molecular INTeraction database (MINT):

MINT (Chatr-aryamontri *et al.*, 2007) stores molecular interaction data extracted from several publications. Most of its curation work is focused on physical interactions, direct interactions and colocalizations between proteins. Genetic or computationally inferred interactions are not included in the database. It contains 42,044 interactions between 5,256 proteins of the *Saccharomyces cerevisiae* organism. An online graph visualization and editing tool called "MINT Viewer" is available that allows users to view the interaction network and delete edges that are not of interest to the user.

1.2.5 The Munich Information Center for Protein Sequences (MIPS):

MIPS (Mewes *et al.*, 2002) provides information on Open Reading Frames (ORFs), RNA genes and other genetic elements. The research group that maintains MIPS has also applied techniques such as gene disruption in conjunction with powerful expression analysis and two-hybrid techniques as part of a systematic functional genome analysis. These methods generate information on how proteins cooperate in complexes, pathways and cellular networks. In addition, detailed information on transcription factors and their binding sites, transport proteins and metabolic pathways are being included or interlinked to the core data. The database also provides information on the molecular structure and the functional network of the yeast genome.

1.3 *Introduction to Machine Learning*:

Machine learning algorithms (Mitchell, 1997) offer some of the most cost-effective approaches to automated knowledge discovery and data mining (discovery of features, correlations, and other complex relationships and hypotheses that describe potentially interesting regularities) from large data sets. In particular, machine learning algorithms have proven to be very successful for many bioinformatics problems, including protein-protein interaction prediction.

In this chapter, we formulate the problem of PPI prediction as a classification task, i.e. a task where the learning algorithm is provided with experience in the form of labeled examples (a.k.a., training data set or data source) and is asked to classify new unlabeled examples in one of several possible classes. In our case, the training examples consist of existing information about protein interactions extracted from PPI databases. Each example is encoded using a set of variables called

attributes or features. A special attribute, called the class label, is used to represent the class to which that particular example belongs. The class label in a protein interaction prediction problem indicates whether the proteins in a candidate pair interact with each other (i.e., it takes two values: yes and no). The output of a learning algorithm for a classification task is called a classifier. Several strategies can be used to estimate the true error of a classifier. The simplest one is to divide the labeled data into a training set and a test set. The classifier is learned from the training set and its error is estimated using the test set. More commonly, the error is estimated by using a method called *k*-fold cross-validation. To use this method, the labeled data is divided into *k* folds. A classifier is learned from a training set consisting of *k* - 1 folds and tested on the remaining k^{th} fold. The estimate for the true error is obtained by taking the average of the error of the *k* possible classifiers learned by leaving out one fold at a time.

We will review several learning algorithms that have been used to predict PPI interactions in the next few paragraphs.

1.3.1 Decision trees:

Decision tree algorithms (Quinlan, 1986; Breiman *et al.*, 1984) are among some of the most widely used machine learning algorithms for building pattern classifiers from data. Their popularity is due in part to their ability to: select from all attributes used to describe the data, a subset of attributes that are relevant for classification; identify complex predictive relations among attributes; and produce classifiers that are easy to comprehend for humans. The ID3 (Iterative Dichotomizer 3) algorithm proposed by Quinlan (1986) and its more recent variants such as C4.5 (Quinlan, 1993) are representative for a widely used family of decision tree learning algorithms. The ID3 algorithm searches in a greedy fashion, for attributes that yield the maximum amount of information for determining the class membership of instances in a training set *D* of labeled instances. The result is a decision tree that correctly assigns each instance in *D* to its respective class. The construction of the decision tree is accomplished by recursively partitioning *D* into subsets based on values of the chosen attribute until each resulting subset has instances that belong to exactly one of the *m* classes. The selection of an attribute at each stage of construction of the decision tree maximizes the estimated expected information gained from knowing the value of the attribute in question. C4.5 (Quinlan, 1993) is the most popular variant of the ID3 algorithm that has been implemented as the J48 classifier in *WEKA*, the *Waikato Environment for Knowledge Analysis* (Witten & Frank, 2005), a popular machine learning toolkit. Some of the improvements that C4.5 has made over ID3 algorithm are: dealing with missing data, pruning the tree after creation and dealing with attributes of different costs.

1.3.2 Random Forests

Random Forest classifiers are seen to produce highly accurate results for many supervised

classification problems (Breiman, 2001). This algorithm involves the construction of multiple trees from the data. Each tree votes for the class of a new instance and the class with the maximum number of votes is chosen. The method of constructing each tree is described by Breiman (2001) in the following steps:

1. If there are *N* examples in the training set, the tree will be built by sampling *N* examples at random with replacement,
2. If there are *M* input variables, a small subset of these examples m is chosen at each node to find the best split of the data at that node, and
3. There is no pruning of the trees that are constructed at each stage.

1.3.3 Naïve Bayes:

Naïve Bayes is a highly practical learning algorithm (Mitchell, 1997), comparable to more powerful algorithms such as decision trees or neural networks in terms of performance in some domains. In the Naïve Bayes framework, each example x is described by a conjunction of attribute values, i.e. $x = < a_1, a_2, \ldots, a_n >$. The class label of an example can take any value from a finite set $C = \{c_1, c_2, \ldots, c_m\}$. The attribute values are assumed to be conditionally independent given the class label. A training set of labeled examples, $D = \{< x_1 , y_1 >, < x_2 , y_2 >, \ldots , < x_t , y_t >\}$, is presented to the algorithm. During the learning phase, a hypothesis *h* consisting of conditional and prior probabilities is learned from the training set. During the evaluation phase, the trained Naïve Bayes classifier predicts the class label of new instances *x* as follows:

1.3.4 Support Vector Machine:

The Support Vector Machine (SVM) algorithm (Vapnik, 1998; Cortes & Vapnik, 1995; Scholkopf *et al.*, 1997; Cristianini & Shawe-Taylor, 2000) is a binary classification algorithm. If the data are linearly separable, it outputs a separating hyperplane, which maximizes the "margin" between classes. If data are not linearly separable, the algorithm works by implicitly mapping the data to a higher dimensional space, where the data become separable. A maximum margin separating hyperplane is found in this space. This hyperplane in the high dimensional space corresponds to a nonlinear surface in the original space. SVM classifiers are sometimes called "large margin classifiers" because they find a maximum margin separation. Large margin classifiers are very popular due to theoretical results that show that a large margin ensures a small generalization error bound (Vapnik, 1998) and also because they proved to be very effective in practice.

1.3.5 K-Nearest Neighbors:

The K-Nearest Neighbors classifier (Cover & Hart, 1967; Mitchell, 1997) is a simple example of instance-based learning, also known as lazy learning. In the K-Nearest Neighbors algorithm, the nearest neighbors are defined in terms of a metric (e.g., Euclidean distance) *D* between instances. The class label for a new instance x is given by the most common class label among the k training examples nearest to x (according to the distance *D*).

1.3.6 Bagging:

Bootstrap aggregating or bagging (Breiman, 1996) is an algorithm that helps improve the accuracy of a classifier. Bagging works by sampling examples from the training dataset D with replacement to create subsets of the training data, which are called bootstrap samples. Classifiers are learned from the different bootstrap samples. To predict the class label of a new example, the outputs of the resulting classifiers are averaged or the classifiers are allowed to vote for the class of this new example. Bagging has been shown to avoid overfitting and to reduce the variance of learning algorithms in several domains (Breiman, 1996).

1.3.7 REPTree:

REPTree is a supervised inductive learning algorithm implemented in WEKA (Witten & Frank, 2005) that builds a decision/regression tree using information gain/variance reduction (Quinlan, 1986) as the splitting criterion. It then prunes the tree using reduced-error pruning with backfitting (Quinlan, 1993; Mitchell, 1997). Missing values are dealt with by using fractional instances as in C4.5 (Quinlan, 1993).

2. Background and significance:

Several graph-based approaches have been used to address the problem of predicting PPIs. These approaches represent the PPI network as a graph and extract relational and structural features from it. The training dataset provided to the learning algorithms consists of examples represented by such graph-based features, sometimes together with other features (e.g., amino acid composition, GO functions) associated with the protein pairs. The test dataset presented to the classifier produced by the learning algorithm contains protein pairs that are not present in the training dataset. Statistical measures such as accuracy, sensitivity, specificity and AUC score can be calculated to evaluate the performance of the learning algorithms. Several approaches to PPI prediction, including approaches based on graph-based features, are described and compared below.

2.1 *Qi, Bar-Joseph & Klein-Seetharaman (2006)*:

Qi *et al.* (2006) divide the protein interaction prediction task into three sub-tasks: (1) prediction of physical (or actual) interaction among proteins, (2) prediction of proteins belonging to the same complex and (3) prediction of proteins belonging to the same pathway. They use different data sources

for different subtasks: data from the MIPS database (Mewes *et al.*, 2002) for the first subtask, data from the DIP database (Salwinski *et al.*, 2004) for the second subtask and data from the KEGG database (Kanehisa & Goto, 2000) for the third subtask. One hundred sixty two features were constructed, grouped into seventeen distinct categories and studied to understand their effect on the protein interaction prediction subtasks. The categories that the 162 features were grouped into are:

- *Gene expression*: This category includes 20 features (each being a Pearson's correlation coefficient) calculated on 20 gene expression datasets that were recorded under more than 500 conditions (Bar-Joseph *et al.*, 2003).
- *Gene Ontology (Molecular Function, Biological Process & Cellular Component):* These three categories contain information on how many times a pair of proteins occurs in the trees (Ashburner *et al.*, 2000; Christie *et al.*, 2004).
- *Protein Expression:* Features in this category capture the difference in the expression levels for the candidate pair of proteins (Ghaemmaghami *et al.*, 2003).
- *Essentiality:* An essential protein is a protein for which deletion of the encoding gene results in a lethal phenotype, which is usually measured under laboratory conditions. The singleton feature in this category records whether the members of a pair of proteins are essential.
- *High-throughput PPI datasets (HMS_PCI, TAP & Y2H):* These three categories contain information extracted from several high-throughput protein interaction methods (Bader *et al.*, 2003; Gavin *et al.*, 2002; Ho *et al.*, 2002; Ito *et al.*, 2001; Uetz *et al.*, 2000).
- *Synthetic Lethal*: Synthetic interactions are identified if mutations in two separate genes produce a different phenotype from either gene alone, and indicate a functional association between the two genes. Two genes have a synthetic lethal relationship if mutants in either gene are viable but the double mutation is lethal. The single feature in this category was extracted by taking the union of lethality indicators from Tong *et al.* (2001) and MIPS (Mewes *et al.*, 2002).
- *Gene neighborhood/Gene Fusion/Gene Co-occurrence:* The single feature in this category is the disjunction of indicators from the three datasets described by von Mering *et al.* (2002).
- *Sequence Similarity*: The single feature in this category is a BLAST hit indicator for the query protein on the *Saccharomyces* Genome Database or SGD (Christie *et al.*, 2004).
- *Homology-based PPI*: Sequence similarity information is used to identify homology pairs. These pairs are then "BLASTed" against NCBI's non-redundant protein database and the count of their interactions extracted, resulting in four features in this category.
- *Domain-Domain Interaction*: Deng *et al.* (2002) identify domain interactions based on sequence analysis. The value of the single feature in this category is the probability of interaction of a

candidate protein pair.

- *Protein-DNA Transcription Factor (TF) group binding*: Qi *et al.* (2006) group the TFs based on the MIPS protein class catalog into 16 TF groups. For each TF group, the number of TFs that bind to both genes is found and used as one of the 16 attributes in this category.
- *MIPS features (Protein Class and Mutant Phenotype):* These 2 categories contain features that identify if the protein pair belongs to the same protein class (among 25) and mutant phenotype (among 11).

The inductive learning algorithms used in (Qi *et al.,* 2006) are: Random Forests (RF), RF similarity-based k-Nearest-Neighbor, Naive Bayes, Decision Trees (J48), Logistic Regression, and Support Vector Machines (SVM). R50, a partial AUC score, was used to evaluate the performance of the resulting classifiers. R50 is defined as the area under the ROC curve with up to 50 negative predictions. In addition to R50 being a commonly used metric in the machine learning literature, the justification provided by the authors for using this score is based on the fact that the observed frequency of interacting proteins is 1:600, resulting in an estimate of 50 protein interactions among the 30,000 selected pairs. When comparing the classifiers learned from the data using the algorithms listed above, the best relative AUC scores were obtained using RandomForest (RF) and RandomForest similarity-based k-Nearest-Neighbor (kRF) for all tasks. The maximum R50 AUC scores were 0.67 for the DIP-based direct PPI task, 0.25 for the MIPS-based co-complex PPI task, and 0.25 for the KEGG-based co-pathway PPI task. The study by Qi *et al.* also showed that the feature with the highest coverage for all three PPI subtasks was the gene coexpression, followed by the process, component, and function categories extracted from the Gene Ontology (Ashburner *et al.*, 2000).

2.2 *Licamele & Getoor (2006)*:

Licamele and Getoor (2006) combine the link structure of the PPI graph with the information about proteins in order to predict the interactions in a yeast dataset. More specifically, they look at the shared neighborhood among proteins and calculate the clustering coefficient among the neighborhoods for the first-order and second-order protein relations. The Gene Ontology distance between proteins is also considered. However, no distinction is made between direct (physical interaction) and indirect (proteins belonging to the same complex) interactions in the Licamele & Getoor (2006) study. The training set is assembled from multiple data sources such as MIPS (Mewes *et al.*, 1999), BIND (Bader *et al.*, 2001), DIP (Xenarios *et al.*, 2002), yeast two-hybrid (Ito *et al.*, 2001; Uetz *et al.*, 2000) and In vivo pull-down (Gavin *et al.*, 2002; Ho *et al.*, 2002). Classifiers such as Naive Bayes, kNN, Logistic Regression, C4.5, SVM, JRIP and Bagging with REPTrees were used on the resulting dataset. Licamele and Getoor report that the highest accuracy (81.7%) among the classifiers learned was achieved using Bagged

REPTrees and that the corresponding AUC score was 0.8967, when predicting new links from noisy high throughput data.

2.3 *Paradesi, Caragea & Hsu (2007)*:

The approaches by Qi *et al.* (2006) and by Licamele and Getoor (2006) use relational data of the PPI network along with other biologically relevant information (such as, sequence, gene expression data, GO terms, etc.) to predict the protein interactions. Paradesi *et al.* (2007) address the problem of predicting protein-protein interactions based solely on the graph features of the PPI network. They identify nine structural features for the *Saccharomyces cerevisiae* protein interaction network. These features include indegree, outdegree, number of mutual proteins and backward distance between proteins, among others.

Two datasets were used in the Paradesi *et al.* study: the DIP dataset (Salwinski *et al.*, 2004) and also the dataset generated by Qi *et al.* (2006). Similar to the other two approaches described above, Paradesi *et al.* (2007) learn several classifiers, such as Bagged Random Forest, Bagged REPTree, Random Tree, J48 and Classification via Regression from the training data. The results show that the method developed by Paradesi *et al.* (2007) compares well with the methods by Qi *et al.* and Licamele and Getoor, although no sequence information is used (as in the other two approaches), but only the relational features of the network data.

However, please note that the comparison between the approach by Paradesi *et al.* (2007) and the one by Licamele and Getoor (2006) was done using different datasets, for which the final results obtained with each method (reported in published work) were compared. Also, the method that we used for generating negative examples from the dataset provided by Qi *et al.* (2006) does not produce the same negative examples as those used in the study by Qi *et al.* (2006). Thus, the comparisons reported suffer from some data bias. Nevertheless, the complete comparisons based on previously published results are shown below:

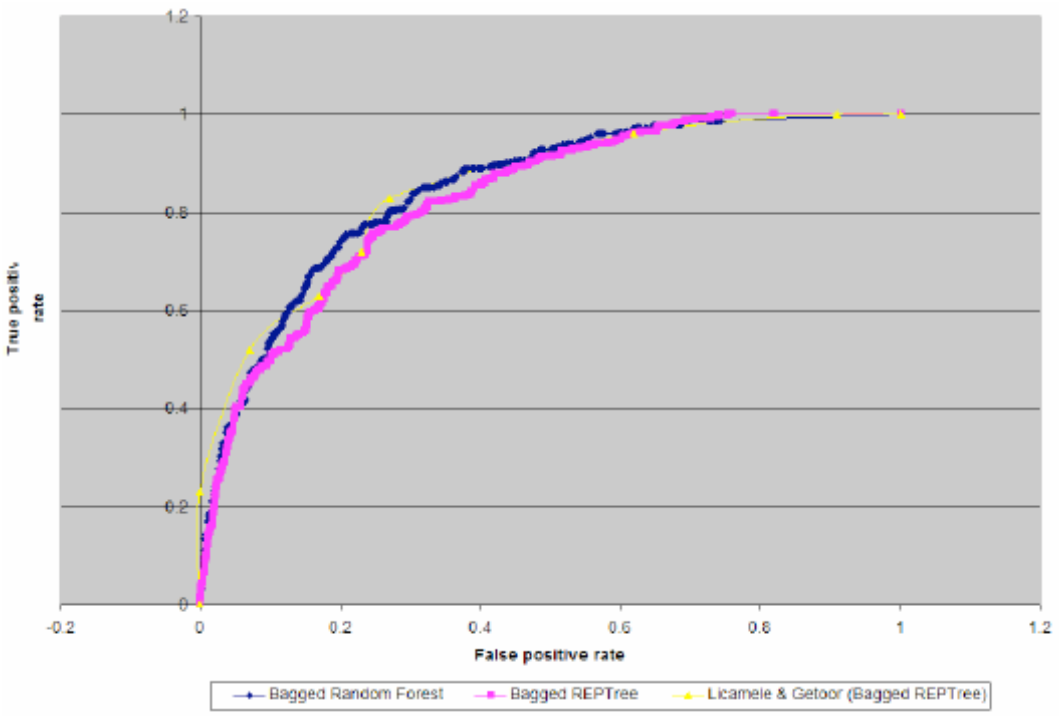

Figure 2-1 Comparison of results by Licamele and Getoor (2006) and Paradesi et al. (2007)

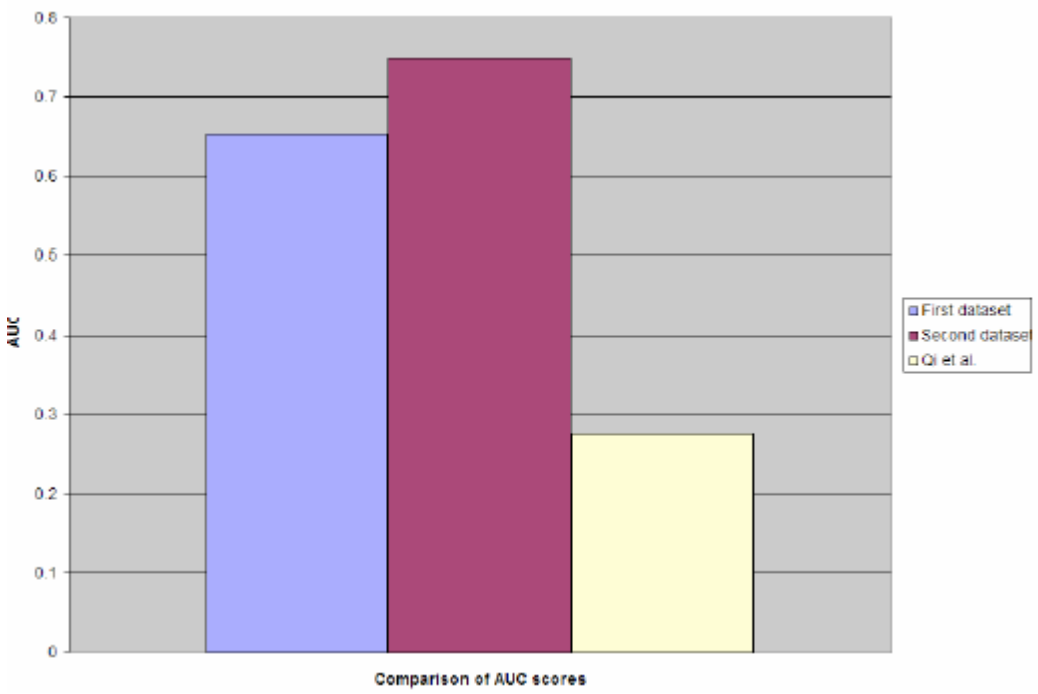

Figure 2-2 Comparison of the results by Qi et al. (2006) and those by Paradesi et al. (2007). The first dataset is DIP (Salwinski et al., 2006), while the second dataset is generated by Qi et al. (2006).

2.4 *Chen & Liu (2005)*:

Protein interaction sites refer to the locations on the protein structures where one protein physically

interacts with another protein. A protein domain is a functionally defined protein region. Chen & Liu (2005) predict PPI using protein domain information. Many domain-based models for protein interaction prediction have been developed, and preliminary results have demonstrated their feasibility (Chen & Liu, 2005). 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. Chen & Liu (2005) introduced a new framework based on random forest for PPI prediction, which explores the contributions of all the possible domain combinations to predicting protein interactions. Furthermore, their model does not assume that domain pairs are independent of each other. They obtained the PPI data from DIP (Salwinski *et al.*, 2004; Deng *et al.*, 2002; Schwikowski *et al.*, 2000; Xenarios *et al.*, 2001). Chen & Liu (2005) extract the domain information for each protein and build a vector of the domain list of each candidate protein pair. The values in the vector are given by the number of occurrences of the domain in both proteins. They obtain a sensitivity of 79.78% and specificity of 64.38%, a better result than that achieved using the MLE method of Deng *et al.* (2002), which yields a sensitivity of 78.30% and a specificity of 37.53%.

2.5 *Paradesi (2008)*:

Paradesi (2008) attempts to perform a fair comparison between methods that use only biological information and methods that use both graph features and biological information. Thus, two new sets of experiments are performed on the datasets provided by Chen and Liu (2005) and by Qi *et al.* (2006), respectively. Various features, including those originally used by Chen and Liu and by Qi *et al.*, but also the graph features used in Paradesi *et al.* (2007), are studied to indentify their importance for the accuracy of prediction. The results are presented in Tables 2-1 and 2-2, respectively, show that the graph features alone can result in very good predictive results, while the sequence features by themselves are less predictive.

Table 2-1 Results obtained from experiments using Chen & Liu (2005) dataset (5-fold cross-validation)

	J48		NB		CVR		SVM	
	Se %	Sp %	Se %	Sp %	Se %	Sp %	Se %	Sp %
Domain	73.3	62.1	73.44	63.08	55.1	0	74.3	73.56
Degree	86.62	85.74	89.78	60.66	85.78	86.98	87.2	76.02
MutualProtein	96.68	59.3	97.5	57.72	55.1	0	98.96	55.94
BackwardDistance	99.52	65.8	99.52	65.8	99.52	65.8	99.52	65.8
Domain + Degree	86.3	86.08	88.94	62.52	85.92	86.7	80.7	77.76
Domain + MutualProtein	85.14	68.4	95.64	59.08	55.1	0	78.62	75.26
Domain + BackwardDistance	91.14	72.46	83.16	71.82	99.52	65.8	86.06	82.68
Degree + MutualProtein	87.6	86.56	93.96	65.24	86.72	87.14	89.26	77.44

Degree + BackwardDistance	92.56	91.54	93.1	71.88	92.02	92.14	93.3	83.54
MutualProtein + BackwardDistance	97.86	80.04	98.18	74.42	99.54	70.88	99.26	76.16
Domain + Degree + MutualProtein	87.8	85.88	93.16	65.94	86.86	86.84	83.18	78.76
Domain + MutualProtein + BackwardDistance	96.94	81.5	97.82	76.42	99.54	70.88	90.7	85
Domain + Degree + BackwardDistance	91.86	91.96	92.3	73.06	91.96	92.24	88.7	85.58
Degree + MutualProtein + BackwardDistance	93.04	93.1	94.68	69.4	92.94	93.14	95.74	85.68
Domain + Degree + MutualProtein + BackwardDistance	93.6	92.4	94.28	70.54	92.76	93.2	91.14	87.4

Table 4-2 Results obtained from experiments using Qi et al. (2006) dataset (5-fold cross-validation)

	J48	RF	NB	CVR	SVM
Feature	0.504	0.7052	0.7244	0.7466	0.504
Degree	0.5	0.7394	0.9442	0.9798	0.5
MutualProtein	0.61	0.806	0.826	0.5	0.622
BackwardDistance	0.79	0.79	0.79	0.5	0.73
Feature + Degree	0.57	0.7276	0.7378	0.9526	0.506
Feature + MutualProtein	0.5944	0.8172	0.737	0.8347	0.626
Feature + BackwardDistance	0.796	0.8438	0.7374	0.8634	0.736
Degree + MutualProtein	0.7052	0.8604	0.9632	0.9828	0.624
Degree + BackwardDistance	0.79	0.833	0.9646	0.9896	0.73
MutualProtein + BackwardDistance	0.79	0.94	0.948	0.5	0.758
Feature + Degree + MutualProtein	0.5784	0.8358	0.7428	0.9496	0.626
Feature + MutualProtein + BackwardDistance	0.796	0.9408	0.7438	0.9052	0.758
Feature + Degree + BackwardDistance	0.796	0.8678	0.7428	0.9768	0.736
Degree + MutualProtein + BackwardDistance	0.79	0.944	0.9774	0.9926	0.758
Feature + Degree + MutualProtein + BackwardDistance	0.796	0.927	0.7468	0.98	0.756

3. Issues and Problems:

There are many methods of inferring protein interactions, some of them described above. The goal of this chapter was to show that methods that use graph-based features from the PPI network, sometimes along with the biological features of the interacting proteins, can produce results better than those using the biological features alone. However, the task of predicting protein interactions using machine learning approaches is far from being solved completely. The machine learning algorithms are obviously not 100% accurate. In particular, the graph-based approaches advocated in the chapter are faced with several problems and issues discussed in this section.

3.1 Combining data from multiple databases:

There are many databases that provide information about protein interactions. However, different databases store protein interactions from different high-throughput interaction detection methods. This results in very few protein interactions that are contained in several databases. Qi *et al.* (2006) states that there are only 293 protein interactions present in all DIP (Salwinski *et al.*, 2004), MIPS (Mewes *et al.*, 2002) and KEGG (Kanehisa & Goto, 2000) databases. Moreover, different databases store different attributes for protein interactions. Let us consider the data available in all databases as published by their respective research groups. The BIND (Bader *et al.*, 2003) database stores information about interactions, complexes and pathways. BIND (Bader *et al.*, 2003) contains a number of large-scale interactions and complex mapping experiments using yeast two-hybrid, mass spectrometry, genetic interactions and phage display. DIP (Salwinski *et al.*, 2004) develops methods of quality assessment and uses them to identify the most reliable subset of the interactions that are inferred from high-throughput experiments. MIPS (Mewes *et al.*, 2002) provides information on Open Reading Frames (ORFs), RNA genes and other genetic elements. It also contains information on how proteins cooperate in complexes, pathways and cellular networks. In addition, detailed information on transcription factors and their binding sites, transport proteins and metabolic pathways are being included or interlinked to the core data. IntAct (Kerrien *et al.*, 2007) manually annotates published manuscripts reporting molecular interaction data and formalizing it by using a comprehensive set of controlled vocabularies in order to ensure data integrity. IntAct (Kerrien *et al.*, 2007) is probably the only database that contains negative examples of protein interactions. MINT (Chatr-aryamontri *et al.*, 2007) focuses on the curation work on physical interactions between proteins. Genetic or computationally inferred interactions are not included in the database. However, von Mering *et al.*(2002) states that there are very few protein interactions that are supported by one of the high-throughput methods. Given the diversity of the protein interactions in different databases and also the diversity of the features, it is very difficult to assemble a comprehensive training dataset that contains all the known protein interactions, as well as all the biological and structural features that can be defined for a pair of interactions, thus making it difficult to use all the existing information to learn more accurate prediction models.

3.2 Negative protein interactions:

One of the most common problems that researchers face while predicting protein interactions using computational methods is to deal with the large number of negative examples. Let us assume that there are around 6,000 proteins in a database and around 80,000 interactions between them are known. This means that there are still 6000^2 – 80,000 = 35,920,000 interactions that are yet to be classified as

true positive or true negative. If we provide these 80,000 positive examples and 35,920,000 negative examples to a random algorithm that predicts that any two given proteins do not interact, we will still obtain an accuracy of 99.78%. Although, this is a good result, it is important that a machine learning algorithm must predict the true positives accurately. We are more interested in the problem of predicting which proteins interact than in that of predicting which proteins do not interact. In order to solve our desired problem, we must find ways of reducing the number of negative examples that will be provided to the algorithms. Qi *et al.* (2006) randomly select protein pairs that do not interact as negative examples. Licamele and Getoor (2006) randomly sample negative protein pairs without replacement. They choose an equal number of positive protein pairs and negative protein pairs. Chen & Liu (2005) also sample negative protein pairs that are roughly equal to the number of positive protein pairs. Paradesi *et al.* (2007) perform a depth-limited Breadth First Search (with depth 2) from each protein and generate protein pairs. By using this technique, they obtain all positive interactions and several negative interactions. All of the above-mentioned techniques of generating negative examples are not accurate because they assume that any protein pair that is not present in the positive interaction list constitutes a negative example. A protein pair must be labeled as a negative interaction if and only if it is experimentally determined that those two proteins do not interact in any high-throughput methods.

3.3 False positives in high-throughput methods:

There are several high-throughput methods for detecting protein-protein interactions, including yeast two-hybrid method (Ito *et al.*, 2001; Uetz *et al.*, 2000), analysis of protein complexes using mass spectrometry (Gavin *et al.*, 2002; Ho *et al.*, 2002), co-immunoprecipitation (Sambrook *et al.*, 2006), etc. Some of the interactions that are identified from different high-throughput methods may be false positives. Many researchers have tried to assess the quality of the high-throughput data (von Mering *et al.*, 2002; Mrowka *et al.*, 2001; Deane *et al.*, 2002). However, they claim that sometimes a subset of interactions that were identified by using one method could not be identified by another method. These interactions are called false positives because they might not have been interactions but were wrongly labeled as interactions by the high-throughput method (Salwinski *et al.*, 2003).

3.4 Representation of data:

There is a need to handle appropriately data from different interaction-detection methods. For example, the yeast two-hybrid screening method, which is one of the most popular methods of detecting protein interactions, uses a bait-and-prey approach to find interactions in yeast (Young, 1998). Furthermore, several databases provide the protein interaction data in the form of different interacting “bait” proteins and “prey” proteins. However, some researchers still treat the protein

interaction network from the yeast two-hybrid screening method as an undirected network. In general, different high-throughput techniques have different interacting relationships (either directed or undirected) between interacting protein pairs.

3.5 Validity of predicted protein interaction:

It might be easier and faster to predict or identify new protein interactions using the graph-based features of the PPI network. However, the newly discovered protein interaction is accurate only from a relational learning (or graph-mining) perspective. There is no guarantee that the new interacting protein pair is biologically valid. The new protein interaction could either be a true positive interaction or a false positive interaction. Moreover, there are protein pairs that are identified as interacting pairs because of the graph-based prediction and a single high-throughput method. These protein pairs are most likely false positive protein interactions because they have been identified only through a single high-throughput protein interaction detection method. Therefore, there is a need to verify the validity of the interactions discovered through the graph-based prediction approach.

3.6 Identification of useful graph-based features:

There are many features that can be extracted from a PPI network as explained by the previous approaches of predicting protein interactions. However, not all features are useful in the prediction task. In fact, some features may actually hurt the learning process of the classifier and thereby lower the accuracy of the results. Qi *et al.* (2006) have discovered that features that might be useful for solving an interaction subtask may not be useful for solving another interaction subtask. The authors have also observed that different combinations of various features affect the performance of the learning algorithms.

4. Suggestions and Recommendations:

In this section, we provide possible solutions and recommendations to the problems and issues presented above.

4.1 Combining data from multiple databases:

Gathering more data and features can provide a more tractable representation of the training data, increasing the achievable accuracy, sensitivity, and specificity of the learning system. Given the nature of the data in the PPI databases, there are two different methods we could use to combine data from multiple sources. The first method is to take the intersection across all databases. This would result in a small number of protein interactions but many features about those interactions. The second method is to take the union across all databases. This would result in a dataset with many protein interactions but also many missing attributes. Thus, there is a tradeoff between data with a small

number of protein interactions but good quality information about these interactions and data with a large number of protein interactions but a lot of missing information about the interactions. This tradeoff suggests that there is a need to efficiently unify information across all databases by the research groups that maintain these databases. To address this need, the research groups that maintain BIND (Bader *et al.*, 2003), MINT (Chatr-aryamontri *et al.*, 2007), DIP (Salwinski *et al.*, 2004), MPact (Güldener *et al.*, 2006) and IntAct (Kerrien *et al.*, 2007) have formed the IMEx consortium to build a large, consistent and non-redundant repository of protein interactions and information about the interactions. According to the IMEx consortium, the data gathered by them will be broader in scope and deeper in information than any individual effort (Kerrien *et al.*, 2007). By integrating multiple data sources, the PPI network will be more complete than before and will enable researchers to achieve better quality graph-based feature extraction.

4.2 Negative protein interactions:

A better method of selecting non-interacting protein pairs (i.e., negative examples) needs to be developed. One possible method for generating negative protein pairs would be to mark protein pairs that do not belong to the same complex or the same pathway as negative, if there is no known interaction between them. Another method for generating negative examples is similar to the previous method, but could be achieved by extracting information from a PPI network. If the proteins in a pair belong to different cliques that are apart by a very large distance in the network and they do not have any known interaction between them, they could be considered to form a negative protein pair. Also, if we cluster the proteins in a network based on complexes and choose a protein pair from different clusters, that pair could be considered to be a negative example. It would help to use the above-mentioned techniques on protein interaction visualization tools such as ProViz (Iragne *et al.*, 2005), iPfam (Finn *et al.*, 2005), VisANT (Hu *et al.*, 2007), etc. and querying tools such as PathBLAST (Kelley *et al.*, 2004), APID (Prieto & De Las Rivas, 2006), etc. to detect negative protein interactions.

4.3 False positives in high-throughput methods:

DIP provides several tests, such as Expression Profile Reliability Index (EPR Index) (Deane *et al.*, 2002), Paralogous Verification (PVM) (Deane *et al.*, 2002) and Domain Pair Verification (DPV) (Deng *et al.*, 2002) as online services, to ensure that the number of false positives is reduced in any given dataset. A graph-based approach to reduce the number of false positives obtained from computational techniques would be to build separate graphs for each of the high-throughput methods. By taking the intersection of all the graphs we could obtain the true positive protein interactions in the simplest case. In a more complex case, we could use protein visualization tools such as ProViz (Iragne *et al.*, 2005), iPfam (Finn *et al.*, 2005), VisANT (Hu *et al.*, 2007), etc. to identify proteins that

frequently occur in most of the graphs and infer actual protein interactions among them.

4.4 Representation of data:

The protein interaction network must be treated as a directed network especially when using yeast two-hybrid screening data in order to enable the classification algorithm to learn using bait and prey proteins. More importantly, there needs to be a way of dealing with representation of data collected from several high-throughput methods. In other words, if the experiment proves that the interaction occurs in one way from one protein to another, the interaction must be treated as a directed edge; otherwise, the interaction must be treated as an undirected edge. In a few cases, there might be protein interactions that require the above-mentioned data representation. In order to solve a protein interaction prediction problem that involves data with different representations, the problems must be divided into minimal sub-problems that can be solved using the desired representation and learning algorithm. The results from the sub-problems must be combined using either a weighting system or a voting technique, and the protein pair must be classified as interacting or not.

4.5 Validity of predicted protein interaction:

We could label each positive interacting protein pair identified by the graph-based interaction prediction method as a true positive example. However, most probably the results from the machine learning graph-based approach are not 100% accurate when the learned classifier is used to classify new data in the test dataset. One of the techniques of validating newly discovered protein interactions is to perform a high-throughput protein interaction detection method to experimentally validate if the protein pair interacts or not. If a protein pair has been identified as an interacting pair by only one high-throughput method, then the result of the graph-based prediction can help confirm the protein pair as a true positive interaction. In other words, the results of graph-based PPI prediction algorithms can be used along with the high-throughput interaction detection methods to identify true positive interactions. Therefore, graph-based interaction prediction methods can serve a two-fold purpose – to identify new interactions and to strengthen our confidence in the results of existing high-throughput methods.

4.6 Identification of useful graph-based features:

It is important to use only features that help in the learning process. One method by which this can be achieved is by running classification algorithms on each feature individually and selecting only those features that provide higher accuracy. Another method of identifying important features is by applying dection on the input data. There are several statistical measures to identify the importance of features such as Entropy, Gini index, etc. It is also important to identify the right combination of features to better predict protein interactions. This can be achieved by performing several experiments with various combinations of features. It is often computationally difficult to perform many

experiments with thousands of features to detect the right combinations of features. A better solution would be to first calculate the most useful features, thereby reducing the size of the feature list, and then perform experiments by varying the combinations of the selected features. It has been observed by the authors that by choosing the important features and reducing the size of the feature list, one may not achieve the highest accuracy possible. In other words, a comprehensive examination of all features will provide the highest accuracy for some combination of features, while an examination of a reduced set of features will be computationally less-intensive. Thus, there is a trade-off between the number of features chosen and the desired accuracy.

5. Future trends:

The future trends in the field of predicting protein interactions using graph-based features look very promising due to the following changes in this area:

5.1 Increase in quality and quantity of data:

There is a rapid increase in the discovery of new proteins and interactions based on high-throughput methods. There is also a growth in techniques to identify actual protein interactions and eliminate false positive interactions. The IMEx consortium, as mentioned previously in this chapter, is allowing individual researchers and research groups to submit protein interaction information. This newly submitted data is run through several tests and manually curated to ensure that the interaction is a true positive interaction. As the quality and quantity of data increases, the PPI network becomes more complete, thereby allowing more complete features to be extracted from the network. There is scope for researchers to work on increasing the quality and quantity of protein interactions by developing new computational techniques.

5.2 Improvement in classification algorithms:

There are also rapid advances in the machine learning and data mining algorithms. Most of the supervised and unsupervised learning algorithms used in the prediction of protein interactions were developed for tasks other than that. Although it has been observed that these algorithms have worked well for the protein interaction prediction task, there is a need for developing custom algorithms that can handle protein interaction data even better.

5.3 Use of protein interaction network analysis tools:

There are many protein interaction visualization tools such as ProViz (Iragne *et al.*, 2005), iPfam (Finn *et al.*, 2005), VisANT (Hu *et al.*, 2007), etc. and querying tools such as PathBLAST (Kelley *et al.*, 2004), APID (Prieto & De Las Rivas, 2006), etc. available. These tools allow users to view and search for proteins in any PPI network. They can be exploited to gather several graph-based

features from the PPI network. The visualization and querying tools can also be used to split the PPI network into several overlapping sub-graphs. Interactions can be predicted at the sub-graph level and these predictions can be combined to identify protein interactions at the original graph level. A protein pair can be labeled as interacting if it is observed that the interaction between the protein pair appears in more than one sub-graph.

5.4 Development of different approaches:

Although there is an increase in data and improvement of algorithms and tools, attention must be paid to improve the approaches of solving the protein interactions prediction problem. There is no one-solution-solves-all-problems approach anymore. Instead, there is a need for developing approaches that solve the problem by applying an ensemble of various machine learning algorithms for different subgraphs of the PPI network. In other words, one could extract different graph-based features from different subsets of the PPI network and run different machine learning algorithms on the features, depending on the data. The different machine learning classifiers could "vote" on the class, and the weighted average of the output could be assigned as the actual class.

6. Conclusion:

In this chapter, we provide an overview of several methods for predicting protein interactions using biological and graph features extracted from a PPI network. Thus, machine learning algorithms that can be used to predict protein interactions based on biological and graph-based features are described, along with some specific methods that make use of such algorithms and features. Furthermore, several issues and open problems in the PPI prediction area are presented together with suggestions and recommendations for how to deal with them. At last, important future trends are highlighted.

7. References:

- Altman, D.G., Bland, J.M. (1994). Diagnostic tests. 1: Sensitivity and specificity. *BMJ*, 308(6943):1552.
- Albert, I., & Albert, R. (2004). Conserved network motifs allow protein-protein interaction prediction. *Bioinformatics*. 20(18):3346-52.
- Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet.*25(1):25-9.
- Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F., Pawson, T., & Hogue, C.W. (2001).

BIND--The Biomolecular Interaction Network Database. *Nucleic Acids Res*, 29(1):242-5.

- Bader, G.D., & Hogue, C.W. (2002). Analyzing yeast protein-protein interaction data obtained from different sources. *Nat Biotechnol.* 20(10):991-7.
- Bader, G.D., Betel, D., & Hogue, C.W. (2003) BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Res*, 31(1):248-50.
- Bar-Joseph, Z., Gerber, G.K., Lee, T.I., Rinaldi, N.J., Yoo, J.Y., Robert, F., Gordon, D.B., Fraenkel, E., Jaakkola, T.S., Young, R.A., Gifford, D.K. (2003). Computational discovery of gene modules and regulatory networks. *Nat Biotechnol.* 21(11):1337-42.
- Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L., Studholme, D.J., Yeats, C., Eddy, S.R. (2004). The Pfam protein families database. *Nucleic Acids Res*;32(Database issue):D138-41.
- Breiman, L., Freidman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24 (2): 123-40.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45 (1), 5-32.
- Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., Lu, H., Zhang, J., Sun, S., Ling, L., Zhang, N., Li, G., & Chen, R. (2003). Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Res*, 31(9):2443-50.
- Chatr-aryamontri, A., Ceol, A., Palazzi, L.M., Nardelli, G., Schneider, M.V., Castagnoli, L., & Cesareni, G. (2007). MINT: the Molecular INTeraction database. *Nucleic Acids Res*, 35(Database issue): D572–D574.
- Chen, X.W., & Liu, M. (2005). Prediction of protein-protein interactions using random decision forest framework, *Bioinformatics*, 21(24):4394-4400.
- Christie, K.R., Weng, S., Balakrishnan, R., Costanzo, M.C., Dolinski, K., Dwight, S.S., Engel, S.R., Feierbach, B., Fisk, D.G., Hirschman, J.E., Hong, E.L., Issel-Tarver, L., Nash, R., Sethuraman, A., Starr, B., Theesfeld, C.L., Andrada, R., Binkley, G., Dong, Q., Lane, C., Schroeder, M., Botstein, D., Cherry, J.M. (2004). Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from *Saccharomyces cerevisiae* and related sequences from other organisms. *Nucleic Acids Res*. 32(Database issue):D311-4.
- Cortes, C. & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3):273-297.
- Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, Vol. 13, No. 1, pp. 21-27.
- Cristianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge

University.

- Deane, C.M., Salwinski, L., Xenarios, I. & Eisenberg, D. (2002). Protein interactions: two methods for assessment of the reliability of high throughput observations. *Mol Cell Proteomics*, 1(5):349-56.
- Deng, M., Mehta, S., Sun, F., & Chen, T. (2002). Inferring domain-domain interactions from protein-protein interactions. *Genome Res*, 12(10):1540-8.
- Fawcett T. (2004). *ROC Graphs: Notes and Practical Considerations for Researchers*. Technical report, Palo Alto, USA: HP Laboratories.
- Fields, S., Song, O. (1989). A novel genetic system to detect protein-protein interactions. *Nature*; 340(6230):245-6.
- Finn, R.D., Marshall, M., & Bateman, A. (2005). iPfam: visualization of protein-protein interactions in PDB at domain and amino acid resolutions. *Bioinformatics*, 21(3):410-2.
- Frank, E., Wang, Y., Holmes, G., & Witten, I.H. (1998). Using model trees for classification. *Machine Learning*, 32 (1), 63-76.
- Gavin, A.C., Bösche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J.M., Michon, A.M., Cruciat, C.M., Remor, M., Höfert, C., Schelder, M., Brajenovic, M., Ruffner, H., Merino, A., Klein, K., Hudak, M., Dickson, D., Rudi, T., Gnau, V., Bauch, A., Bastuck, S., Huhse, B., Leutwein, C., Heurtier, M.A., Copley, R.R., Edelmann, A., Querfurth, E., Rybin, V., Drewes, G., Raida, M., Bouwmeester, T., Bork, P., Seraphin, B., Kuster, B., Neubauer, G., & Superti-Furga, G. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141-7.
- Ghaemmaghami, S., Huh, W.K., Bower, K., Howson, R.W., Belle, A., Dephoure, N., O'Shea, E.K., Weissman, J.S. (2003). Global analysis of protein expression in yeast. *Nature*, 425(6959):737-41.
- Güldener, U., Münsterkötter, M., Oesterheld, M., Pagel, P., Ruepp, A., Mewes, H.W., Stümpflen, V. (2006). MPact: the MIPS protein interaction resource on yeast. *Nucleic Acids Res*, 34(Database issue):D436-41.
- Harbison, C.T., Gordon, D.B., Lee, T.I., Rinaldi, N.J., Macisaac, K.D., Danford, T.W., Hannett, N.M., Tagne, J.B., Reynolds, D.B., Yoo, J., Jennings, E.G., Zeitlinger, J., Pokholok, D.K., Kellis, M., Rolfe, P.A., Takusagawa, K.T., Lander, E.S., Gifford, D.K., Fraenkel, E., Young, R.A. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99-104.
- Ho, Y., Gruhler, A., Heilbut, A., Bader, G.D., Moore, L., Adams, S.L., Millar, A., Taylor, P., Bennett, K., Boutilier, K., Yang, L., Wolting, C., Donaldson, I., Schandorff, S., Shewnarane, J., Vo,

M., Taggart, J., Goudreault, M., Muskat, B., Alfarano, C., Dewar, D., Lin, Z., Michalickova, K., Willems, A.R., Sassi, H., Nielsen, P.A., Rasmussen, K.J., Andersen, J.R., Johansen, L.E., Hansen, L.H., Jespersen, H., Podtelejnikov, A., Nielsen, E., Crawford, J., Poulsen, V., Sørensen, B.D., Matthiesen, J., Hendrickson, R.C., Gleeson, F., Pawson, T., Moran, M.F., Durocher, D., Mann, M., Hogue, C.W., Figeys, D., & Tyers, M. (2002). Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. *Nature*, 415(6868):180-3.

- Hu, Z., Ng, D.M., Yamada, T., Chen, C., Kawashima, S., Mellor, J., Linghu, B., Kanehisa, M., Stuart, J.M., & DeLisi, C. (2007). VisANT 3.0: new modules for pathway visualization, editing, prediction and construction. *Nucleic Acids Res*, 35(Web Server issue):W625-32.
- Huh, W.K., Falvo, J.V., Gerke, L.C., Carroll, A.S., Howson, R.W., Weissman, J.S., O'Shea, E.K. (2003). Global analysis of protein localization in budding yeast. *Nature*, 425(6959):686-91.
- Hsu, W.H., King, A.L., Paradesi, M.S.R., Pydimarri, T., & Weninger, T. (2006). Collaborative and Structural Recommendation of Friends using Weblog-based Social Network Analysis, *Proc. of Computational Approaches to Analyzing Weblogs* - AAAI 2006 Technical Report SS-06-03, 55-60.
- Iragne, F., Nikolski, M., Mathieu, B., Auber, D., & Sherman, D. (2005). ProViz: protein interaction visualization and exploration. *Bioinformatics*, 21(2):272-4.
- Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., & Sakaki, Y. (2001). A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci U S A.*, 98(8):4569-74.
- Kanehisa, M. & Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res,* 28, 27-30.
- Kelley, B.P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B.R., Ideker, T. (2004). PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Res*, 32(Web Server issue):W83-8.
- Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., Kohler, C., Khadake, J., Leroy, C., Liban, A., Lieftink, C., Montecchi-Palazzi, L., Orchard, S., Risse, J., Robbe, K., Roechert, B., Thorneycroft, D., Zhang, Y., Apweiler, R., & Hermjakob, H. (2007). IntAct--open source resource for molecular interaction data. *Nucleic Acids Res*, 35(Database issue):D561-5.
- Licamele, K., & Getoor, L. (2006). Predicting Protein-Protein Interactions Using Relational Features, *Proc. of ICML Workshop on Statistical Network Analysis*.
- Liu, H. & Motoda, H. (1998). Feature Selection for Knowledge Discovery and Data Mining. Kluwer.
- MacBeath, G., & Schreiber, S.L. (2000). Printing proteins as microarrays for high-throughput

function determination. *Science*; 289(5485):1760-3.

- Maslov, S., & Sneppen, K. (2002). Specificity and stability in topology of protein networks, *Science*, 296(5569):910-3.
- Mewes, H. W., Heumann, K., Kaps, A., Mayer, K., Pfeiffer, F., Stocker, S., & Frishman, D. (1999). MIPS: a database for genomes and protein sequences. *Nucleic Acids Res*, 27, 44–48.
- Mewes, H.W., Frishman, D., Güldener, U., Mannhaupt, G., Mayer, K., Mokrejs, M., Morgenstern, B., Münsterkötter, M., Rudd, S., & Weil, B. (2002). MIPS: a database for genomes and protein sequences. *Nucleic Acids Res*, 30(1):31-4.
- Mitchell T. (1997). *Machine Learning*, McGraw Hill.
- Mrowka, R., Patzak, A. & Herzel, H. (2001). Is there a bias in proteome research? *Genome Res.*, 11, 1971-73.
- Najafabadi, H.S, & Salavati, R. (2008). Sequence-based prediction of protein-protein interactions by means of codon usage. *Genome Biology*, 9:R87
- Paradesi, M.S.R. (2008). Graph-Based Protein-Protein Interaction Prediction in *Saccharomyces Cerevisiae*. Master's thesis, Kansas State University, Department of Computing and Information Sciences.
- Paradesi, M.S.R., Caragea, D., & Hsu, W.H. (2007). Structural Prediction of Protein-Protein Interactions in *Saccharomyces cerevisiae*, *Proc. of IEEE 7th International Symposium on BioInformatics and BioEngineering*, vol. 2, pp. 1270-1274.
- Paradesi, M.S.R., Wang, L., Brown, S.J., & Hsu, W.H. (2006). Mining Domain Association Rules From Protein-Protein Interaction data, *Intelligent Engineering Systems through Artificial Neural Networks*, vol. 16, 213-218.
- Prieto ,C., & De Las Rivas, J. (2006). APID: Agile Protein Interaction DataAnalyzer. *Nucleic Acids Res*, 34(Web Server issue):W298-302.
- Qi, Y., Bar-Joseph, Z., & Klein-Seetharaman, J., (2006). Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins: Structure, Function, and Bioinformatics*, Volume 63, Issue 3, 490-500.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81-106.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Salwinski, L. & Eisenberg, D. (2003). Computational methods of analysis of protein-protein interactions. *Curr. Opin. Struct. Biol.*, 13, 377-382.
- Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., & Eisenberg, D. (2004). The Database of Interacting Proteins: 2004 update. *Nucleic Acids Res*, 32(Database issue):D449-51.

- Sambrook, J. & Russell, D.W. (2006). Identification of Associated Proteins by Coimmunoprecipitation, *CSH Protocols*, doi:10.1101/pdb.prot3898.
- Schwikowski, B., Uetz, P., Fields, S. (2000). A network of protein-protein interactions in yeast. *Nat Biotechnol*;18(12):1257-61.
- Scholkopf, B., Sung, K.-K., Burges, C. J. C., Girosi, F., Niyogi, P., Poggio, T., & Vapnik, V. (1997). Comparing support vector machines with gaussian kernels to radial basis function classiers, *IEEE Trans. on Signal Processing*, vol. 45, no. 11, pp. 2758-65.
- Shoemaker, B.A., & Panchenko, A.R. (2007). Deciphering Protein–Protein Interactions. Part I. Experimental Techniques and Databases. *PLoS Comput Biol* 3(3): e42.
- Shoemaker, B.A., & Panchenko, A.R. (2007). Deciphering Protein–Protein Interactions. Part II. Computational Methods to Predict Protein and Domain Interaction Partners. *PLoS Comput Biol* 3(4): e43.
- Tong, A.H., Evangelista, M., Parsons, A.B., Xu, H., Bader, G.D., Pagé, N., Robinson, M., Raghibizadeh, S., Hogue, C.W., Bussey, H., Andrews, B., Tyers, M., Boone, C. (2001). Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, 294(5550):2364-8.
- Tong, A.H., Lesage, G., Bader, G.D., Ding, H., Xu, H., Xin, X., Young, J., Berriz, G.F., Brost, R.L., Chang, M., Chen, Y., Cheng, X., Chua, G., Friesen, H., Goldberg, D.S., Haynes, J., Humphries, C., He, G., Hussein, S., Ke, L., Krogan, N., Li, Z., Levinson, J.N., Lu, H., Ménard, P., Munyana, C., Parsons, A.B., Ryan, O., Tonikian, R., Roberts, T., Sdicu, A.M., Shapiro, J., Sheikh, B., Suter, B., Wong, S.L., Zhang, L.V., Zhu, H., Burd, C.G., Munro, S., Sander, C., Rine, J., Greenblatt, J., Peter, M., Bretscher, A., Bell, G., Roth, F.P., Brown, G.W., Andrews, B., Bussey, H., Boone, C. (2004). Global mapping of the yeast genetic interaction network. *Science*, 303(5659):808-13.
- Uetz, P., Giot, L., Cagney, G., Mansfield, T.A., Judson, R.S., Knight, J.R., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P., Qureshi-Emili, A., Li, Y., Godwin, B., Conover, D., Kalbfleisch, T., Vijayadamodar, G., Yang, M., Johnston, M., Fields, S., & Rothberg, J.M. (2000). A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, 403(6770):623-7.
- Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.
- von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S.G., Fields, S. & Bork, P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417, 399-403.
- Witten, I.H. & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*, 2nd Edition. Morgan Kaufmann.

- Xenarios, I., Salwnski, L., Duan, X. J., Higney, P., Kim, S. M., & Eisenberg, D. (2002). DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*, 30, 303–5.
- Young, K.H. (1998). Yeast two-hybrid: so many interactions, (in) so little time... *Biol Reprod*, 58(2):302-11.