Structural Prediction of Protein-Protein Interactions in Saccharomyces cerevisiae

Martin S.R. Paradesi, Doina Caragea, William H. Hsu
Department of Computing and Information Sciences, Kansas State University
Manhattan, KS 66506 USA
{pmsr, dcaragea, bhsu}@cis.ksu.edu

***Abstract*—Protein-protein interactions (PPI) refer to the associations between proteins and the study of these associations. Several approaches have been used to address the problem of predicting PPI. Some of them are based on biological features extracted from a protein sequence (such as, amino acid composition, GO terms, etc.); others use relational and structural features extracted from the PPI network, which can be represented as a graph. Our approach falls in the second category. We adapt a general approach to graph feature extraction that has previously been applied to collaborative recommendation of friends in social networks. Several structural features are identified based on the PPI graph and used to learn classifiers for predicting new interactions. Two datasets containing *Saccharomyces cerevisiae* PPI are used to test the proposed approach. Both these datasets were assembled from the Database of Interacting Proteins (DIP). We assembled the first data set directly from DIP in April 2006, while the second data set has been used in previous studies, thus making it easy to compare our approach with previous approaches. Several classifiers are trained using the structural features extracted from the interactions graph. The results show good performance (accuracy, sensitivity and specificity), proving that the structural features are highly predictive with respect to PPI.**

Keywords: protein-protein interaction, graph mining, machine learning

I. INTRODUCTION

Protein-protein interactions (PPI) play an important role in the study of biological processes. Many PPI have been discovered over the years and several databases have been created to store the information about these interactions (e.g. BIND, DIP, MIPS, IntAct, MINT and MIPS). Mering *et al.*[8] state that about 80,000 interactions between yeast proteins are currently available from various high-throughput interaction-detection methods. Determining PPI using high-throughput methods is not only expensive and time-consuming, but also generates a high number of false positives and false negatives. Therefore, there is a need for computational approaches that can help in the process of identifying real protein interactions. From a machine learning point of view, this problem can be seen as a binary classification problem, and can be addressed using supervised learning algorithms. In this paper, we use a graph mining approach to predict the existence of a PPI in a network of interacting proteins.

II. PREVIOUS WORK

Several methods have been designed to address the task of predicting protein-protein interactions. Most of them [1], [2], [10] and [12] use features extracted from protein sequences (e.g., amino acids 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. When using the PPI network to design features, several node and topological features can be extracted directly from the associated graph.

Qi *et al.*[8] 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 apply several feature classifiers on the prediction tasks considered. Their results show that RandomForest is the one of the top two classifiers for all tasks; the other one is RandomForest similarity-based k-Nearest-Neighbor.

Licamele & Getoor[4] combine the link structure of the PPI graph with the information about proteins in order to predict the interactions in a yeast dataset, gathered from several databases. 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. They obtained reasonably good accuracy of 81% when predicting new links from noisy high throughput data.

The abovementioned approaches 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. However, as opposed to these approaches, we use only the relational features of the PPI network data in our study.

III. OUR APPROACH

In related work, but a completely different application domain, Hsu *et al.*[3] address the problem of collaboratively recommending friends for a person, based on structural features extracted from a given social network graph. Their approach to the collaborative recommendation of friends uses the link structure of the social network and also information about mutually declared interests. They use structural features

(of individual vertices or of the links) to learn classifiers that can be used to predict possible but unknown links (u, v) in the LiveJournal social network. The experimental results show that their system differentiates friends from non-friends in a connected group of users with greater accuracy than the recommender system that is currently used by LiveJournal.

Noticing the similarity between the friends recommendation problem and the protein-protein interaction prediction problem (i.e., proteins can be associated with users and interactions can be regarded as friendship relationships), in this paper, we explore the approach used in Hsu *et al.*[3] in the context of a protein "friends" recommendation, that was previously explored in both Qi *et al.*[8] and Licamele & Getoor[4].

Nine relational features (such as the indegree and outdegree of the proteins in the graph, mutual "friends" among proteins and backward distance between proteins in the graph) are extracted from the PPI network using graph mining techniques described by Hsu *et al.*[3]. As opposed to previous approaches, we don't use any features based directly on sequence or GO information. Our results show that the structural features inferred from the graph can be highly predictive with respect to PPI prediction. They compare favorably with the results reported by Licamele & Getoor[4] in terms of accuracy, and also with the results reported by Qi *et al.*[8] in terms of AUC scores. We also explore the relative importance of the features used. The results confirm the previous findings reported in Hsu *et al.*[3] that graph features are useful in recommending friends to users in a network.

IV. Experiment Design

A. Dataset

We used two different datasets to evaluate our approach experimentally. Both datasets contain yeast data. The yeast organism was chosen primarily because there is more information about yeast protein interactions than about any other organism. The first PPI dataset of budding yeast (*Saccharomyces cerevisiae*) was retrieved from the Database of Interacting Proteins (DIP) database in April 2006 (using a procedure similar to the one described in Salwinski *et al.*[11]). It consists of 2554 different proteins and 5952 interactions between protein pairs. The second dataset of yeast is similar to the one used by Qi *et al.*[8] and consists of the positive interactions retrieved from DIP during September-October 2004. It contains 1536 different proteins and 2865 interacting pairs. The datasets were parsed in order to construct directed networks of interacting protein pairs. We adopt the approach in Maslov & Sneppen[5] and represent the PPI network as a directed graph with a directed edge from a "bait" protein to a "prey" protein. We draw a link between two proteins if and only if there exists an interaction between those two proteins. The absence of an interaction between two proteins results in not adding a link between those two proteins in the graph structure.

B. Feature Analyzers

We perform a depth-limited breadth-first search exhaustively at each node (protein) in the graph (within a depth of 2) and generate candidate edges between proteins. Each example in the PPI dataset defines a candidate edge (u, v) in the underlying directed graph of the protein-protein interaction network. The classification problem reduces to the problem of classifying proteins within a distance d(u, v) as either 1 (interacting) or 2 (non-interacting). The following features are considered for each candidate edge in the network:

1. Indegree of the start node: Denotes the popularity (importance) of the start node (i.e., of the protein associated with the start node).

2. Indegree of the end node: Denotes the popularity (importance) of the end node (i.e., of the protein associated with the end node).

3. Outdegree of the start node: Denotes the number of proteins interacting with the protein at the start node.

4. Outdegree of the end node: Denotes the number of existing proteins interacting with the protein at the end node; correlates loosely with the likelihood of a reciprocal link.

5. Number of mutual "friends" of a protein w, such that u → w ^ w → v, for some proteins u and v.

6. Number of mutual "friends" of a protein w, such that v → w ^ w → u, for some proteins u and v.

7. Number of mutual "friends" of a protein w, such that u → w ^ v → w, for some proteins u and v.

8. Number of mutual "friends" of a protein w, such that w → u ^ w → v, for some proteins u and v.

9. Backward distance from v to u in the graph: identifies how far the protein v is from protein u.

The diagrammatic representations of the nine features considered are as shown in Figure 1 (a – i) below:

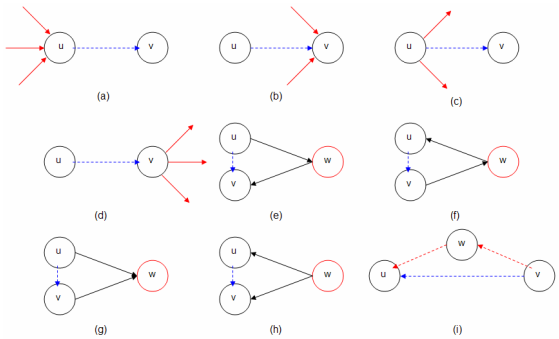

Figure 1: Node and topological features. The objects in red denote the feature that we calculate. The dashed lines (in blue) above indicate that a link between two proteins u and v may be either present or absent, i.e. either u or v are directly connected or indirectly connected via another node w.

Our technique consists of the following steps:

1. Preprocess the data and construct a graph network from the PPI data.

2. Generate candidate interacting proteins from the graph by performing BFS search.

3. Extract the node and topological features for the candidate interacting proteins from the graph.

4. Divide the candidate proteins into training and test data.

5. Learn several classifiers using the training PPI dataset.

6. Test the classifiers learned on the test dataset.

7. Compare the results obtained with results reported using other approaches.

V. RESULTS

Based on the methodology described in the previous section, 20,496 protein-protein interaction candidate edges were discovered in the first dataset; 17,502 of the candidate edges resulted in negative examples (absence of a direct link between proteins), while 2,994 of them resulted in positive examples (presence of a direct link between proteins). In the second dataset, 7,242 candidate edges were discovered; 1,607 of them resulted in positive examples, while 5,635 of them resulted in negative examples. Thus, most of the candidate edges discovered (~86% in the first dataset, ~78% in the second dataset) were negative examples. It is easy to see that a classifier that predicts all examples as negative examples can achieve an accuracy of 86% for the first dataset and an accuracy of 78% for the second dataset. To avoid this, we balanced the data by randomly sampling 2,994 negative examples without replacement from the total number of negative examples in the first dataset, to get a 50%-50% split of positive and negative samples. Similarly, we sampled 1,607 negative examples from the second dataset. We split both datasets into a training set containing 80% of the examples (50% positive and 50% negative) and a test set. The test set is obtained from the dataset containing 20% of the examples, by adding negative examples until the distribution matches the one of the original dataset. The classifiers used in this study are: Bagged RandomForest, RandomTree, J48, Bagged REPTree and ClassificationviaRegression, all available in WEKA. The classified (training) and non-classified (test) instances were provided to WEKA[13] in its native Attribute-Relation File Format (ARFF). The decision to use these classifiers was based on the results (with respect to the best classifiers) reported by Qi *et al.*[8] and Hsu *et al.*[3]. The classification results obtained for the first and second datasets are as shown as ROC curves in Figures 2 and 3 respectively.

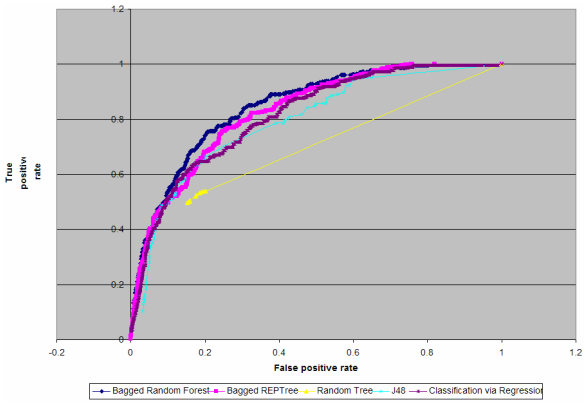

Figure 2: ROC curves for Bagged Random Forest, Bagged REPTree, Random Tree, J48 and Classification via Regression learning algorithms using the first dataset.

Figure 2 shows the ROC curves of the different classifiers used in our approach on the first dataset extracted from DIP in April 2006.

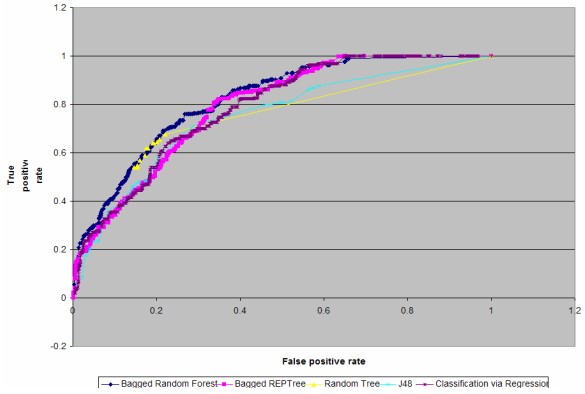

Figure 3: ROC curves for Bagged Random Forest, Bagged REPTree, Random Tree, J48 and Classification via Regression learning algorithms using the second dataset.

Figure 3 shows the ROC curves of the different classifiers used in our approach on the second dataset obtained from Qi et al.[8]. We extracted the true positive rate and false positive rate values from the ROC curve for REPTree Bagging as given by Licamele & Getoor[4]. Similarly, we identified the true positive and false positive rate values for our Bagged REPTree and Bagged RandomForest results. The comparison of our results with the results of Licamele & Getoor[4] is shown below in Figure 4.

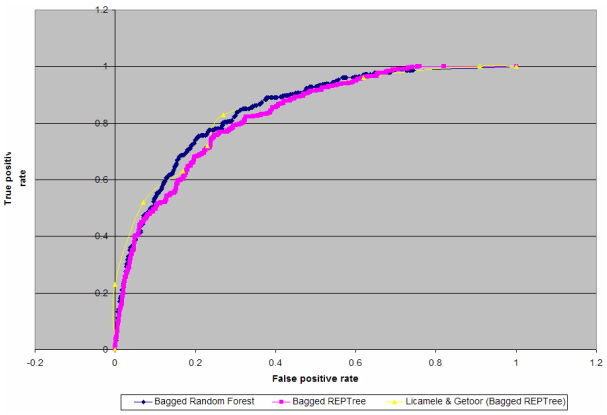

Figure 4: ROC curve comparing our best classifier (Bagged Random Forest), our Bagged REPTree and Bagged REPTree results as reported by Licamele & Getoor on their dataset[4].

Figure 4 shows that our approach compared well with the approach used by Licamele & Getoor[4]. We obtain a slightly higher accuracy (82.02%) and a slightly lower AUC score (0.845) using our best classifier (Bagged Random Forest) when compared with their results of Bagged REPTree (accuracy of 81.7% and AUC score of 0.8967). We also extracted the AUC score for RandomForest on the DIP dataset as given by Qi et al.[8] with the same 1:600 ratio of positive and negative examples as they used in their paper (i.e., 1 positive example for every 600 negative examples). We calculated the AUC score for RandomForest on our two datasets and compared the results (Figure 5).

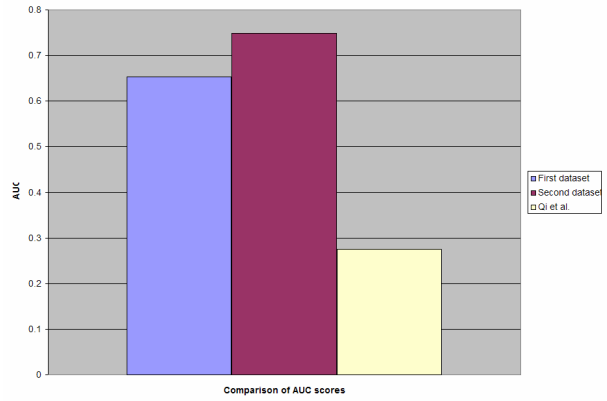

Figure 5: Comparison of AUC scores of RandomForest using our approach on the first and second dataset and the approach used by Qi *et al.*[8]

Figure 5 shows that the AUC score generated by RandomForest using our approach was significantly higher than that observed by Qi *et al.*[8] at the same ratio of positive and negative examples. We study different ratios of positive and negative examples to identify the optimum ratio which will give the best AUC score. The results are shown in Figure 6.

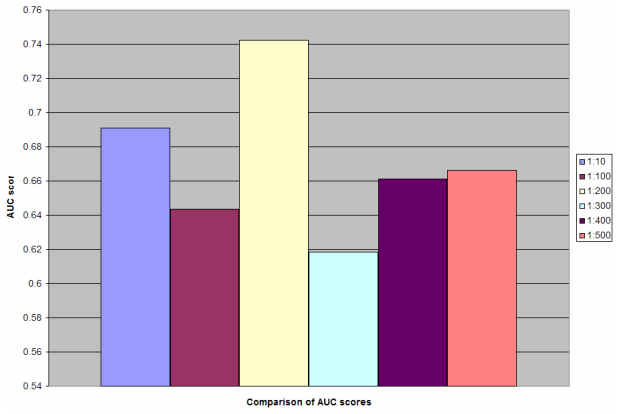

Figure 6: Comparison of AUC scores for different ratios using RandomForest on the second dataset only

Figure 6 shows that the AUC scores vary without any pattern when the ratios of positive and negative examples are increased. This is because, the positive examples are randomly sampled and we infer that different samples might change the AUC score for that particular ratio.

Based on the results, we conclude that our method of predicting protein-protein interactions performs slightly better than the existing methods for the same task. The comparisons have shown that our method compares well with the method by Licamele & Getoor[4] (our approach has a better accuracy but lower AUC score when compared with the approach of Licamele & Getoor[4]). The comparisons have also shown that we obtain a better AUC score using our approach on the same dataset used by Qi *et al.*[8]. The results are encouraging especially due to the fact that we do not use any features based on sequence or Gene Ontology information as used in the previous approaches to the PPI prediction problem.

In a separate comparative experiment, we also applied a Support Vector Machines (SVM) inducer for learning the PPI prediction task on the second dataset. We used a linear kernel in the SVM inducer and obtained an accuracy of 67.81%, precision of 70.36% and a recall of 61.54%. These results are not as good as those of other learning algorithms described in this paper.

VI. Conclusion & Future Work

In this study, we have addressed the problem of predicting protein-protein interactions based on an interaction network graph. We have identified nine structural features for *Saccharomyces cerevisiae* protein interaction networks. Based on these features, we have learned several classifiers and evaluated them on separate test sets. We have compared our results with previous results obtained for the same problem using different approaches (which use relational features of the PPI network). The results look promising. Future work is aimed at exploring the possibility of including features extracted from protein sequences. We expect that the addition of features derived from sequence will result in better ROC curves. Finally, we aim to use more learning algorithms to determine if higher accuracy and AUC score can be obtained.

ACKNOWLEDGMENTS

We thank Vikas Bahirwani, Tejaswi Pydimarri, Tim Weninger and other members of Knowledge Discovery in Databases Laboratory at Kansas State University for their helpful discussions on the social network problem.

REFERENCES

[1] Bock, J., & Gough, R. (2001). Predicting protein-protein interactions from primary structure. *Bioinformatics* 17 (pp. 455–460).

[2] Chou, K., & Cai, Y. D. (2006). Predicting protein-protein interactions from sequences in a hybridization space. *Journal of Proteome Research*, 5 (pp. 316–322). American Chemical Society.

[3] Hsu, W.H., King, A.L., Paradesi, M.S.R., Pydimarri, T., Weninger, T. 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.

[4] Licamele, K., Getoor, L. Predicting Protein-Protein Interactions Using Relational Features, *Proc. of ICML Workshop on Statistical Network Analysis 2006*.

[5] Maslov S., Sneppen K. Specificity and stability in topology of protein networks, *Science*, vol. 296. no. 5569, pp. 910 – 913, 2002.

[6] Mering, C.V, Krause, R., Snel, B., Cornell, M., Oliver, S.G., Fields, S., Bork, P., Comparative assessment of large-scale data sets of protein–protein interactions, *Nature* 417, 399-403.

[7] Paradesi, M.S.R., Wang, L., Brown, S.J., Hsu, W.H., Mining Domain Association Rules From Protein-Protein Interaction data, *Intelligent Engineering Systems through Artificial Neural Networks*, vol. 16, 213-218.

[8] Qi, Y., Bar-Joseph, Z., Klein-Seetharaman, J., Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins: Structure, Function, and Bioinformatics*, Volume 63, Issue 3, 2006, 490-500.

[9] Qi, Y., Klein-Seetharaman, J., Bar-Joseph, Z., A mixture of experts approach for protein-protein interaction prediction. *Proceedings of NIPS workshop on Computational Biology and the Analysis of Heterogeneous Data 2005*.

[10] Sakaki, Y. (2000). Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc Natl Acad Sci*, 97, 1143–1147.

[11] 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 Research*, 32, D449--D451.

[12] 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. A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature* 2000;403:623–627.

[13] Witten I.H., Frank E. (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[14] Zhang L.V., Wong S.L., King O.D., Roth F.P., Predicting co-complexed protein pairs using genomic and proteomic data integration, *BMC Bioinformatics 2004,5:38*.