Predicting Protein-Protein Interactions Using Relational Features

Louis Licamele [a]*; Lise Getoor [a]

[a]Computer Science Department, University of Maryland, College Park

ABSTRACT

Motivation: Proteins play a fundamental role in every process within the cell. Understanding how proteins interact, and the functional units they are part of, is important to furthering our knowledge of the entire biological process. There has been a growing amount of work, both experimetal and computational, on determining the protein-protein interaction network. Recently researchers have had success looking at this as a relational learning problem.

Results: In this work, we further this investigation, proposing several novel relational features for predicting protein-protein interaction. These features can be used in any classifier. Our approach allows large and complex networks to be analyzed and is an alternative to using more expensive relational methods. We show that we are able to get an accuracy of 81.7% when predicting new links from noisy high throughput data.

Contact: licamele@cs.umd.edu

1 INTRODUCTION

Recently there has been a growing interest in the ability to predict protein-protein interactions. Learning about these interactions will not only allow scientists to attain important new knowledge about how biochemical pathways work and are regulated but can also lead to the development of novel and better drugs. The need for computational methods for predicting the most likely interactions is important because in even a simple genome like yeast, which has roughly 6,000 proteins, there are around 18,000,000 possible interaction pairs. The number of known direct interactions for this organism is in fact only in the order of 30,000 thousand protein pairs. The cost to test each interaction is expensive in both time and money. There are currently several high throughput methods, but they have large error rates, sometimes upwards of 40%.

An interaction between two proteins can mean several different things. In its purest sense, it represents that two proteins directly, physically interact with each other. The prediction of this type of direct interaction has been the focus of many researchers (Bock and Gough, 2001; Chou and Cai, 2006). The features used in this one-to-one prediction consist of physical properties of the two proteins and include everything from the size of the protein and its sub-cellular localization to the type of physical subunits (alpha coils/beta sheets) the secondary structure will contain.

Recently, a number of researches have shifted their focus to predicting when two proteins belong to the same protein complex or pathway. This can be considered as a form of indirect interaction. This type of interaction is much more relational in nature, e.g. if A interacts with B, and B interacts with C, then we are more likely to believe that A interacts with C.

*to whom correspondence should be addressed

Yu *et al.* (2006) is one of the first works to directly exploit relational information in protein-protein interaction prediction. They have shown how completing defective cliques in protein-protein interaction networks can lead to improved predictions. In their approach, a new link is predicted between protein A and protein B if they both were fully connected to some clique X of proteins. Although their method produces accurate predictions, it is limited in the number of new predictions it can find due to the limited number of large cliques.

Jaimovich *et al.* (2006) formalized this as a statistical relational learning task and applied relational Markov networks (RMNs). Using this approach they were able to obtain good results for the prediction task. However, because the approximate inference techniques they use have difficulty converging on dense networks, this approach is limited to only working with small subsets of the interaction network. More recently Airodi *et al.* (2006) have proposed a mixed-membership stochastic block model approach that also handles relational data and has shown to work well for this prediction task.

Here, we propose a general approach for using structural information to predict protein-protein interactions. We investigate a variety of methods for describing link structure and we combine these link characterizations with information about the individual proteins in a generic off-the-shelf non-relational classifier. Our approach is applicable in more settings than the defective clique approach. Because the computational complexity of prediction in our approach is not dependent on the density in the underlying protien network, our approach more scalable than RMN methods. For this work we will not make a distinction between direct and indirect interactions.

2 DATASET

The focus of this work is on the *Saccharomyces cerevisiae* genome, commonly known as baker's or budding yeast. This is one of the most studied organisms for this task and is believed to have one of the most complete interaction networks of any genome. The interaction datasets that will be used include the BIND, DIP and MIPS datasets as well as the in vivo pull-down hybrid and the yeast two hybrid datasets, all of which are listed in Table 1. One thing to note is the different types of interactions that these datasets include. For instance, the DIP dataset contains direct interactions between proteins while the MIPS dataset considers two proteins to be interacting if they are present in the same complex. These datasets contain records for proteins interacting with themselves, however, we will follow what others in this area have done and remove these links.

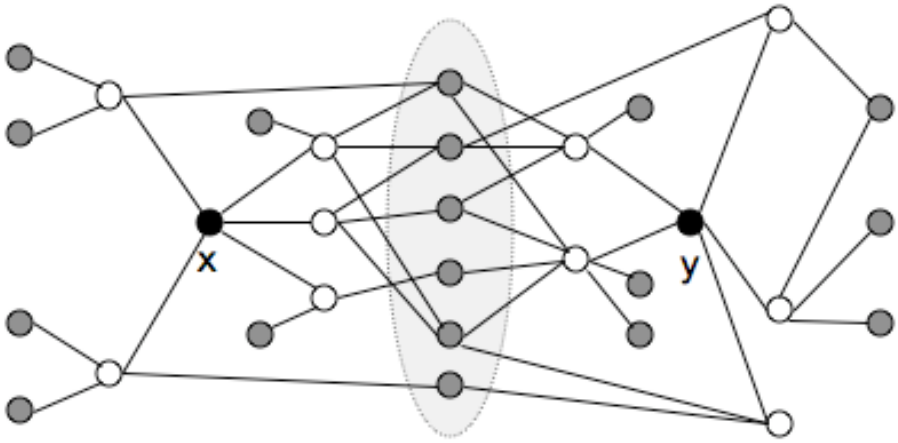

Fig. 1. The second-order shared neighborhood of x and y is highlighted, and does not include nodes which are present in the 1st order shared neighborhood. Nodes in the first order neighborhood are white, and nodes in the full second order neighborhood are grey. Links are not shown between nodes in the same neighborhood for simplicity. The size of the shared neighborhood is 6 and the shared neighborhood ratio is 1/3 (6/18).

3 PROTEIN FEATURES

These datasets only contain relational data; however, the majority of previous work in this area has focused on using attributes of the proteins themselves. There is an abundant amount of features that we can use, including everything from the expression levels of proteins and where they are most commonly found (sub-cellular localization) to features that we can predict will exist in the three dimensional structure of the folded protein.

One set of protein features that we use is listed in Table 2. These features are adapted from the features used in the analysis of TopNet (Yu *et al.*, 2004), a tool for comparing biological sub-networks. Another set of features is available through the yeast genome project (www.yeastgenome.org). These features include the molecular weight, PI, CAI, protein length, nucleotide length, and the codon bias. In addition, we have a feature representing the distribution of the different amino acids in the protein. Finally, we create a boolean feature for each location in the cell that a protein can be found (ambiguous, mitochondrion, vacuole, spindle pole, cell periphery, vacuolarmembrane, er, nuclear periphery, endosome, bud neck, microtubule, golgi, golgi-to-vauole, peroxisome, actin, nucleolus, cytoplasm, er-to-golgi, golgi-to-er, lipidparticle, nucleus, bud, and punctuate composite) as well as the abundance of the protein that was found in this location.

4 STRUCTURAL FEATURES

Next, we constructed several structural features. The first structural feature we introduce is based on structural information contained in the Gene Ontology (Consortium, 2000). We introduced a feature designed to capture the similariy between the Gene Ontology labels of the two proteins.

Determining accurate and valid comparisons in the Gene Ontology is a significant challenge. First there is the issue that the Gene Ontology is modeled at varying degrees of completeness, depending on areas which have received more or less research attention. In addition, a label may appear in multiple places in the ontology. Here we propose a simple measure which we have found effective. We define the GO distance as distance from the two labels to their closest common ancestor, and weight this by the depth of the closest common ancestor in the ontology. This capture the notion that nodes at lower depths, e.g. closer to the root, are more general terms and hence less informative. We also tried simply using the distance to closest common ancestor, but this did not work well.

Additional features were constructed based on the relational structure of the protein interaction network. Two proteins are neighbors in this network if we have laboratory evidence that the two proteins interact. The features we construct capture the neighborhood similarity in various ways.

The simplest features to consider are ones that directly compare first-order neighbors. For example, we can look at the shared neighbors of two proteins. An example of the first-order shared

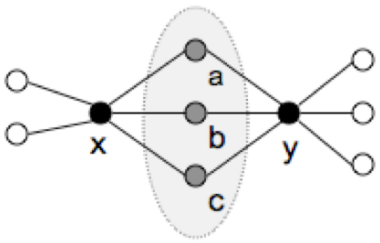

Fig. 2. The first-order shared neighborhood of nodes x and y is highlighted, and consists of a,b and c. To keep the layout simple, links are only shown if they involve x or y. The size of the shared neighborhood is 3 and the shared neighborhood ratio is 3/5.

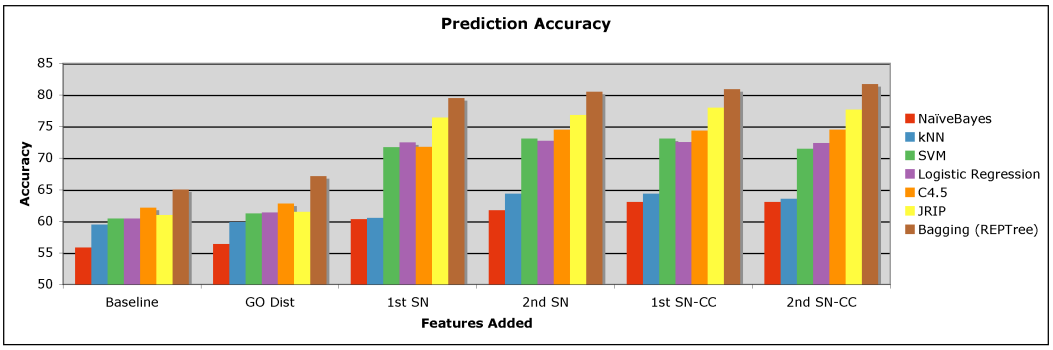

Fig. 3. Prediction accuracy for each of the classifiers and for all of the features. The baseline consists of the non-relational features. Going from left to right, the features listed are added into the cumulative set of features, with the baseline. GO Dist is the Gene Ontology distance, 1st SN is the first-order shared neighborhood, 2nd SN is the second-order shared neighborhood, 1st SN-CC is the clustering coefficient of the first-order shared neighborhood and 2nd SN-CC is the clustering coefficient of the second-order shared neighborhood.

neighborhood of two proteins is show in Figure 2. We introduce a feature which simply measures the size of the shared neighborhood. In addition, we introduce a feature that captures the proportion of the full neighborhood of the two proteins that is within the shared neighborhood. For this we will use the ratio of the proteins that are shared to the whole neighborhood.

Protein-protein interaction networks are complex and there is still a large amount that is unknown about how they work. It is possible that the interactions of the neighbors of a protein may also be informative. We also examine the second-order shared neighborhood (shown in Figure 1). We introduce features which capture the size of the second-order shared neighborhood and the proportion of second-order neighborhood that is shared.

As Yu *et al.* (2006) have shown, it is important to consider how connected the potential protein interaction pair is to a clique. In our case, the clique of interest is over the shared neighborhood, which can be either the first or second-order shared neighborhood.

We measure the connectedness of this neighborhood using the clustering coefficient of the shared neighborhood. The clustering coefficient of the shared neighborhood of x and y is represented in Equation 1. We define N_{xy} as the shared neighborhood of x and y.

$$C_{xy} = \frac{2\,|\{(v_j, v_k) \in E \mid v_j, v_k \in N_{xy}\}|}{|N_{xy}| \times (|N_{xy}| - 1)} \tag{1}$$

5 METHODS

We tried several standard machine learning classifiers. Here we give details on the one that performed the best: a boosted REP Tree. A REP Tree is a Decision Tree with Reduced Error Pruning.

5.1 Decision Trees

A decision tree is a classifier in which the nodes in the tree represent decisions and the classifications are determined at the leaves. Each non-leaf node in the tree represents a decision, or a test, on one of the attributes. The outcome of this test dictates which branch should be followed when leaving this node. New instances are classified by sorting them down the tree from the root to a leaf node which specifies the classification of the instance.

The process of learning decision trees can be broken down into two phases. In the first phase a tree is learned which completely explains the training set. Decision tree learning is a greedy algorithm in which the attribute that best splits the data coming into a node is used as the test for the outcome of the node. Information gain is most commonly used to measure the *best* attribute to use for splitting a node. The attribute with the biggest information gain is the one that reduces the entropy the most. Entropy is defined in Equation 2 by Mitchell (1997) where $p_\oplus$ represents the ratio of positive instances in S while $p_\otimes$ is the ratio of the negative instances. Equation 3 shows how information gain is calculated from the change in entropy.

$$Entropy\,(S) \equiv -p_\oplus \log_2 p_\oplus - p_\otimes \log_2 p_\otimes \tag{2}$$

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{3}$$

Decision Tree Learning Algorithm. Adapted from Mitchell (1997).

1. Begin at the root
2. Select attribute $a \in A$ with the largest information gain that has not yet been used in the path to this node
3. For each value $v_i \in Values(a)$
   a. Create a child node c_i
   b. Attach all instances with attribute $a = v_i$ to the child node c_i
   c. If all instances are of the same class x_j
      (1) Label node c_i as a leaf with classification x_j
   d. If there are unused attributes
      (1) Go to step 2 with node c_i
   e. Label the current node as a leaf
      (1) Set the classification to be the most common class of the instances attached to this node

The tree that is created is the best fit on the training data, but it most likely overfits the training samples. This brings us to the second phase in which the tree is pruned in order to reduce its

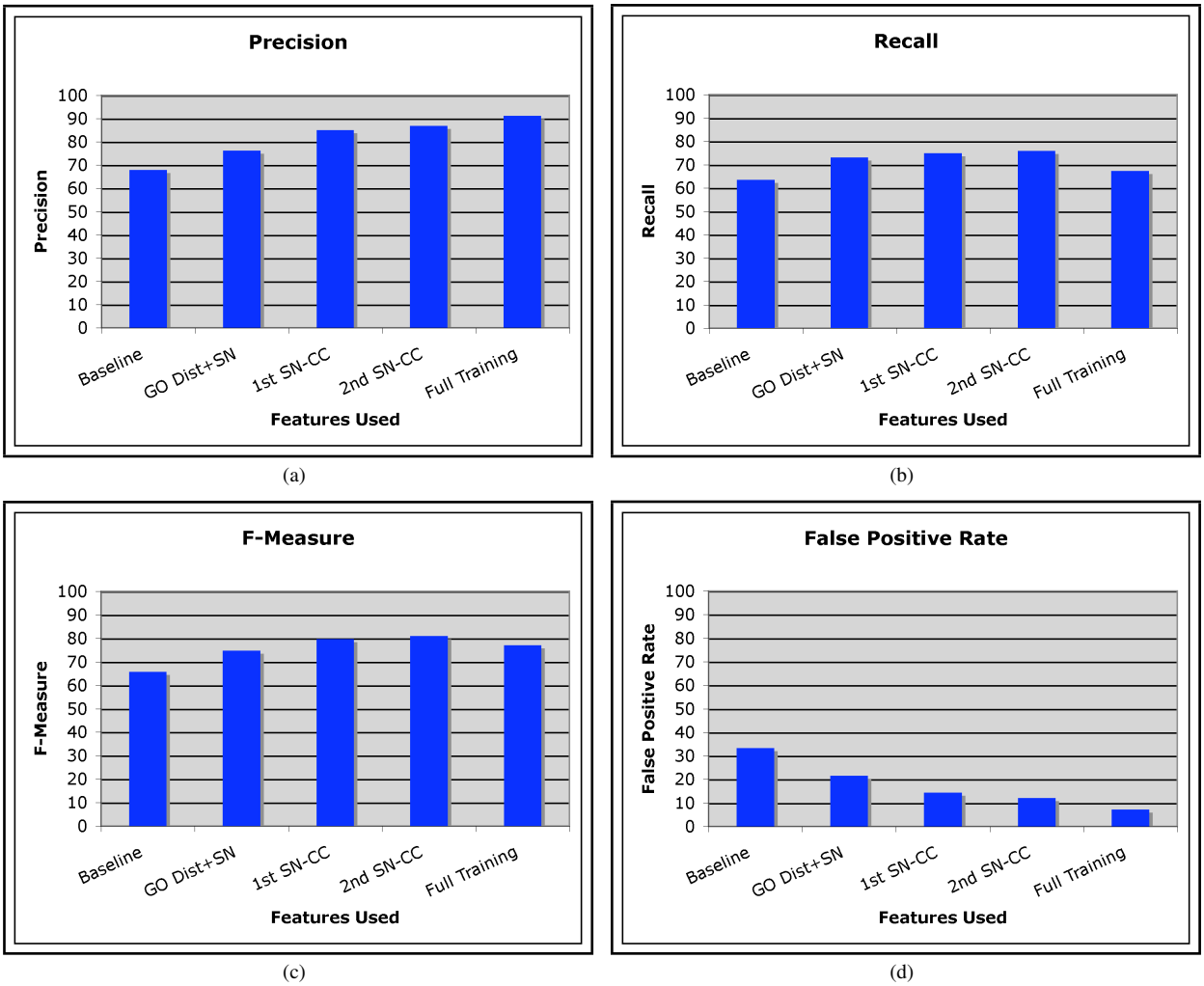

Fig. 4. Statistics for the different features. These are all from using Bagging(REPTrees). (a) shows the precision. The recall is shown in (b) with the F-measure in (c). In (d) we display the False Positive Rate. Additionally, the last column in each of the graphs represents the results for training on the full training set, which results in a small decrease in recall but has a large increase in precision.

dependency on the training data and allows the tree to be generalized to fit other examples. There are many different methods to prune decision trees, but here we will explain that of the best classifier which was REP trees.

Quinlan (1987) first introduced Reduced Error Pruning (REP) as a method to prune decision trees. REP is an simple pruning method though it is sometimes considered to overprune the tree. A separate pruning dataset is required, which is considered a downfall of this method because data is normally scarce. However, REP can be extremely powerful when it is used with either a large number of examples or in combination with boosting. The pruning method that is used is the replacement of a subtree by a leaf representing the majority of all examples reaching it in the pruning set. This replacement is done if this modification reduces the error, i.e. if the new tree would give an equal or fewer number of misclassifications.

5.2 Bagging

Classification errors in machine learning can come from several sources: bias, variance and noise. The bias refers to the accuracy of the algorithm itself, the variance measures the precision or specificity of the algorithm, and the noise is the intrinsic target noise. Our main goal is to limit the variance of the model. The variance measures how changes in the training data can effect the model. This requires many training sets to test. With limited data we can simulate multiple training sets by using bootstrapping, which leads us to bagging.

Bagging (Breiman, 1996) is bootstrapping with aggregation. Essentially, a classifier is built by resampling and combining the results from several iterations and averaging the results across the different iterations. For a training set of n instances, a single bootstrap sample is created by randomly sampling, with replacement, n instances. The sampled set has the same number of instances, but some examples are over-represented while others are completely left out. The variance is reduced by smoothing out the

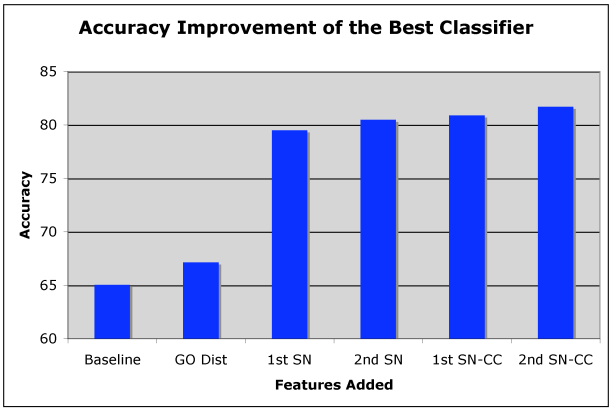

Fig. 5. Accuracy of the best classifier – Bagging with REPTrees. The improvement over the baseline is substantial, and each addition of a new feature is statistically significant over the previous set.

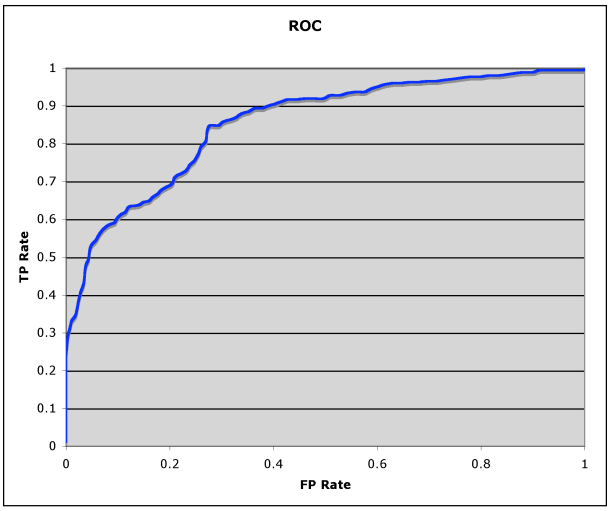

Fig. 6. ROC curve for REPTree Bagging with the full set of features (The AUC is .8967).

prediction across all the pseudo distinct training sets which learns which examples are driving the classifier in the wrong direction.

6 EVALUATION

There are common gold standards used by the community that can be used for validation of the predicted links. Specifically it is common to use interactions that are present across several datasets. These gold standards include only true positives, or protein-protein interactions that are known to exist. However, there is no gold standard for proteins pairs that do not interact. Instead we assume that protein pairs that are not known to interact are the negative examples.

Baseline classifiers included our own implementation of Naive Bays and Logistic Regression, as well as kNN, C4.5, JRIP and Bagging with REPTrees from Weka and SVMs using SVMLight. 5 fold cross validation was performed. Each training set contained 4,000 positive examples and 4,000 negative examples. The test set contained 1,000 positive examples and 1,000 negative examples. All of these instances were randomly sampled without replacement from the full set of known interactions.

The results for all of the classifiers are shown in Figure 3. The results are shown for each classifier starting with the baseline. As can be seen, Bagging with REPTrees does the best across all feature sets so we will focus our analysis on this classifier. The results from using Bagging are presented more clearly in Figure 5.

6.1 Baseline

The baseline, which contains only the non-relational features, was able to achieve an accuracy of 65.07%. The precision for the positive class is 67.9% and the recall is 63.5%. The false positive rate, which is also very important in this domain, is 33.2%. We will use this baseline to compare how our relational features help in the prediction of interactions between two proteins.

6.2 Gene Ontology Distance

Adding in the Gene Ontology distance between the two proteins increase the accuracy in all of the classifiers. For Bagging with REPTrees the accuracy was increased to 67.14%. This result is statistically significant by a paired t test.

6.3 Shared Neighborhood

We next evaluate the features representing the first order shared neighborhood to our set of features. As expected this increases the accuracy of our classifier substantially to 79.5%. By including this simple feature we have reduced the error from the baseline by 41.3%.

By adding in the second-order shared neighborhood, we increase the accuracy to 80.5%. Though only a small increase, it is statistically significant.

6.4 Clustering Coefficient of Shared Neighborhood

We have shown that the first and second order shared neighborhoods are good features to use in this task, but we can improve further on these results if we combine this with more details about these neighborhoods. We therefore add the clustering coefficient of the first-order shared neighborhood. This leads to an increased accuracy of 80.9% (statistically significant). More importantly it leads to a large increase in the precision (Figure 4a) as well as a reduction in the false positive rate (Figure 4d).

To complete our features, we next add in the clustering coefficient of the second-order shared neighborhood. This continues to increase the accuracy to 81.74%. The F-measure continues to increase (Figure 4c) while reducing the false positives (Figure 4d). This result at first might seem surprising because this feature actually ends up helping more than the second order shared neighborhood features themselves, or even the clustering coefficient of the first-order shared neighborhood. This result makes sense, because as we move out to the second order shared neighborhood we are introducing more noise, and by using the clustering coefficient of this larger shared network we are able to reduce the noise and create a more informative feature. The ROC curve for using all of the features is shown in Figure 6. The AUC is .8967.

6.5 Scalability

To show the scalability of this method we trained on the full set of validated protein interaction pairs. This training set consists of

Table 1. Overview of the interaction datasets. Adapted from Yu *et al.* (2004).

DATASET	NO. ORFS COVERED	NO. OF INTERACTIONS
MIPS COMPLEX CATALOGS (MEWES *et al.*, 1999)	871	8,250
BIND (BADER *et al.*, 2001)	3,789	5,965
DIP (XENARIOS *et al.*, 2002)	4,716	15,113
YEAST TWO-HYBRY (ITO *et al.*, 2000)	3,278	4,393
YEAST TWO-HYBRID (UETZ *et al.*, 2000)	1,044	981
IN VIVO PULL-DOWN (GAVIN *et al.*, 2002)	1,361	31,304
IN VIVO PULL-DOWN (HO *et al.*, 2002)	1,578	25,333

Table 2. Overview of biologically inspired protein categories. Adapted from Yu *et al.* (2004)

CATEGORY	ORFS	GROUPS	DESCRIPTION
EXPRESSION (CHO *et al.*, 1998)	6.130	13	CELL-CYCLE EXPRESSION DATA
PROTEIN SIZE (CHERRY *et al.*, 1997)	6,092	12	DERIVED FROM AMINO ACID DATA
AMINO ACID COMPOSITION (GRANTHAM, 1974)	6,092	14	DERIVED FROM GENOMIC SEQUENCE
SUBCELLULAR LOCALIZATION (KUMAR *et al.*, 2002)	2,902	4	TRANPOSON TAGGIN
FUNCTION (MEWES *et al.*, 1999)	3,936	2	MIPS FUNCTIONAL CATALOGS
SEQUENCE CONSERVATION (TATUSOV *et al.*, 1997)	4,139	5	COG DATABASE
TERTIARY STRUCTURE (FOLD) (LO CONTE *et al.*, 2002)	3,471	452	SCOP DATABASE
FOLD CLASS (LO CONTE *et al.*, 2002)	3,471	7	SCOP DATABASE
SECONDARY STRUCTURE (SEN *et al.*, 2005)	6,092	7	PREDICTED BY GOR IV
SOLUBILITY (KROGH *et al.*, 2001)	6,092	14	PREDICTED BY TMHMM SERVER

13,000 positive examples and 13,000 randomly sampled negative examples. Our method, since it relies on simple classifiers, can easily handle large training sets. Unlike an RMN, we are not limited by the denseness of the network. This will only lead to slightly longer preprocessing time to select these relational features out of the database. The results of this analysis is shown in Figure 4 (the last column). Though this led to a slight decrease in the overall accuracy to 79.4%, we were able to reduce the false positive rate to 7.2%. This is extremely important to a biologist. A reduction in false positives is very important as it will reduce both monetary costs as well as time required in the lab to verify any new predictions that are found.

7 CONCLUSION

We have presented a method for using relational features from the protein-protein interaction network to help discover new interacting pairs of proteins. Our method combines features of the proteins with structural aspects of the network and is able to make use of off-the-shelf machine learning algorithms. We have shown that in addition to the first order shared neighborhood, the second order shared neighborhood is informative. The clustering coefficient for the shared neighbors is also informative.

The accuracy of 81.7% achieved by our method is in line with or higher than the majority of methods which do not make use of relational information. Most of these methods have only been applied to predicting direct interactions and not the more general case of interactions which indicate the proteins belong to the same complex, which many consider harder to predict. Our method is able to handle complex and dense networks unlike RMNs. RMNs can perform better in certain cases, but our method is simpler and provides an alternative to RMNs when they are not sufficient. The ability of being able to train on a large set of instances has led to the ability to obtain a very low false positive rate as compared with other methods.

8 FUTURE WORK

In order to better compare our method with the non-relational approaches it would be useful to examine how our approach will perform on predicting the direct protein-protein interactions. This will provide a much higher baseline to compare our method against. It is our belief that the new relational features will continue to help. It would also be interesting to examine how the clustering coefficient of shared neighborhoods might be informative in other domains and types of networks.

REFERENCES

Airodi, E., Blei, D., Xing, E., and Fienberg, S. (2006). Mixed membership stochastic block models for relational data, with applications to protein-protein interactions. In *Proceedings of International Biometric Society-ENAR Annual Meetings*.

Bader, G., Donaldson, I., Wolting, C., Ouellette, B., Pawson, T., and Hogue, C. (2001). Bind–the biomolecular interaction network database. *Nucleic Acids Research*, pages 242–245.

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

Breiman, L. (1996). Bagging predictors. *Machine Learning*, **24**(2), 123–140.

Cherry, J. M., Ball, C., Weng, S., Juvik, G., and et. al (1997). Genetic and physical maps of saccharomyces cerevisiae. *Nature*, **387**(6632 Suppl), 67–73.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., and et. al (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell*, **2**(1), 65–73.

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

Consortium, T. G. O. (2000). Gene ontology: tool for the unification of biology. In *Nature Genetics. 25: 25-29*.

Gavin, A. C., Bsche, M., Krause, R., and Grandi, P. e. a. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, **415**(6868), 141–147.

Grantham (1974). Amino acid difference formula to help explain protein evolution. *Science*, **185**, 862.

Ho, Y., Gruhler, A., Heilbut, A., Bader, G. D., and et. al (2002). Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, **415**(6868), 180–183.

Ito, T., Tashiro, K., Muta, S., Ozawa, R., Chiba, T., Nishizawa, M., Yamamoto, K., Kuhara, S., and 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**(3), 1143–1147.

Jaimovich, A., Elidan, G., Margalit, H., and Friedman, N. (2006). Towards an integrated protein-protein interaction network: a relational markov network approach. *J Comput Biol*, **13**(2), 145–164.

Krogh, A., Larsson, B., von Heijne, G., and Sonnhammer, E. L. (2001). Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *J Mol Biol*, **305**(3), 567–580.

Kumar, A., Agarwal, S., Heyman, J. A., Matson, S., and et. al (2002). Subcellular localization of the yeast proteome. *Genes Dev.*, **16**(6), 707–719.

Lo Conte, L., Brenner, S. E., Hubbard, T. J., Chothia, C., and Murzin, A. G. (2002). Scop database in 2002: refinements accommodate structural genomics. *Nucleic Acids Res*, **30**(1), 264–267.

Mewes, H. W., Heumann, K., Kaps, A., Mayer, K., Pfeiffer, F., Stocker, S., and Frishman, D. (1999). Mips: a database for genomes and protein sequences. *Nucleic Acids Res*, **27**(1), 44–48.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

Quinlan, J. R. (1987). Simplifying decision trees. In B. Gaines and J. Boose, editors, *Knowledge Acquisition for Knowledge-Based Systems*, pages 239–252. Academic Press, London.

Sen, T. Z., Jernigan, R. L., Garnier, J., and Kloczkowski, A. (2005). Gor v server for protein secondary structure prediction. *Bioinformatics*, **21**(11), 2787–2788.

Tatusov, R. L., Koonin, E. V., and Lipman, D. J. (1997). A genomic perspective on protein families. *Science*, **278**(5338), 631–637.

Uetz, P., Giot, L., Cagney, G., Mansfield, T. A., and et. al (2000). A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, **403**(6770), 623–627.

Xenarios, I., Salwnski, L., Duan, X. J., Higney, P., Kim, S. M., and Eisenberg, D. (2002). Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res*, **30**(1), 303–305.

Yu, H., Zhu, X., Greenbaum, D., Karro, J., and Gerstein, M. (2004). Topnet: a tool for comparing biological sub-networks, correlating protein properties with topological statistics. *Nucleic Acids Res*, **32**(1), 328–337.

Yu, H., Paccanaro, A., and Gerstein, M. (2006). Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, **22**(7), 823–829.