Application of Data Mining Techniques to Protein-Protein Interaction Prediction*

A. Kocatas[1], A. Gursoy[1], and R. Atalay[2]

[1] Computer Engineering Department,
Koç University, 34450 İstanbul, Turkey
[2] Department of Molecular Biology and Genetics
Bilkent University, 06800 Ankara, Turkey

Abstract. Protein-protein interactions are key to understanding biological processes and disease mechanisms in organisms. There is a vast amount of data on proteins waiting to be explored. In this paper, we describe application of data mining techniques, namely association rule mining and ID3 classification, to the problem of predicting protein-protein interactions. We have combined available interaction data and protein domain decomposition data to infer new interactions. Preliminary results show that our approach helps us find plausible rules to understand biological processes.

1 Introduction

With recent advances in modern biology and biotechnology, the amount of biological data keeps accumulating in unprecedented speed. Therefore, it is extremely important to analyze such a vast and diverse collection of data to understand biological processes. Data mining is one of the emerging areas to extract knowledge from large sets of data. In this paper, we will discuss application of rule mining and ID tree learning methods to the prediction of protein-protein interactions.

Protein-protein interactions are key to understanding biological processes and disease mechanisms in organisms. Most protein-protein interactions have been discovered by laboratory techniques such as yeast two-hybrid system that can detect all possible combinations of interactions. However, these findings can be superfluous and do not necessarily explain exact relationship between proteins. It is important to relate certain features of proteins and their functions for both to understand the process and also lead new knowledge based on this understanding.

In this study, we focus on relationship between the sites of proteins that are involved in interactions. Such regions of proteins are called domains. Proteins can be characterized by combination of domains and proteins interact with each other through their domains to carry out biological functions. Using databases

* This work was partially supported by the Turkish Academy of Sciences to RCA (in the framework of young Scientist Award Program-RCA/TUBA-GEBIP/2001-2-3)

A. Yazici and C. Şener (Eds.): ISCIS 2003, LNCS 2869, pp. 316–323, 2003.

of known protein-protein interactions and databases of domain decomposition of proteins, it is possible to draw certain relationships. For example, if we can conclude rules such as proteins having domain x generally interact with proteins having domain y, then this knowledge might help biologists to interpret biological processes better, and predict unknown interactions as well.

This paper is organized as follows: First, we briefly explain association rule mining and its application to protein-protein interaction and domain decomposition data. Then, ID3 classification method and its application is discussed. Specific databases, namely DIP [1], Pfam [3], and Yeast [6] databases, and preprocessing of them to be used in rule mining is explained. Finally, we present preliminary results and some plausible biological explanations for the rules found.

2 Association Rule Mining

Association rule mining is a data mining technique that was proposed in [4]. It has emerged because of the need for extracting rules from the supermarket shopping basket data. With the use of bar code system, experimental data about shopping baskets is very easy to collect. Such databases include a set of transactions. Every transaction represents the shopping basket of a customer, which simply includes a list of different items that she bought.

Below is a sample database of transactions, where $T = \{T_1, T_2, ..., T_M\}$ is the set of transactions and $I = \{i_1, i_2, ..., i_N\}$ are the set of distinct items that can appear in a typical transaction.

$$\boxed{\begin{array}{l}
T_1 = \{i_1, i_2, i_3, i_4\} \\
T_2 = \{i_1, i_2, i_3, i_5\} \\
T_3 = \{i_2, i_3, i_4\} \\
T_4 = \{i_1, i_2, i_5, i_7\} \\
T_5 = \{i_1, i_2, i_3\}
\end{array}}$$

An example association rule that can be extracted from this database is: $i_1, i_2 \rightarrow i_3$. This rule basically means that if a customer has bought i_1 and i_2, he also bought i_3. There are two variables that measure the dependability of an association rule. First one is the *support* value, which is defined as the number of transactions that contain all the items in the rule. For the rule $i_1, i_2 \rightarrow i_3$, the support value is 3. The second value that characterizes an association rule is its *confidence* value, which is the ratio of the support of the rule to the number of transactions that contain the left side (if part) of the rule. Again for the rule above, the confidence value is $(3 \div 4) * 100 = 75\%$.

Finding association rules over a large database of transactions is a time consuming operation. However, recent methods that use clever algorithms, efficient data structures and parallel algorithms cope well with the problem. Both parallel and sequential efficient algorithms were implemented to face the high computational needs of the problem. However, details of these algorithms are beyond the scope of this paper. For our experiments, we have used an implementation [8] of *Apriori* [2], which is one of the most efficient sequential algorithms.

2.1 Proposed Method

In this section, we present a method that is used to adapt the protein-protein interaction data to be used in association rule mining. Clearly, the nature of the protein-protein interaction data is not the same as the supermarket basket data. However, we have the freedom of modifying the layout of the data to make it to be used in association rule mining.

A database of supermarket basket data simply includes only a list of transactions with the items per shopping basket stored in every record. However, interaction data is not that much plain. Every interaction involves two proteins, say protein A_x and protein B_y. A typical protein interaction database entry is like: $ProteinA_x \iff ProteinB_y$.

Using a second database, which stores the domain decomposition of the proteins, we convert the database of interacting proteins to a format where every protein is substituted with its set of domains. After all of the interacting proteins are decomposed into their set of domains, the final data entry looks like: $\{d_i, d_j, d_k...\} \iff \{d_l, d_m, d_n, ...\}$. In this data, left hand side (LHS) and right side (RHS) of the $\iff$ declaration corresponds to the set of domains for protein A_x and B_y, respectively.

Further, we should collapse RHS and LHS of the above interactions into a single set in order to make it applicable for association rule mining. However, it would be impossible to interpret the output of the association rule miner if we solely merge two sets into one set. For instance, assuming that we have merged the LHS and RHS of $ProteinA_x$ and $ProteinB_y$ into one set, where $x, y = 0, 1, ...p$ and p is the number of distinct proteins, resulting transactions are like: $T = \{d_i, d_j, d_k, ...d_l, d_n, d_m\}$. If we give interaction data in this form to the association rule miner, output rules will be like $d_i, d_j, ... \rightarrow d_k, d_l....$ The problem is that we cannot identify which domain in such a rule refers to which side of an interaction in the initial interaction data. The LHS and RHS information is eventually lost in such kind of approach, which is not acceptable. The solution is to put tags on the domains, which make us able to differentiate between the LHS and RHS domains. After putting the tags, an interaction looks like: $T_p = \{d_{iL}, d_{jL}, d_{kL}...d_{lR}, d_{nR}, d_{mR}\}$. Now that we know which domain is a RHS domain and which is a LHS protein by looking at the elements in the resulting rules, a final modification is to add the reverse of every interaction to the set of interactions. This is because an interaction among two proteins is not a directional relation and $ProteinA_x \iff ProteinB_y$ is the same as saying $ProteinB_y \iff ProteinA_x$. What's more, results of association rule miner can be biased in cases such as a set of domains are often seen in LHS of the interactions.

When this final form of data is given as an input to the association rule miner, we expect various kind of rules. Some of the rules will appear with higher support and confidence values, while some will appear with lower dependability parameters.

Let α and β denote the set of domains that appear in the LHS and RHS of an association rule, respectively. Then, we typically expect rules like: $\alpha \rightarrow \beta$.

Rules which have only LHS domains on one side and only RHS domains on the other side and vice versa are the most meaningful ones, because they imply a rule on the opposite sides of an interaction. Finding such rules enables us to make predictions like "if a protein has α domains and another protein has β domains, they will probably interact. We represent these rules as:

$$\alpha \rightarrow \beta : \alpha \in LHS \ \& \ \beta \in RHS \text{ and } \alpha \rightarrow \beta : \alpha \in RHS \ \& \ \beta \in LHS$$

Another rule type can be the following, where LHS and RHS of the interaction is composed of the same kind of domains (i.e all LHS domains):

$$\alpha \rightarrow \beta : \alpha \in LHS \ \& \ \beta \in LHS \text{ and } \alpha \rightarrow \beta : \alpha \in RHS \ \& \ \beta \in RHS$$

The third type is more complex. Namely, at least one side of the rule is composed of different kind of domains (i.e. left side of the rule is composed of LHS and RHS domains.) Following is one possible rule of this type, which we call as *composite rules.*:

$$\alpha \rightarrow \beta : \alpha \in LHS \ \& \ \beta \in LHS \ \& \ \beta \in RHS$$

Machine learning techniques have been used for extraction of knowledge on protein-protein interactions in other studies as well. The work done in [5] uses association rule mining to extract similar rules about proteins like: a protein which has the features α interacts with a protein which has the features β. Their work is similar to our approach however we only concentrate on domains of proteins whereas their work considers many features of proteins such as sequence, location in the cell etc.

3 ID3 Classification

Identification tree learning algorithm is another popular machine learning algorithm. The goal of ID3 is to find an approximate function that maps the a set of predicates to discrete values. ID3 represents this function as a decision tree. Decision trees classify instances by sorting them from the root of the tree down to the leaves. In a decision tree, going down from the root, towards any of the leaves, one can make a certain decision about the current problem instance represented by that leaf.

A set of examples are needed to train an ID3 learning algorithm. Every single record of an example has a set of attributes that classify the example to a distinct set according to the value of an attribute. Decision trees simply ask questions about the attributes to the whole example set at every node in order to classify the examples in consideration. Given this example set, output of an ID3 algorithm is a decision tree, which can be represented by a set of rules. The rules are constructed in such a way that from the root, down to the leaves, every single path in the tree can be represented as a rule.

ID3 algorithm tries to find the optimum decision tree among the set of possible decision trees. In order to build the best decision tree, the algorithm tests which attribute is the best classifier at every node. The measure used for the attributes is the *information gain* of the classification, due to an attribute. In other words, in every node, we measure the information gain of an attribute that classifies the set of examples at that node and select the attribute with highest *information gain* to apply to that node. We cannot mention the full details of the algorithm here, because of the space considerations.

3.1 Proposed Method

In order to create an example set out of a set of protein-protein interactions, we should find a suitable representation. It is important that we decide on what an *attribute* is and what are its *values*. The simplest way of doing this is to represent the existence of every domain as an attribute. Then, every attribute becomes a boolean predicate and can only take values 0 and 1. To complete this representation, we should name the attributes. A simple way of naming the attributes is to give names to the domains as $\{d_1, d_2, ..., d_{2N}\}$ where N is the number of distinct domains. In this representation, d_i is the name of the attribute that tests if the domain indexed by i exists in the interaction. All i's where $i > N$ represents the attributes that test the existence of the domains that appear in RHS of the protein-protein interaction.

We can categorize the types of rules that the ID3 algorithm can produce from such an example set into a few categories:

1. $\forall\ \alpha\ : \alpha \in LHS \rightarrow +$
2. $\forall\ \alpha\ : \alpha \in RHS \rightarrow +$
3. $\forall \alpha\ : (\exists \alpha : \alpha \in LHS\ \&\ \alpha = 1)\ \&\ (\exists \alpha : \alpha \in RHS\ \&\ \alpha = 1) \rightarrow +$
4. $\alpha \in (LHS\ or\ RHS)\ \&\ \alpha = 0 \rightarrow +$
5. $\alpha \in (LHS\ or\ RHS)\ \&\ \alpha \in \{0, 1\} \rightarrow -$

Where α is the attributes included in the rule and $\alpha \in LHS$ means that α is an attribute seen on LHS of the interactions, + and − determines the presence of interaction. Among these rules, the most interesting type is the third one. First two types include information about only one side domains, namely left or right hand side. Fourth type has all its attributes set to zero, meaning that if a set of domains does not exists, then interaction occurs. This is very hard to interpret, because it is about rules that depend on the non-existence of some domains. Last type is the negative interaction rules, so they should not be considered as interaction rules.

Negative examples should also be presented to the ID3 learning algorithm. Since there is no database of non-interacting proteins, we have used a method that shuffles the domains and creates artificial proteins while preserving the domain occurrence frequencies within the generated proteins. This provides a better way of creating a negative example set instead of taking the complement of the interaction data, because the resulting proteins are more like natural and a small error rate is expected on classification of these proteins. This is because we don't expect a large number of proteins from a random set to interact with each other. However, the best approach would be to get the results of biological experiments that were carried out to find the protein-protein interactions. Extracting the negative samples from those experiments would give the best results because they are natural data, obtained from biological experiments. Nevertheless, ID3 learning algorithm is known to be robust to errors in the training set. So, we expect that this randomized artificial protein interactions will not contribute to the error too much.

4 Databases Used

DIP [1] database stores information about protein-protein interactions that are confirmed experimentally. Over 17000 interactions are listed in DIP database, which can be used freely for academic purposes. DIP is presented in both html and xml format. Information in the xml file is basically divided into two major sections: nodes and edges. Nodes are presented by their IDs and various other fields like cross-links and features. Edges (which represent the interactions) are listed with two nodes per entry. Number of nodes listed in the database is 6807 and the number of edges listed is 17693.

Yeast Database was created by Uetz et al. [6] and presented in the Curagen's web site in html format. All of the interactions presented here are identified experimentally by high throughput yeast two-hybrid screens on open reading frames of Saccharomyces Cerevisiae genome sequence. Every protein listed in the database is associated with its interacting pairs and also cross-links to other databases, also with a visualization that shows the role of the protein in the whole genetic network.

Pfam is the protein family database [3]. It supports searching by keywords or sequences and retrieves the domains (families) of a given protein. We have used Pfam in order to get the domain decomposition of the proteins, by extracting data from a large text file. This is the swisspfam part of Pfam, which is keyed by the Swiss-Prot [7] names and accession numbers of the proteins.

5 Results and Discussion

23910 interactions were given to the association rule miner and various experiments were carried out with varying support and confidence values. As the support and confidence values get more strict, number of rules found by the algorithm decreases. However, it is important to decide on which support and confidence value pair gives the closest match to a set of rules which consists of logical, valuable rules. For example, number of rules generated given a minimum support of 0.1% and a minimum confidence of 10% is around 130000, which is very high. Indeed, when examined more closely, one can see that most of these rules are trivial, or does not mean much at all, because their dependability measures are very low. In order to find more meaningful rules, one should increase the minimum support and confidence variables. However, then, we face the risk of missing some valuable, but not so frequent rules. Below is a table which shows the change in the number of rules with fixed support and varying confidence and varying support and fixed confidence.

Support	Confidence	Number of Rules
0.5	10	416
0.5	50	398
0.5	90	376
0.1	50	111872
0.5	50	398
0.9	50	4

It is seen from the table above that the number of rules is gradually decreasing as the minimum confidence requirement is increased linearly. The same is true with the minimum support value, but the table proves that the support is a more constraining variable than the confidence value. As the support increases linearly, a rapid decrease in the number of rules is observed.

Still in the search for optimum support and confidence pair for our data, we look for other conclusions. For example, looking at the number of composite rules may give us an idea about the dependability of the resulting rules. Composite rules, by their nature, contain both RHS and LHS domains in either side of the rule. Although we have observed composite rules, it is hard to associate a biologically meaningful explanation for such rules. A careful look at the data, however, leads to an explanation. Assume that the rules $L_1 \rightarrow R_1$ and $L_1 \rightarrow R_1R_2$ have been already produced. Then the algorithm will produce $L_1R_1 \rightarrow R_2$ as well since the following is always true: $support(L_1) \geq support(L_1R_1) \geq support(L_1R_1R_2)$. It was observed that the number of composite rules also gradually decreases with increasing support and confidence requirements. In our experiments, it was observed that beyond 0.2% support and 20% confidence, these rules totally disappeared from the resulting rule sets.

On the other hand, the number of useful rules detected by the ID3 algorithm were far fewer than that of the association rule mining. One of the rules that were detected by the ID3 algorithm was: $PF01423\_L = 1, PF01423\_R = 1 \rightarrow +$. This rule briefly says that if both the right hand side and the left hand side proteins contain PF01423 domain, they will interact. Indeed, this domain is a domain that is specific to *Sm proteins*. In the literature, Sm proteins are known to be involved in mRNA splicing. Seven Sm Proteins form a complex around the Sm site to splice the mRNA. This information verifies that the rule is indeed, correct.

Following are two of the rules that were detected by association rule mining method:

$$PF00227\_L \rightarrow PF00227\_R$$
$$PF00069\_L \rightarrow PF00134\_R$$

PF00227 of rule 1 is the Proteosome A-type and B-type domain annotated in Pfam. It is also claimed that members of this domain form a large ring based complex, which verifies that proteins that contain this domain interact with each other. Rule 2, on the other hand, is related with two distinct domains: PF00134 is the cyclin, N-terminal domain and PF00069 is the protein kinase domain. It is mentioned in Pfam that cyclins regulate the cell division cycle in eukaryotes and protein kinases form a complex with them.

6 Conclusion

In this paper, we have described and used two different methods to find rules about protein-protein interactions in domain decomposition level. It was observed that some of the rules found out by the techniques were indeed true and interactions among these domains were mentioned in the literature. It was also observed that not both techniques produce the same set of rules. In fact, association rule mining outperforms the ID3 method in the number of rules generated. However, it is difficult to find the correct support and confidence values for the association rule mining algorithm.

As a future work, different features can be incorporated along with the domain decomposition of the proteins. We believe, for example, motifs and amino acid patterns, as well as expression profiles and micro-array data would yield to interesting rules about protein-protein interactions. From the biological side, rules that are generated frequently with reasonable support and confidence value pairs can be checked with laboratory experiments. By this way, we can understand if the method can discover novel protein-protein interactions.

References

1. Ioannis Xenarios, Lukasz Salwinski, Xiaoqun Joyce Duan, Patrick Higney, Sul-Min Kim and David Eisenberg: DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Research Vol. 30 No. 1 303–305, 2002
2. Rakesh Agrawal, Ramakrishnan Srikant: Fast Algorithms for Mining Association Rules. Proc. 20th Int. Conf. Very Large Data Bases, VLDB 1994
3. Bateman A, Birney E, Cerruti L, Durbin R, Etwiller L, Eddy SR, Griffiths-Jones S, Howe KL, Marshall M, Sonnhammer EL: The Pfam Protein Families Database. Nucleic Acids Research, Vol. 30 276–280, 2002
4. R. Agrawal, T. Irnielinski, and A. Swami: Mining Association Rules between Sets of Items in Large Databases. Proceedings of A CM SIGMOD, 207–216, May 1993
5. T. Oyama, K. Kitano, K. Satou and T. Ito: Extraction of knowledge on protein-protein interaction by association rule discovery, Bioinformatics. Vol. 18, no. 5 2002
6. Uetz et al: A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. Nature, Vol. 403. Page 623–627, 2000
7. Boeckmann B., Bairoch A., Apweiler R., Blatter M.-C., Estreicher A., Gasteiger E., Martin M.J., Michoud K., O'Donovan C., Phan I., Pilbout S., Schneider M: The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res, Vol. 31. Page 365–370, 2003
8. Christian Borgelt's Software Page: http://fuzzy.cs.uni-magdeburg.de/˜borgelt/