International Journal of Chemical, Materials and Biomolecular Sciences
ISSN: 2415-6620
Vol:2, No:8, 2008

Observation of the Correlations between Pair Wise Interaction and Functional Organization of the Proteins, in the Protein Interaction Network of *Saccaromyces Cerevisiae*

N. Tuncbag, T. Haliloglu, and O. Keskin

***Abstract*—**Understanding the cell's large-scale organization is an interesting task in computational biology. Thus, protein-protein interactions can reveal important organization and function of the cell. Here, we investigated the correspondence between protein interactions and function for the yeast. We obtained the correlations among the set of proteins. Then these correlations are clustered using both the hierarchical and biclustering methods. The detailed analyses of proteins in each cluster were carried out by making use of their functional annotations. As a result, we found that some functional classes appear together in almost all biclusters. On the other hand, in hierarchical clustering, the dominancy of one functional class is observed. In brief, from interaction data to function, some correlated results are noticed about the relationship between interaction and function which might give clues about the organization of the proteins.

***Keywords*—**Pair-wise protein interactions, DIP database, functional correlations, biclustering.

I. Introduction

PROTEINS are large molecules responsible for executing and regulating various biological functions. Although some protein structures can be functional alone, most of them have to associate with other proteins to act in the biological processes. In other words, they perform their functions by interacting with other proteins. The complex functions in the biological systems are a result of the large network of the proteins formed by pair wise protein – protein interactions. As a result, the network of interactions between the proteins increases the understanding of protein functions and this network controls the lives of cells [1]. Moreover, protein interaction networks provide functional information about the uncharacterized proteins and remote similarities between proteins [2].

Nurcan Tuncbag is M. S. student in Computational Science and Engineering, Koc University 34450, Sariyer, Istanbul, Turkey (e-mail: ntuncbag@ku.edu.tr).

Prof. Turkan Haliloglu is with Chemical Engineering Department, Bogazici University, Bebek, Istanbul, Turkey (e-mail: halilogt@ boun.edu.tr).

Assist. Prof. Ozlem Keskin is with Chemical and Biological Engineering Department, Koc University, 34450, Sariyer, Istanbul, Turkey (e-mail: okeskin@ku.edu.tr).

There are databases, which catalog the interactions between the proteins, provide quick access to the experimental interaction data and also to the large scale properties of these biological networks. One of the interaction databases is the Database of Interacting Proteins (DIP) [3]. In addition to documenting experimentally determined pair wise physical interactions, DIP combines information from a variety of sources to create a single, consistent set of protein-protein interactions. The data within the DIP can be extracted both manually and computationally [3]. DIP interaction participating proteins have a special identity number as "DIP:nnnN". Besides this accession number, DIP provides cross-references to the three major sequence databases, SWISSPROT, PIR and GenBank. By the help of the cross references from DIP to other sequence databases, it is possible to obtain information about general aspects of the proteins. In this way, from the pair wise interaction data in DIP, the functional correlation of two interacting proteins can be found by the cross-references [3,4].

The Gene Ontology project (GO) gives consistent descriptions of gene products in different databases. The GO annotations describe the gene products from three ways: (i) *cellular component* is the component of the cell; for instance, ribosome, nucleus etc.; (ii) *biological process* is series of events accomplished by one or more ordered assemblies of molecular functions; for example cellular physiological process or signal transduction; (iii) *molecular function* describes activities, such as catalytic or binding activities at the molecular level. The GO terms goes from broad terms to more specific terms. For example, "binding" is a broad GO term, at the second level the GO term below binding, "protein binding" give more specific information. Each term in GO have a unique numerical identifier like (GO:nnnnnnn) [5].

In this study, in order to analyze the functional correlations and organization of the proteins, we started with the interaction data in DIP database and continued with functional description of the proteins. For this purpose, Gene Ontology project is used to annotate the functional correspondence. Our work can give some clues about the relationship between interaction and function.

In the Table I, the name of the used websites, its URL's and contents are shown.

TABLE I
THE WEBSITES USED IN THE PROJECT

Name	URL	Content
Database of Interacting Proteins (DIP)	dip.doe-mbi.ucla.edu	Pair wise protein-protein interactions.
SWISSPROT	www.expasy.org/sprot	Sequence database
Gene Ontology (GO)	www.geneontology.org	Describes how gene products behave in a cellular context.

II. METHODS AND RESULTS

The main focus of this paper is to find the set of correlated proteins in the protein-protein network of *S. cerevisiae*; in this way, to observe the correlation between the pair wise physical interactions between *S. cerevisiae* proteins and functional organization between them. For this purpose, a novel method is used in order to find the cross correlations between the *S. cerevisiae* proteins. The flowchart of the observation process is shown in Fig. 1.

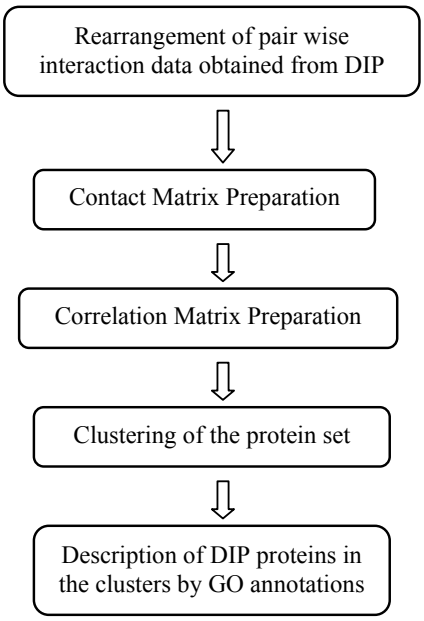

Fig. 1 The flowchart of the project in order to observe the correlation between protein function and interaction

The used pair wise interaction data set was obtained from the DIP in December 11, 2005. The data set contains the pair wise CORE interaction data of the yeast proteins. CORE interaction means that the interaction data verified by one or more computational verification methods [3]. In the data set, there are totally 2635 proteins and 6342 core interactions observed among these proteins.

The network of protein interactions are represented as an undirected graph with proteins as nodes and interactions as undirected edges [2]. In the algorithm of the project, firstly, the pair wise protein-protein interaction data available in DIP for yeast is rearranged to prepare a matrix to represent the interaction data among the proteins. For this purpose, each protein is referenced to an integer identifier. The protein identifiers are also used as indexes to form the contact matrix (A). By using the model below, the contact matrix is prepared. According to this model, it is controlled whether the protein *(i)* interacts with another protein *(j)* or not. If *i* interacts with *j*, 1 is inserted at the *ij*th element of the matrix, if not, 0 is inserted at this element. And the i^{th} element of the diagonal of the matrix A is the negative of the i^{th} row sum [6].

If the protein *(i)* interacts with protein *(j)* then;

$$A_{ij} = 1 \tag{1}$$

If the protein *(i)* does not interact with protein *(j)* then

$$A_{ij} = 0 \tag{2}$$

The diagonal of the contact matrix is

$$A_{ii} = -\sum_{j=1}^{n} A_{ij} \tag{3}$$

If there are N proteins in the interaction network, then the contact matrix is NxN in size and has N eigenvalues and N corresponding eigenvectors. However, some of the eigenvalues of this matrix are zero because of its sparsity. Because the contact matrix is singular and do not have inverse, the pseudo inverse of the matrix is taken. The pseudoinverse (A^+) of a matrix is a generalization of the inverse matrix. The computationally simplest way to calculate the pseudoinverse of a matrix is using singular value decomposition (SVD). If $A = U \Sigma V^*$ is the singular value decomposition of A, then the the psuedoinverse of A is $A^+ = V\Sigma^+U^*$. For a diagonal matrix such as Σ, which consists of the singular values of matrix A, the pseudoinverse of this diagonal matrix is the reciprocal of each non-zero element on the diagonal [7].

The size of the contact matrix in the project is 2635 x 2635 – because there are 2635 proteins in the dataset – and the sparsity of it is 0.9978, which means that 99.8 % of the matrix elements are zero. The sparsity pattern of the contact matrix for yeast core interaction data in DIP is shown in Fig. 2.

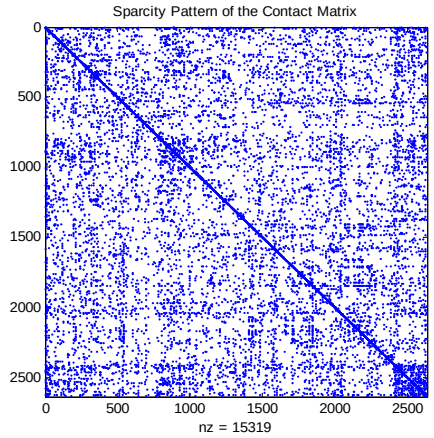

Fig. 2 Sparsity pattern of the contact matrix for the pair wise interaction data where nz is the number of non zero elements

The pseudo inverse matrix gives the cross-correlations between the proteins. However, the pseudo inverse of the contact matrix is not normalized yet. After taking the pseudo inverse of the contact matrix, the next step is to prepare the normalized cross-correlation matrix between the proteins [6]. By normalization of pseudo inverse matrix, the cross-correlation matrix is found. Cross-correlations between the DIP proteins are calculated as in Eq. (4). The matrix C gives the normalized cross correlations between the proteins [6].

$$C(i,j) = \frac{A_{ij}^{+}}{\left\{A_{ii}^{+} \cdot A_{jj}^{+}\right\}^{1/2}} \qquad (4)$$

A_{ij}^{+} means the ij^{th} element of the pseudo inverse matrix.

As a result, all the diagonal elements of the cross-correlation matrix (C) are equal to 1 which means that the protein *(i)* is 100 % correlated with itself. In the cross-correlation matrix, the correlation value of each element changes between – 1 and 1. (–) correlation values represent anti-correlation between two proteins, 0 represents no correlation between them and (+) correlation values represents how much correlated are two proteins. To perform all these steps, Python 2.4 Programming Codes has been used.

After obtaining cross correlations between the proteins, the next aim is the clustering of the proteins in the correlation matrix according to their correlation values. For first trial in the project, hierarchical clustering was used. By "clusterdata" function in MATLAB, the proteins were clustered hierarchically. This function first computes the Euclidean distance between pairs of objects in the correlation matrix. Then, it creates a hierarchical cluster tree, using the Single Linkage algorithm, and finally, constructs clusters from this hierarchical cluster tree. The cluster numbers are found according to the cutoff value. The cutoff value is a threshold for cutting the hierarchical tree generated by linkage into clusters [8]. The optimum cluster number is found by the graph "cutoff value vs. cluster number", shown in Fig. 3. In the graph, while the cutoff value increases, the number of clusters decreases and goes to 1 cluster. According to this graph, the optimum cutoff value is chosen as 1.154. The number of clusters for this optimum cutoff value is 507.

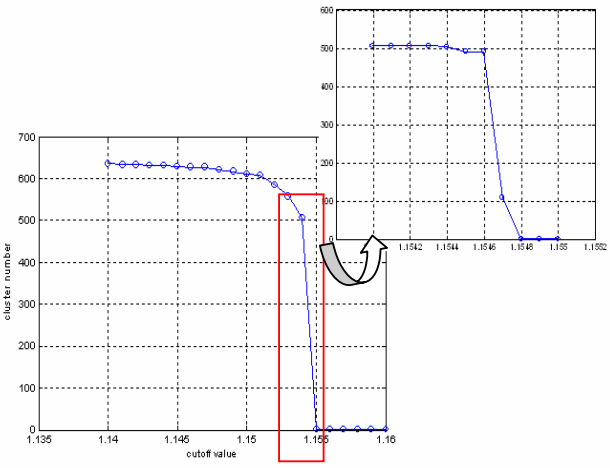

Fig. 3 the cutoff value vs cluster number graph of hierarchical clustering by MATLAB

In the hierarchical clustering results, the cluster sizes are small; also some of them have only one member. The member number distribution of the clusters in hierarchical clustering for this dataset is shown in Fig. 4. In the analysis of the clusters in the functional perspective, the clusters which have less than 5 members are eliminated. As a result, cluster numbers decrease from 507 clusters to 93 clusters. The cluster size distribution after elimination of the redundant clusters is shown in Fig. 5.

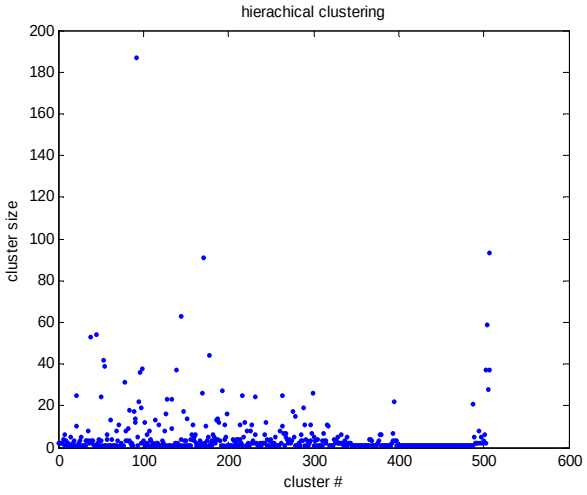

Fig. 4 Cluster size distribution for hierarchical clustering

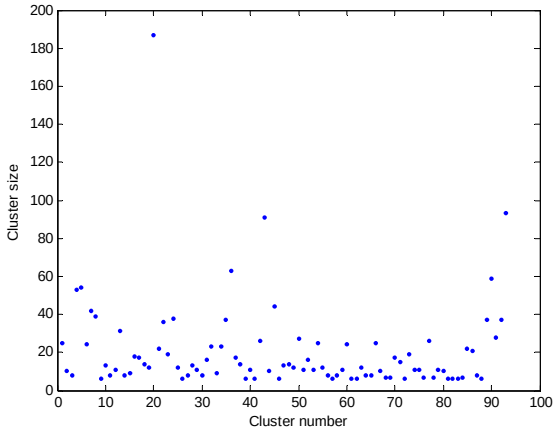

Fig. 5 Cluster size distribution for hierarchical clustering after elimination of the redundant clusters

Hierarchical clustering is one of the mostly used methods for clustering a data set in biological systems. However, in biological systems, a protein can function in more than one process. In other words, one protein can be put more than one cluster. Because of this situation, biclustering method seems more appropriate for biological systems and for clustering these proteins. Biclustering algorithm does not force the proteins to belong to one cluster. To bicluster the proteins in the dataset effectively, the EXPANDER software [9] was used. EXPANDER is a package for the analysis of gene expression data, contains various data analysis algorithm implementations. One of them is biclustering analysis. The biclustering tool of the EXPANDER uses SAMBA algorithm to bicluster the data set [9]. The detailed information about biclustering and EXPANDER software can be found in http://www.cs.tau.ac.il/~rshamir/expander/expander.html. In this work, the normalized cross-correlation matrix (C) is biclustered. Firstly, the matrix is loaded in the EXPANDER software, and then the SAMBA algorithm is runned. As a result, there is 344 biclusters found. However, lots of clusters have same members at high ratios. For example, some biclusters have 70 % or more similarity, means that 70% of members of a biclusters are same with another bicluster. Because of this situation, the biclusters are associated according to their similarity ratios. As a result, for 70 % similarity, the bicluster number decreases from 344 to 222. The cluster size distribution after association of the clusters is shown in Fig. 6.

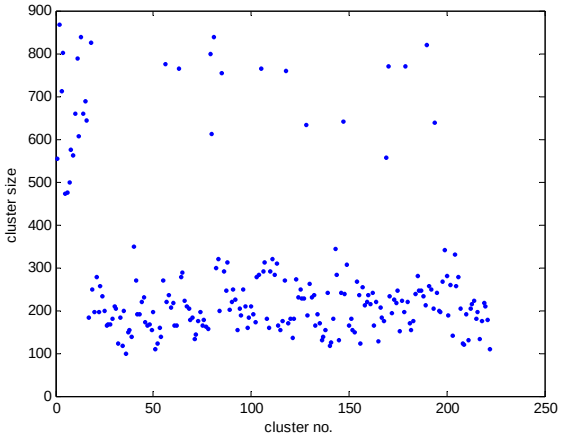

Fig. 6 Cluster size distribution of the biclusters

After the correlation matrix is clustered by hierarchical clustering and by biclustering, the proteins in the clusters and biclusters are interpreted according to the molecular functions, whether there are functional correlations between the proteins in the clusters. For this purpose, each protein in the clusters are described with the GO annotations and analyzed whether there is functional relationship between them. The GO ID's of each protein is found by cross references from DIP to GO. There is no direct way to get GO annotations from DIP database as shown in Fig. 7. Firstly, the cross-reference between the DIP database and SWISSPROT is used. Each DIP name is changed by SWISSPROT ID's. However, the SWISSPROT ID's of 306 of the 2635 proteins in the dataset are not found in the cross reference between DIP and SWISSPROT. Then, the cross-references between the SWISSPROT and GO annotations are found. By this way the function, in which the protein participate is found for the proteins in each clusters. The first level functions are used from GO annotations. At the first level GO annotations, there are 19 different function classes which are shown in Table II. 12 of these functional categories are occupied by the proteins in the core interaction data set. Each protein is assigned to one or several of the 12 functional classes.

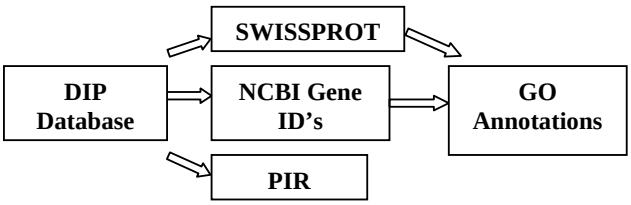

Fig. 7 Cross – reference from DIP database to GO annotations

The huge amounts of the proteins in the interaction dataset have binding and catalytic activity as seen in Table II. They are excluded from the functional categories, since they would over amplify the results. The functional categories include chemoattractant activity (#5), chemorepellant activity (#6),

energy transducer activity (#7), nutrient reservoir activity (#11), obsolete molecular function (#12) and triplet codon-amino acid adaptor activity (#19). When we focus on the functions, the proteins in the data set are occupied in the functional classes of antioxidant activity (#1), chaperone regulator activity (#4), enzyme regulator activity (#8), motor activity (#10), protein tag (#13), signal transducer activity (#14), structural molecule activity (#15), transcription regulator activity (#16), translation regulator activity (#17) and transporter activity (#18). In the Fig. 8, the functional distribution of the all yeast proteins in the data set are shown.

TABLE II
THE NUMBER OF FUNCTION IN THE DATA SET

GO Function (first level)	ID	# of proteins participate at that function
antioxidant activity	1	6
binding	2	1512
catalytic activity	3	1092
chaperone regulator activity	4	2
chemoattractant activity	5	0
chemorepellant activity	6	0
energy transducer activity	7	0
enzyme regulator activity	8	152
molecular function unknown	9	0
motor activity	10	12
nutrient reservoir activity	11	0
obsolete molecular function	12	0
protein tag	13	3
signal transducer activity	14	41
structural molecule activity	15	139
transcription regulator activity	16	136
translation regulator activity	17	37
transporter activity	18	155
triplet codon-amino acid ad. act.	19	0

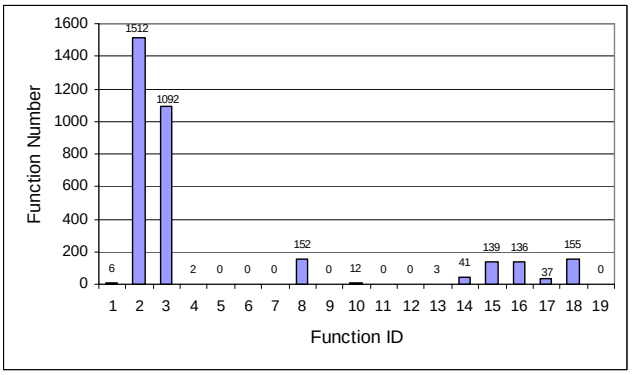

Fig. 8 functional distribution of the all yeast proteins in the data set

After each protein is described by its functional annotations, the hierarchical clusters and biclusters are checked separately, whether they are correlated or they are not. When the functional annotations are analyzed (excluding the protein binding (#2) and catalytic activity (#3), since because they are in all clusters at high ratios normally) generally almost in every bicluster the functional classes, enzyme regulator activity (#8), signal transducer activity (#14), structural molecule activity (#15), transcription regulator activity (#16), translation regulator activity (#17) and transporter activity (#18), take place as blocks. However, in the clusters of hierarchical clustering, these functional class blocks are not seen. The functional classes of binding (#2) and catalytic activity (#3) exists dominantly in all clusters in hierarchical clustering like biclusters, but we do not see the same functional behavior of the proteins and 6 functional category blocks. Generally, in hierarchical clusters, one functional class is dominated except the classes #2 and #3. Both in small hierarchical clusters and small biclusters, one functional class is dominant and separate from others. To be more specific about it, we choose one of the 222 biclusters.

The bicluster number 40 is chosen to be observed more detailed. This bicluster had 350 proteins; because one protein can be assigned one or several functional classes, there are formed totally 473 functions and 84 % of these functions are in the class of binding (#2) and catalytic activity (#3).

TABLE III
THE FUNCTIONAL CLASSES OF THE BICLUSTER # 40

Functional Class	# of proteins participate at that function
binding	210
catalytic activity	188
enzyme regulator activity	15
signal transducer activity	4
structural molecule activity	12
transcription regulator activity	15
translation regulator activity	12
transporter activity	17

When binding and catalytic activity are disregarded, again the 6 functional groups are observed together in bicluster 40 as in the rest of the data set. For biclustering results, we can conclude that these 6 functional groups are working collectively in the yeast. Because we started from interaction data, it can be suggested that these functional grouping of the proteins shows the correlation between interaction and function. In Fig. 8, the partition of the functional categories in bicluster 40 is shown.

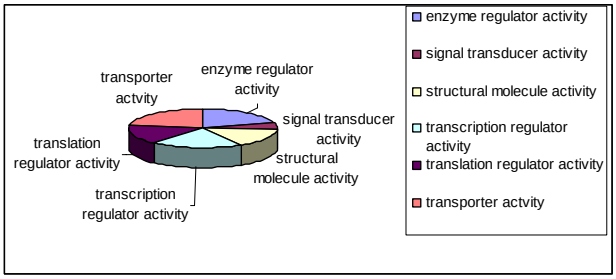

Fig. 9 The graph of the 6 functional classes of bicluster 40

The same procedure is followed also for the hierarchical clusters. The cluster number 20 is chosen for this procedure. This cluster has 187 members. As seen in Table IV, there was only one functional class except binding and catalytic activity. The functional class transcription regulator activity (#16) is the single category, when category #2 and #3 are disregarded.

TABLE IV
THE FUNCTIONAL CLASSES OF THE CLUSTER # 20

Functional Class	**# of proteins participate at that function**
binding	95
catalytic activity	105
transcription regulator activity	19

The complete list of the hierarchical clusters and biclusters with its functional annotations is available at home.ku.edu.tr/~ntuncbag/yeastclusters.

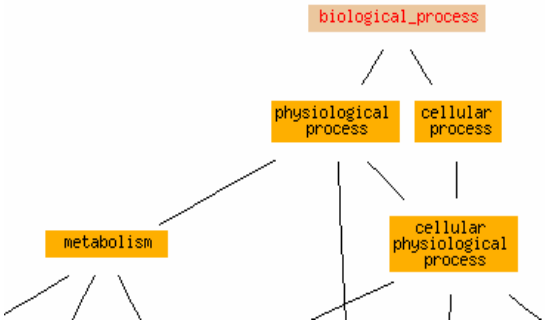

Fig. 10 GO process annotations tree of the bicluster 40

By the usage of the server in Yeast Genome Database (www.yeastgenome.org), the GO annotations tree for the bicluster 40 are drawn from the process side. For bicluster 40, the biological processes of them is identified and it is observed that only 2 of the 12 processes are occupied by the proteins in the bicluster 40 as seen in Fig. 10 which are physiological process and cellular process. When the tree is examined at the low levels, we see that the most of the proteins in this cluster is participating into the rRNA metabolism. This situation shed light on the hypothesis that the interaction is related to molecular function and process.

III. CONCLUSION

The starting point of this paper was the hypothesis that interacting proteins have a high probability to belong to same functional class. For this purpose, after obtaining cross correlations between yeast proteins from interaction data, two clustering methods were used and at the end, two different results were obtained. As a result of biclustering, we observed the collective existence of same functional classes. Moreover, after observation of one bicluster in the view of processes, dominancy of one process was observed in the entire of the bicluster. On the other hand, in hierarchical clustering the dominancy of one functional class is noticed, especially in the small sized clusters.

In the future, after observation of the biological processes, which the yeast proteins are participating, for each cluster and bicluster, the results would be clearer.

ACKNOWLEDGMENT

We thank Attila Gursoy for his useful discussions during this study. Also, we thank The Scientific and Technological Research Council of Turkey (TUBITAK).

REFERENCES

[1] S.-H. Yook, Z. N. Oltvai, A. L. Barabasi "Functional and topological characterization of protein interaction networks" in *Proteomics*, 2004, pp. 928–942.
[2] A.-L. Barabasi, Z. N. Oltvai "Network Biology: Understanding the Cell's Functional Organization" in *Genetics*, 2004, pp. 101–111.
[3] L. Salwinski, C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, D. Eisenberg "The Database of Interacting Proteins: 2004 update" in *Nucleic Acids Research*, 2004, pp.D449-51
[4] I. Xenarios, L. Salvinski, X. J. Duan, P. Higney, S.-M. Kim, D. Eisenberg "DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions" in *Nucleic Acids Research*, 2002, pp. 303–305.
[5] Gene Ontology Consortium, "Gene Ontology (GO) project in 2006 ", in *Nucleic Acids Research*, 2006, pp. D322–326.
[6] O. Keskin, "Comparison of Full-atomic and Coarse-grained Models to Examine The Molecular Fluctuations of c-AMP Dependent Protein Kinase", in *Biomolecular Structure and Dynamics*, 2002, pp. 1-13.
[7] L. N. Trefethen, D. Bau, Numerical Linear Algebra. Washington: Siam, 1997, ch.2.
[8] D. Raicu. (2004, February). Statistics with MATLAB. Available: http://facweb.cs.depaul.edu/Dstan/teaching/tutorials/Statistics%20with%20Matlab.pdf
[9] R. Shamir, A. Maron-Katz, A. Tanay, Chaim Linhart, I. Steinfeld, R. Sharan, Y. Shiloh, R. Elkon " EXPANDER – an integrative program suite for microarray data analysis", in *BMC Bioinformatics*, 2005, 6:232