Data Pre-Processing Methodology for Computational Experiment in Analyzing Protein Interaction Networks

Jamaludin Sallim[1]
Member, IEEE
jamals@cs.usm.my

Rosni Abdullah[2]
rosni@cs.usm.my

Ahamad Tajudin Khader[3]
tajudin@cs.usm.my

[1,2,3]*School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia.*

Abstract

This paper describes a process flow of data pre-processing for computational experiment in analyzing protein interaction networks. The aim of this process flow is to fulfill the requirement of our graph-based algorithm's solution and to ensure that the representation of protein interaction network which consist of nodes and edges are accurate. The raw data is obtained from one of protein interaction database called Database of interacting Proteins (here-on known as DIP)). We describe briefly the DIP and since our algorithm requires the distance matrix as an input; a few important steps need to be taken to process the data. We limit this paper to data pre-preprocessing only and the algorithmic process is not discussed.

1. Introduction

Understanding protein interaction networks is one of the important problems in bioinformatics and computational biology. Analyzing and studying the protein interaction networks (here-on known as PIN) will provide valuable insights into the protein functions and how to classify and cluster them. Protein-protein interactions is the association of protein molecules and protein interaction networks is the study of these associations from the perspective of networks. If protein A interacts with protein B, it means that scientifically they are exhibiting similar functions. The prediction of protein-protein interaction was experimentally performed by researchers with various scientific and computational methods. Based on G. Pandey et. al (2007), even though protein interaction networks are one of promising types of biological data for discovering protein functions and clustering, it is known that these networks are both incomplete and inaccurate.

We describe protein interaction network concept by giving an example of general protein interaction networks constructed in Figure 1.

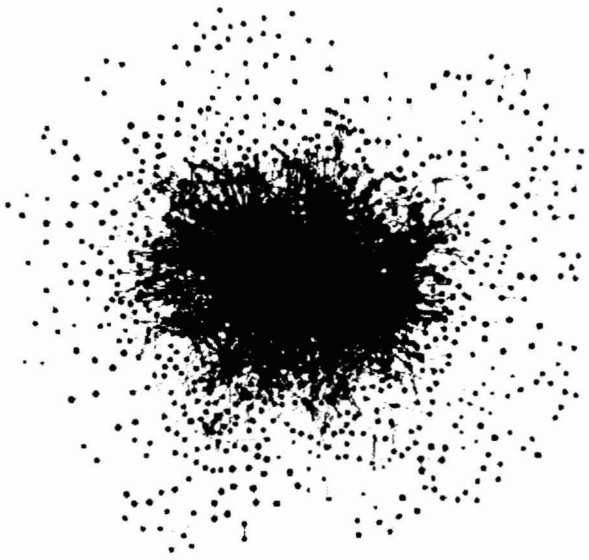


Figure 1. The protein interaction constructed on 11,000 yeast interactions involving 2,401 proteins [2,3].

In this paper we only present the process flow of data pre-processing, starting from extracting and parsing the raw data from DIP tables, loading them to a file using igraph (see methods section) to obtain adjacency matrix and finally the data converted by the distance matrix a_{ij} and the distance matrix will be the input of our algorithm.

2. Representing Protein Interaction Networks

Biological networks are usually represented using various theoretic formalisms and for the protein interaction networks, they are usually represented in a graph format. In a graph-based representation, nodes

978-0-7695-3136-6/08 $25.00 © 2008 IEEE
DOI 10.1109/AMS.2008.91

correspond to proteins and edges correspond to interacting proteins. A graph is usually denoted by G, or by $G(V,E)$, where V is the set of vertices or nodes and $E \subseteq V \times V$ is the sets of edges of G. Nodes joined by an edge are said to be adjacent. A neighbor of a node v is a node adjacent to v. From this graph, we obtain the adjacency matrix and finally we transform the adjacency matrix to distance matrix in order to fulfill our algorithm approach (the detail of these processes will be discussed in Methods section). In our approach, the distance in the distance matrix actually represents how similar the set of the different proteins interaction partners are. Therefore, if protein A and protein B both interact with the same other proteins then the distance will be 0. For instance, if protein A interacts with protein X, Y and Z and protein B also interacts with protein X, Y and Z then the distance between protein A and B will be 0. If protein B only interact with protein X, Y and other protein which protein A not interact with, the distance between protein A and protein B will be > 0. If they interact with completely different sets of proteins then the distance will be large where the exact number will be depend on equation we use. We describe this concept in the following figures. (The to-and-fro arrow shows the interaction between proteins).

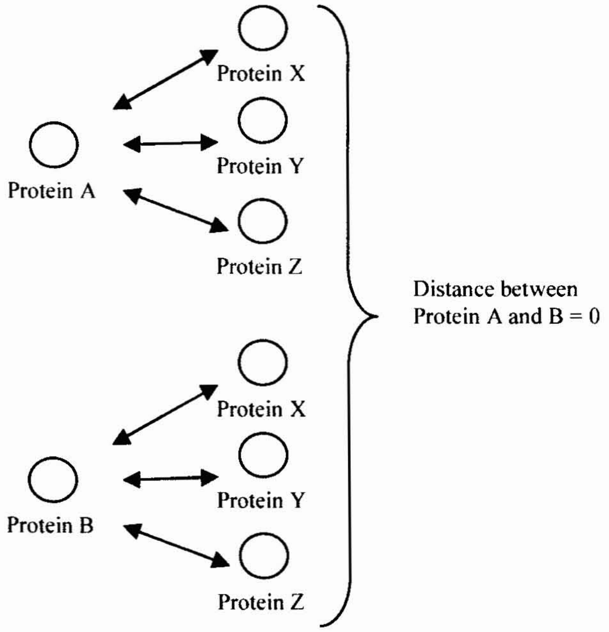


Figure 2. The interaction of protein A and B with similar set of interacting partners.

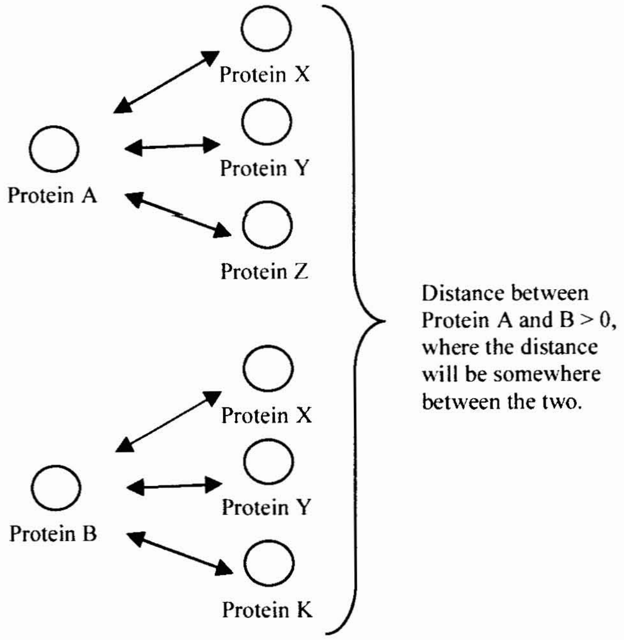


Figure 3. The interaction of protein A and B with similar and dissimilar set of interacting partners.

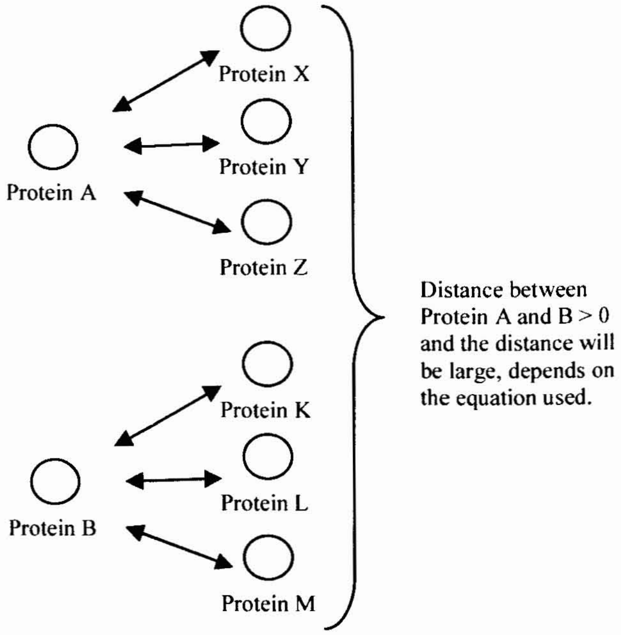


Figure 4. The interaction of protein A and B with dissimilar set interacting partners.

3. Tools

Our data extraction and parsing were implemented in the C programming language and was run on Intel®™ 2 processor 1.83GHz with 1.00 GB RAM installed with Microsoft Windows XP Professional operating system. We use *igraph*, a C-library for creating and manipulating graph [5] to build the graph for our interaction data representation. *igraph* is. We load the extracted and parsed protein interaction data to a file using igraph. Then we obtain the adjacency matrix and distance matrix. Finally, this distance matrix will be the input to the algorithm of our experiment. This is the most crucial part in data pre-processing because our algorithm highly depends on the accuracy of protein interaction networks representation. To ensure the representation of protein interaction network that connecting nodes is accurate, we also use igraph to measure the accuracy (*see 4.5 in methods section*).

4. Process Flow

In this section, we discuss the data sources, data extracting & parsing process and the protein interaction data sets used in our work. The measurement of accuracy and obtaining adjacency matrix using igraph and transforming approach will also be discussed.

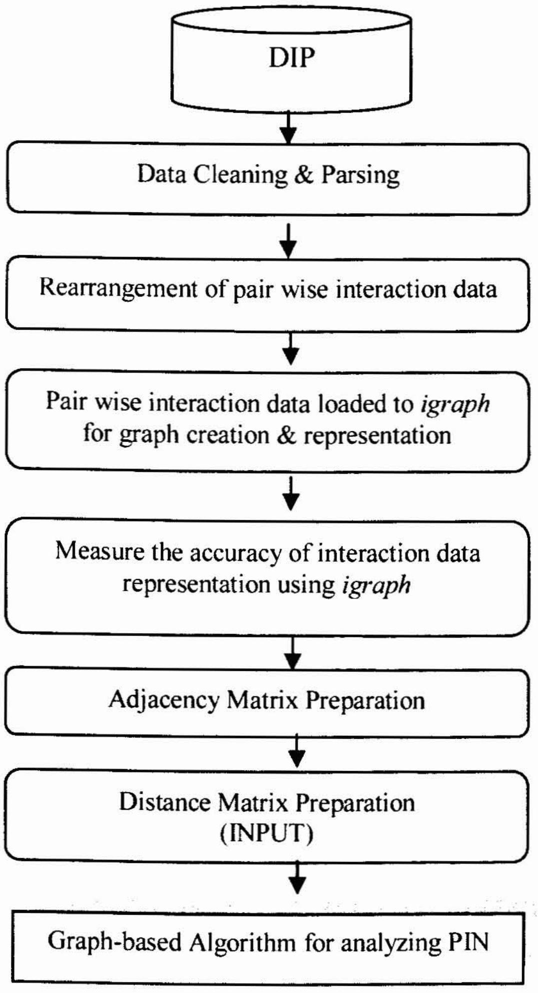

DIP
Data Cleaning & Parsing
Rearrangement of pair wise interaction data
Pair wise interaction data loaded to *igraph* for graph creation & representation
Measure the accuracy of interaction data representation using *igraph*
Adjacency Matrix Preparation
Distance Matrix Preparation (INPUT)
Graph-based Algorithm for analyzing PIN

Figure 5. The process flow of data pre-processing.

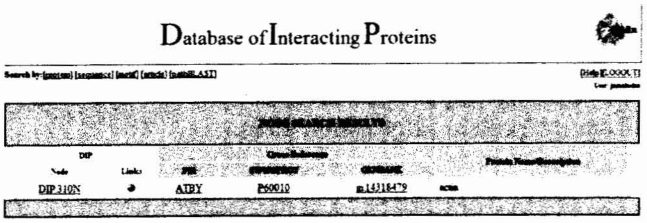
4.1. Data Sources

Database of Interacting Proteins (DIP)
For our experiment, we chose the protein interaction database called DIP. It was initially developed and maintained by the Molecular Biology Institute at UCLA to store and organize information on binary protein-protein interactions. It contains experimentally verified protein-protein interaction data. The information is curate from scientific literature and archives, by biology experts and computational automated methods. DIP provides the gene name, description, enzyme code, cellular localization as well as cross references to other protein sequence databases such as Swiss-Prot, PIR and GenBank. The interaction information includes the ranges of amino acids, protein domains of the interacting proteins and the experiments used to detect the interaction. Table 1 illustrates DIP database statistics obtained from DIP website as of November 2007. [4].

Table 1: DIP database statistic obtained from DIP website

Number of proteins	19490
Number of organisms	161
Number of interactions	56186
Number of experiments describing an interactions	64208
Number of data sources (including articles)	3291

From the online database, we also perform a sample query by searching the protein *actin* present in *Saccharomyces Cerevisiae* (yeast). Figure 2 shows the result obtained from DIP for this searching query. In this result, they also provide cross-references to PIR, SWISS-PROT and GenBank with a short name description and name of protein.


Database of Interacting Proteins

Search by: [protein] [sequence] [motif] [article] [pathBLAST]	[Help] [LOGOUT]

DIP Node	Links	PIR	SWISSPROT	GENBANK	Protein Name/Description
DIP 310N	●	ATBY	P60010	g14318479	actin

Figure 6. Searching query results from DIP

By clicking on the ***Node Link,*** it will provide more detail on the cross-reference information including links to Protein Domain Database (PRINTS), Protein Families (Pfam) database among others. This result also provides a link to visualize graphically the protein interaction network as shown in Figure 3. The red circle represents starting node, the oranges circle represent the first-shell nodes and the yellow circles represent the second-shell nodes. In this image, only edges linking the first-shell and the second- shell are drawn. Each interaction is given a DIP edge identification number, which can be referred in searching query results. The text *"Lsb1p"* is referring to the protein name for the node when we put the cursor on the respective node.

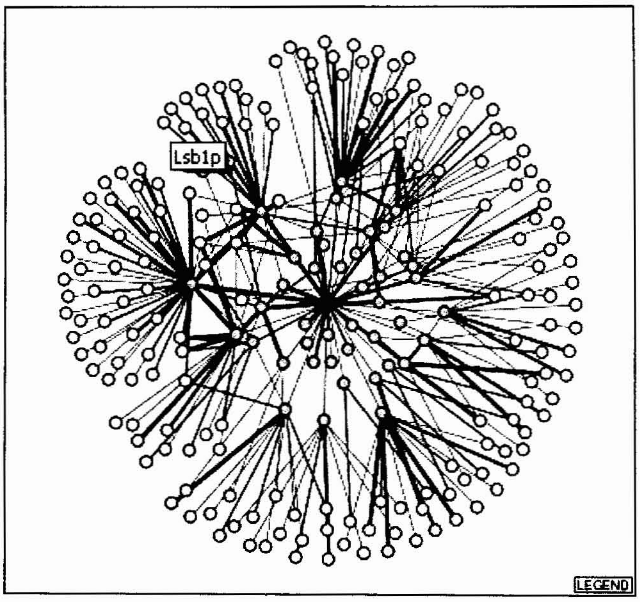


Figure 7. DIP protein interaction network.

DIP database is composed of three linked tables: a table of protein information, a table of protein-protein interactions and a table describing experiments detecting the protein-protein interaction. Figure 4 shows the schematic tables that contain the following information:[6].

i) The protein information table contains protein identification codes from SWISS-PROT, PIR and GenBank sequence database.

ii) The interaction table describes proteins that interact from the protein information table.

iii) The experimental article table details the experiments used to detect the interactions from the interaction table and their associated literature citations.

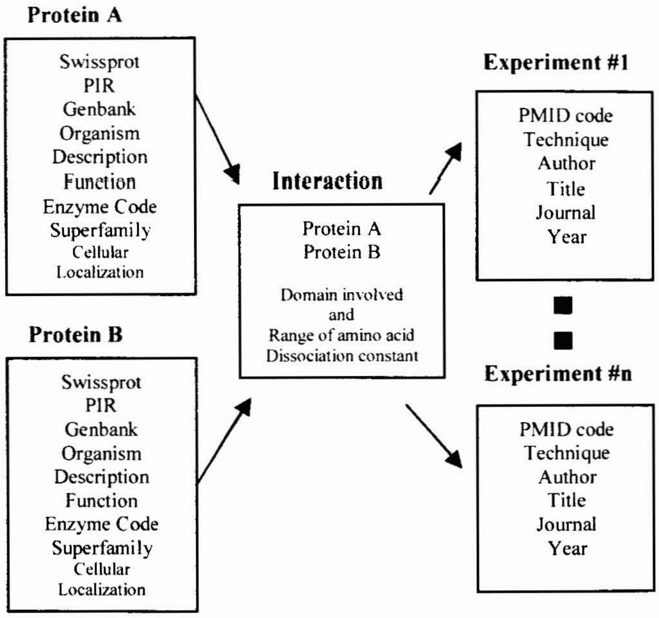


Figure 8. The relational structure of DIP [5]

4.2. Protein Interaction Data Sets

We downloaded the high-throughput protein-protein interaction networks of budding yeast (*Saccharomyces Cerevisiae*) which contained 4920 protein-protein interaction. The reason we choose this data sets is to give us the ability to conduct a general study of protein interaction networks using our graph-based algorithm.

4.3. Protein Interaction Data Parsing

Extracting PPI Data from DIP

The downloaded data was stored in the MASM Listing format. We view this file using MS Visual Studio editor. From this file, it shows that each protein node has an unique DIP id of the form DIP:#N and interaction edge has the form of DIP:#E, where # is a number. It also shows the interaction partner in the fourth column. For protein interaction networks, the most important information is a node ID, interaction partner and interaction id.. Figure 9 illustrates information stored in MASM Listing format and viewed by using Visual Studio.

```
DIP:2551N  AAC1   YMR056C DIP:1189N APG12  YBR217W
DIP:11374E
DIP:2551N  AAC1   YMR056C DIP:7726N BIO5   YNR056C  DIP:56132E
DIP:2551N  AAC1   YMR056C DIP:1330N LSM1   YJL124C  DIP:3267E
DIP:2551N  AAC1   YMR056C DIP:4449N PUF3   YLL013C  DIP:6745E
DIP:2551N  AAC1   YMR056C DIP:2425N RAD3   YER171W DIP:13079E
```

Figure 9. Interaction data from DIP in the MASM Listing format (for the first five records only).

We remove the unnecessary information from this file and the required information was exported to an array-file. At this stage we also assign the new code id that will be used by *igraph.* We use C programming to perform all these processes. Table 2 shows the array-based file which only contain the required information.

Table 2. An array-based file containing the required information.

Protein _ID	igraph _code	Partner	igraph _partner code	Interaction _id	Array Map
DIP:310N	0	DIP:213N DIP:973N DIP:6340N DIP:5904N DIP:860N . .	1 2 3 4 5 . .	DIP:1147E DIP:1175E DIP:45002E DIP:15127E DIP:1180E . .	0,1 0,2 0,3 0,3 . .

4.4. Load protein interaction data to igraph

After task (4.3) done, we pass this data to igraph. Figure 11 illustrate igraph.

```
#include <igraph.h>

int main(void) {
   igraph_t graph;
   igraph_vector_t v;
   igraph_vector_t result;
   igraph_real_t edges[] = { 0,1. 0,2. 0,3. 0,4. 0,5. 0,6. 0,7. 0,8. 0,9. 0,10,
                             0,11. 0,12. 0,13. 0,14. 0,15. 0,16. 0,17. 0,18. 0,19.
                             0,20.0,21. 0,22. 0,23. 0,24. 0,25. 0,26. 0,27. 0,28.
                             0,29. 0,30. 0,31. 0,32. 0,33. 0,34. 0,35. 0,36. 0,37.
                             };

   igraph_vector_view(&v, edges, sizeof(edges)/sizeof(double));
   igraph_create(&graph, &v, 0, IGRAPH_UNDIRECTED);
```

Figure 10. Protein interaction data loaded to *igraph.*

4.5. Measure the accuracy of protein interaction networks representation

With using igraph, we measure the accuracy of protein interaction networks by obtaining the 'closeness' of the interacting nodes. The closeness measures how easily other nodes can be reached from it. Then we compare the result with the DIP database to ensure that it really represent the interacting nodes.

4.6. Create the adjacency matrix for the graph using igraph

We apply adjacency matrix to simplify the technique of treating protein interaction networks. For the set of interacting proteins, we make the edge weights of all interacting pairs of protein equal to 1 and 0 if there were no interaction. We use igraph to create the adjacency matrix. Figure 11 show the adjacency matrix for one of our data sets.

```
0  0  1  0  0  0  0  0  0  0  0  0  0  0  1
0  0  0  0  1  0  0  0  0  1  0  0  0  0  1
0  1  0  1  0  0  0  1  0  0  1  0  0  0  0
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
1  0  0  1  0  0  1  1  0  0  0  1  0  0  1
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
```

Figure 11. Adjacency matrix for data sets of protein interaction networks which generated by igraph.

4.7. Transforming adjacency matrix to distance matrix.

Our graph-based algorithm require distance matrix as an input data where this data represents the protein interaction matrix. This means we have to transform the adjacency matrix to distance matrix. Three approaches are studied; Dijkstra, Floyd and Jing Liu. Even though three of them were able to represent the protein interaction matrix and proven successfully performing the transformation, we found that only Jing Liu's approach is able to suit with our algorithm. With this transformation, we give the distance matrix to our algorithm as an input for the computational experiment in analyzing the protein interaction networks. Figure 5 illustrate the distance matrix obtained from the transformation.

```
0  633 0  257  390  0  91  661  228  0  412  227
169  383  0  150  488  112  120  267  0  80  572  196
77  351  63  0  134  530  154  105  309  34 29  0
259  555  372  175  338  264  232  249  0  505  289  262
476  196  360  444  402  495  0  353  282  110  324  61
208  292  250  352  154  0  324  638  437  240  421  329
297  314  95  578  435  0  70  567  191  27  346  83
```

Figure 12. The distance matrix obtained from the transformation formula.

5. Summary

In this study, we have performed data pre-processing for computational experiment in analyzing protein interaction networks. We describe the protein-protein interaction as a connectivity graph represent by the interaction matrix a_{ij} We start from downloading yeast protein interaction data from comprehensive protein interaction database, DIP http://dip.doe-mbi.ucla.edu/..We extract and parse the data using C programming language and export them to array-based file. We used *igraph*, a C-library for graph creating and manipulating to obtain the adjacency matrix. We also use igraph to measure the accuracy of interacting protein that represented in our graph. With this approach, we have solved one major problem in representing protein interaction network since it was clearly stated that the representation of protein interaction network having a problem in the aspect of accuracy. Finally, we transform the adjacency matrix to distance matrix, which is the input for our graph-based algorithm.

6. References

[1] G. Pandey, M. Steinbach, R. Gupta, T. Garg, and V. Kumar, "Association analysis-based transformations for protein interaction networks: a function prediction case study. Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007): 540-549, 2007.

[2] N. Przulj, D.A. Wigle, and I. Junsica. Functional topology in a network of protein interactions. *Bioinformatics*, 20(3):340–348, 2004.

[3] C. von Mering, R. Krause, B. Snel, M. Cornell, S.G. Oliver, S. Fields and P. Bork. Comparative assesement of large-scale data sets of protein-protein interactions. *Nature*, 417 (6887):399-403, 2002.

[4] http://dip.doe-mbi.ucla.edu/dip/Stat.cgi Access on November 2007.

[5] G. Csárdi and T. Nepusz, The igraph software package for complex network research. InterJournal Complex Systems, 1695, 2006.

[6] I. Xenarios, D.W. Rice, L. Salwinski, M.K Baron, E.M. Marcotte and D. Eisenberg, "DIP: The Database of Interacting Proteins", *Nucl. Acids Res.* 28:289-91, 2000.

[7] I. Xenarios, L. Salwinski, X.J. Duan, P. Higney, S.M. Kim and D. Eisenberg, "DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions", *Nucl. Acids Res.* 30: 303-305, 2002.