International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 3 | ISSN : 2456-3307

Construction of Protein-Protein Interaction Network Using Community Molecular Detection

J. Monika*, K. Srinivas
Computer Science Department, VR Siddhartha College, Student, Vijayawada, India

ABSTRACT

The number of proteins continues grow. Machine learning is a subfield of computer science that includes the study of systems that can learn from data, rather than follow only explicitly programmed instructions. Some of the most common techniques used for machine learning are Support Vector Machine, Artificial Neural Networks, k-Nearest Neighbor and Decision Tree. Machine learning techniques are widely used techniques in bioinformatics to solve different type of problems. In the year of 2014, the genome project was completed. Some of the proteins have an individual functionality. But there is no accurate information about function for remaining proteins and its network. In general, by using the *In-Vitro* and *In-Vivo* techniques are predict the functionality of proteins and its network. But the experimental investigation is costly and time consuming. To overcome this problem, *In-silico* technique was used such as molecular modeling, etc., but some limitation here is low accuracy. So here to construct Protein-Protein Interaction network for target protein. In this frame work, a novel technique is applied called Community Molecular Detection (CMD). Collect the dataset from "yeastExpData" package called litG. The CMD algorithm operates in two steps, first step is connected components, and second step is community prediction. The first step of CMD, find the connected components by using degree distribution. The second steps, molecular community prediction, takes the output of connected components graph and then find communities.

Keywords : Support Vector Machine, Artificial Neural Networks, K-Nearest Neighbor, Decision Tree, Protein-Protein Interaction Networks, Communities.

I. INTRODUCTION

Proteins represent the most important class of bio molecules in living organisms. They carry out majority of the cellular processes and act as structural constituents, catalysis agents, signaling molecules and molecular machines of every biological system. In all cell functions proteins are virtually involved. Every single protein has specific function within the body. Some of the few proteins are involved in bodily movement, while others are involved in structural support [1]. Proteins differ in functions as well as structures.

Information about the molecular networks that define cellular function, and hence life, is exponentially increasing. One such network is the aggregate collection of all publicly available Protein–Protein Interactions (PPIs) [2].The PPI network is also helpful in drug discovery for particular disease[3].the volume of which in Saccharomyces cerevisiae has dramatically increased in a relatively short time period. For achieving all the proteins network information, proposes a novel algorithm called CMD(Community Molecular Detection).This

algorithm is developed based on MCODE Algorithm[4]

The volume of PPI data has presented the opportunity to analyze systematically the topology of such a large network for functional information using several graph theory-based approaches, and use this to construct models for predicting essentiality[5][6], genetic interactions, function, protein complexes and cellular pathways. In the PPI[2] network, where nodes in the graph represent proteins and the edges that connect them correspond to interactions. To determine graph properties of the network, such as the degree or connection of nodes, the number and complexity of highly connected subgraphs, the shortest path length for indirectly connected nodes, alternative paths in the network and fragile key nodes.

The detection of protein complexes[4] using PPI networks can help in understanding the mechanisms regulating cell life.The problem of detecting protein complexes using PPI networks can be computationally addressed using clustering techniques. Clustering consists of grouping data objects into groups (also called clusters or communities) such that the objects in the same cluster are more similar to each other than the objects in the other clusters (Jain, 1988). As observed in (Fortunato, 2010), a generally accepted definition of 'cluster' does not exist in the context of networks, as it depends on the specific application domain. However, it is widely accepted that a community should have more internal than external connections. For biological networks, the most common assumption is that clusters are groups of highly connected nodes, although recently the notion of community intended as a set of topologically similar links has been successfully used.

1.1. Graph

A loop is an edge that joins a vertex to itself. In a graph, G, and edge is a multiple edge if there is another edge in E(G) which joins the same pair of vertices. A simple graph is a graph with no loops or multiple edges. The most important characteristic of a graph is the degree or connectivity of a vertex. The degree of a vertex is the number of other vertices connected to it.

Path between two nodes is the sequence of edges connecting those nodes. There are several paths for specific two nodes. The minimum number of edges required to reach a node from the other node is the shortest path between two nodes. A path is closed if its first and last vertices are same. Path length is the number of edges in the path. Distance within a network is measured in terms of path [5]. A cycle of length n, denoted Cn in a graph G is a closed path of length n. Two vertices are connected if and only if there exists a path from one vertex to another. A graph G is a connected graph if, for every vertex v, there is a path to every other vertex in V(G). A graph G is a tree if and only if it is a connected graph with no cycles and has exactly one simple path from one vertex to every other vertex.

Protein-Protein Interaction network can be modeled as an undirected, un weighted graph G= (V, E) where V is the set of proteins and E is the set of interactions such that the elements in E is a set of pair of proteins which interact with each other[5]. Graph theory and graph algorithms are well understood field of computers science. Graph mining is the process of extracting subgraphs from graphs to find a useful information regarding the data which the graph is associated. Several graph mining techniques are there to extract subgraphs. Frequent sub graph mining, clustering, classification etc. are some of the well-known techniques used in graph mining. Graph algorithms suits for one application may not suit for another.

The protein-protein Interactions have the power law feature of scale free networks. That is, few nodes are of high degree and others are of less degree. Since most proteins participate in only a few interactions and a few proteins participate in huge number of interactions, the protein-protein interaction network follows the power law.

Another characteristic of protein-protein interaction network is that, it possesses "small world effect". That means, two nodes can be connected through a short path of few edges.

An important characteristic of protein-protein interaction network is disassortativity. That means highly connected nodes rarely directly link to each other.

Among the graph mining techniques, it is learned that graph clustering is very useful in mining group of proteins that performs a specific biological function [5]. There are two types of protein-protein interaction clustering methods.

1. Distance based clustering: will not consider the topological properties of the network.

2. Graph based clustering: based on the topological properties of the network. The graph based clustering methods include:

a. Local neighborhood density search method

b. Flow Simulation method

c. Population based stochastic search method

From the previously published papers it has been learned that analysis of topological properties of protein-protein interaction graph can pave a way to biological inference. Since it is decided to analyze the topological properties of the protein-protein interaction graph to extract the clusters, graph based clustering method has been applied to evaluate the network.

II. BACKGROUND WORK

K.srinivas et.al.[9]proposes a Protein-Protein Interaction detection methods are categorically classified into three types, namely, *in vitro*, *in vivo*, and *in silico* methods. In *in vitro* techniques, TAP tagging was developed to study PPIs under the intrinsic conditions of the cell. The *in vitro* methods in PPI detection are tandem affinity purification, affinity chromatography, protein arrays, protein fragment complementation, phage display, X-ray crystallography, and NMR spectroscopy.In *in vivo* techniques, a procedure is performed on the whole living organism itself. The *in vivo* methods in PPI detection are yeast two-hybrid (Y2H, Y3H) and synthetic lethality. *In silico* techniques are performed on a computer (or) via computer simulation. The *in silico* methods in PPI detection are sequence-based approaches, structure-based approaches, chromosome proximity, gene fusion, *in silico* 2 hybrid, mirror tree, phylogenetic tree, and gene expression-based approaches. While available methods are unable to predict interactions with 100% accuracy, computational methods will scale down the set of potential interactions to a subset of most likely interactions. These interactions will serve as a starting point for further lab experiments. protein interaction data will improve the confidence of protein-protein interactions and the corresponding PPI network when used collectively. Recent developments have also led to the construction of networks having all the Protein-Protein Interactions using computational methods for signal transduction pathways and protein complex identification in specific diseases.

k.srinivas et.al.[10] proposes, To reconstruct a network with strong interactions identified by topological features of a graph. Breast cancer is one of the highly cited chronicle disease in modern world. Based on earlier publications and investigations considered breast cancer based target protein ERBB2 and its interacting protein dataset of homo sapiens category from STRING database to illustrate graph reduction, selection of major nodes and reconstruction of the network with dominating set of nodes. STRING provides dataset based on both physical and functional associations of genes and

proteins. To illustrate pruning activity by measuring betweenness centrality(BC) of a node in a third party tool cytoscape. In calculation of centrality of graph one can measure node based betweenness centrality and edge based betweenness centrality. Node betweenness centrality [9] calculates the number of shortest paths from all nodes to all others that pass through given target node. It is ideal to consider high betweenness proteins in reconstruction of the network to retain maximum benefit. Significant research efforts are making to analyze large scale biological networks. Graph pruning maximizes the benefit of graph analysis with reduced graph. In graph Graph reduction through betweenness centrality metric is also suffering from following pitfalls, it is not suitable to dynamic graph. It is expensive and time consuming in calculation of betweenness value to each node of the large size graph.

Ben Hur A et al. [11] proposes the SVM or kernel machines are widely used in bioinformatics and computational biology for classifying biological data [11] as well as protein-protein interaction prediction [12][13][14]. The support vector machine (SVM) classifier is underpinned by the idea of maximizing the margins. Intuitively, the margin for an object is related to the certainty of its classification. Objects for which the assigned label is correct and highly certain will have large margins and objects with uncertain classification are likely to have small margins [15].

Jansen R et al. [16] proposes an Naïve Bayes algorithm. It is a probabilistic classifier that is based on Bayes theorem and it is a popular algorithm owing to its simplicity (the source of simplicity is the assumption that the independent variables are statistically independent.), computational efficiency and easy to interpret. In spite of the simplicity of this classifier, it turns out that Naïve Bayes works quite well in problems involving normal distributions, which are very common in real-world problems.

Chen X et al [17] proposes a Decision tree is a popular machine learning classifier, which has great applications in bioinformatics and computational biology and has shown to be one of the best classifier for protein-protein interaction prediction. In these trees, internal node test features, each branch correspond to feature value and finally leaves assigns a class label. In the training phase, training dataset is partitioned into the subsets according to the feature values and this process is recursively done on the subsets until splitting no effect on the classification.

E.A.Lan Liang et al. [18] proposes the KNN is focused on the question of how to integrate various data sources to enhance the prediction accuracy. They discussed and evaluated several different integration schemes. Their results strongly indicate that integrating information from various data sources could enhance protein function prediction accuracy [18]. At the level of sequence similarity-based predictions, they observed that it is beneficial to consider all available annotated proteins, regardless how evolutionary distant they are from a query protein. They used the simple and efficient k-Nearest Neighbor algorithm, coupled with simple integration of prediction scores from various data sources.

III. PROPOSED APPROACH

In previous techniques like KNN, uses the Euclidian distance technique. So the Euclidean distance between means of peptide sequence spaces is not suitable for measuring the similarity between the C-terminal β-strands of different organisms. Instead, the similarity measure should also represent how strongly their associated sequence spaces overlap. To achieve this, use the Hellinger distance and then our proposed CMD algorithm.

The degree distribution is performed using Hellinger distance. Find the number of connections to a particular protein is called degree distribution. The length of paths between nodes in a graph can be used to induce a distance between nodes. In many cases, the shortest path will be used, but other alternatives may be appropriate for applications. If the graph has weighted edges, then these can easily be accommodated. Multi-graphs (graphs with multiple

types of edges) can have different distances determined by the different types of edges. Other notions of distance, such as the number of paths that exist between two points [19][20], or the number of edge-cuts required to separate two nodes, can also be used.

The Hellinger distance is used to measure the similarity between the proteins. Afterward, an algorithm is applied to clustering proteins sequences using the Hellinger distance.

$$D^2(P,Q) = \frac{1}{2}\sum_{i=1}^{N}(\sqrt{p_i} - \sqrt{q_i})^2 \qquad (1)$$

Note that P and Q are described as N-tuples (vectors) of probabilities, where P= $P_1,P_2,\ldots, P_N$ and Q = $q_1,q_2,\ldots,q_N$, p_i and q_i are assumed to be non-negative real numbers such that $a_ip_i = 1$ and $a_iq_i = 1$.

Hellinger distance is a metric quantity, which means that it has the properties of non-negativity, the identity, and symmetry, besides, to obey the triangle inequality [20][21][22]. The hellinger distance between two variables can be computed between two variables.

Given a series x_i and y_i of n simultaneous observations for two random variables X and Y. Let $f_X(i)$ denote the number of observations i in X. The probabilities then estimated as:

$$p_i = \frac{f_X(i)}{n} \qquad (2)$$

Let $f_Y(j)$ denote the number of observations of j in Y. The probabilities are estimated as:

$$q_i = \frac{f_Y(j)}{n}, \qquad (3)$$

Implement the proposed system using R-programming. Figure 1 represents architecture of proposed algorithm , divide the frame work into five modules. They are,

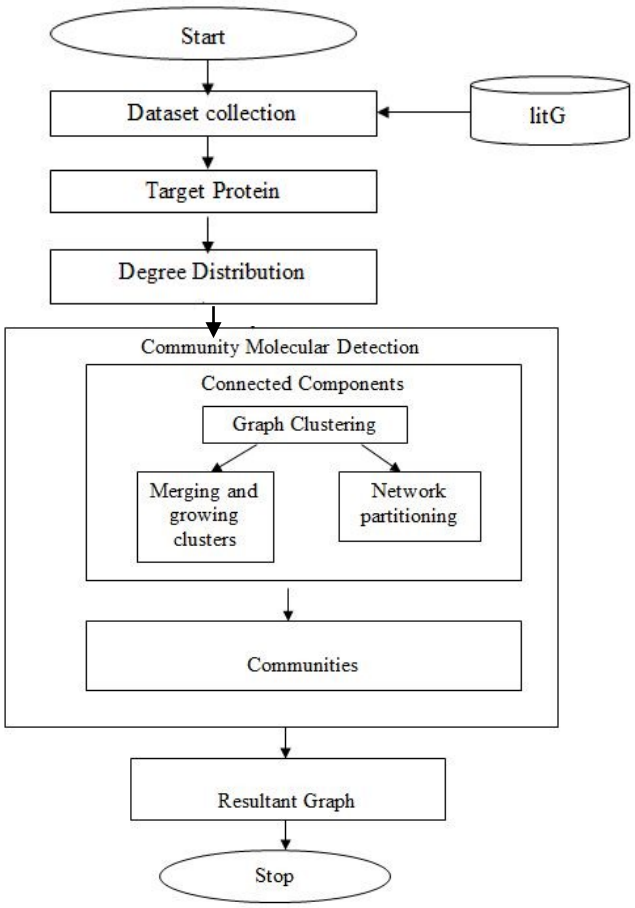


Figure 1: Architecture of proposed Algorithm

A. Dataset Collection

One thing that is useful to know about R is that many R packages come with example data sets, which can be used to familiarize our self with the functions in the particular package. To list the data sets that come with a particular package, you can use the data() function in R. For example, to find the data sets that come with the graph package.

First analyze a curated data set of protein-protein interactions in the yeast Saccharomyces cerevisiae extracted from published papers. This data set comes from with an R package called yeastExpData, which calls the data set as litG.

To read the litG data set into R, First need to load the yeastExpData package, and then we can use the R data() function to read in the litG (Literature and Y2H interaction graphs) data set.The data are stored as instances of the graphNEL class. Each has 2885 nodes, named using yeast standard names. Interactions either represent literature curated interactions, or Y2H interactions.

The graph R package contains many functions for analyzing graph data in R. The nodes () function from the graph package can be used to retrieve the names of the vertices (nodes) in the graph. Note that the order that the proteins are stored in the graph does not have any meaning.

B. Target protein:

Protein-protein interaction data can be described in terms of graphs. In this practical, we will explore a curated data set of protein-protein interactions, by using R packages for analyzing and visualizing graphs.Read the litG data set using the data() function, it is stored as a graph in R. In this graph, the vertices (nodes) are proteins, and edges between vertices indicate that two proteins interact. There are 2885 different vertices in the graph, representing 2885 different proteins. Here I want to take my target protein as YBR009C from the dataset.

C. Degree Distribution:

The degree of a node in a graph is equal to the number of edges containing that node. The degree for all nodes (i.e. proteins) in the PPI network has been computed using Hellinger Distance, and also compute the mean, sorting. For sorted the nodes of the PPI graph by degree, identified nodes in the top 3 and 5%, as well as nodes of degree 1 (since ~25% of nodes of the PPI graph are of degree 1). In these groups of very high and very low degree nodes.

The degree of a vertex (node) in a graph is the number of connections that it has to other vertices in the graph. The degree distribution for a graph is the distribution of degree values for all the vertices in the graph, that is, the number of vertices in the graph that have degrees of 0, 1, 2, 3, etc.

D.Connected Components

In graph theory, a connected component (or just component) of an un directed graph is a sub graph in which any two vertices are connected to each other by paths and which is connected to no additional vertices in the super graph. For example, the graph shown in the figure 2, it has three connected components. A vertex with no incident edges is itself a connected component. A graph that is itself connected has exactly one connected component, consisting of the whole graph.

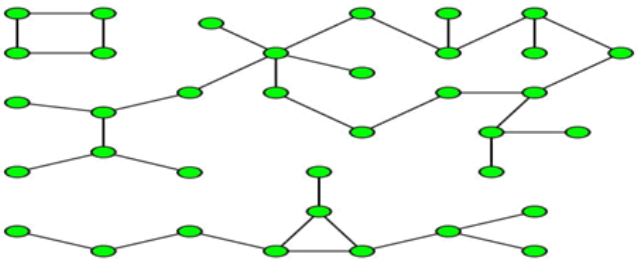


Figure 2.A graph with three connected components.

The connected component is sub graph, in which connect one vertex to another vertices through edges. In the Connected Components, the main task is clustering. Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). Partition the entire network in to a group sub networks.

E.Communities:

In the study of complex networks, a network is said to have community structure if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally. In the particular case of non-overlapping community finding, this implies that the network divides naturally into groups of nodes with dense connections internally and sparser connections between groups. But overlapping communities are also allowed. Figure 3 represents a graph with three communities, The more general definition is based on the principle that pairs of nodes are more likely to be connected if they are both members of the same community(ies), and less likely to be connected if they do not share communities.

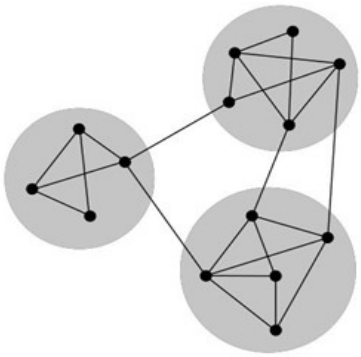


Figure 3. A graph with three communities

Take the output of connected components as input of community section, find out the target protein is comes under which community.

The CMD algorithm operates in two steps, first step is connected components, and second step is community prediction. A network of interacting molecules can be intuitively modeled as a graph, where vertices are molecules and edges are molecular interactions.

If temporal pathway or cell signaling information is known, it is possible to create a directed graph with arcs representing direction of chemical action or direction of information flow; otherwise an undirected graph is used. Using this graph representation of a biological system allows graph theoretic methods to be applied to aid in analysis and solve biological problems.

Density of a graph, G = (V,E), with number of vertices, |V|, and number of edges, |E|, is defined here as |E|; divided by the theoretical maximum number of edges possible for the graph, $|E|_{max}$. For a graph with loops (an edge connecting back to its originating vertex), $|E|_{max} = |V| (|V|+1)/2$ and for a graph with no loops, $|E|_{max} = |V| (|V|-1)/2$. So, density of G, $DG = |E|/|E|_{max}$ and is thus a real number ranging from 0.0 to 1.0.

The first step of CMD, connected components, It is straightforward to compute the connected components of a graph in linear time (in terms of the numbers of the vertices and edges of the graph) using either breadth-first searchor depth-first search. In either case, a search that begins at some particular vertex *v* will find the entire connected component containing *v* (and no more) before returning. To find all the connected components of a graph, loop through its vertices, starting a new breadth first or depth first search whenever the loop reaches a vertex that has not already been included in a previously found connected component.

Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking. The time and space analysis of DFS differs according to its application area. In theoretical computer science, DFS is typically used to traverse an entire graph, and takes time $\Theta(|V| + |E|)$, linear in the size of the graph. In these applications it also uses space $O(|V|)$ in the worst case to store the stack of vertices on the current search path as well as the set of already-visited vertices. Thus, in this setting, the time and space bounds are the same as for breadth-first search and the choice of which of these two algorithms to use depends less on their complexity and more on the different properties of the vertex orderings the two algorithms produce.

Algorithms for finding clusters, or locally dense regions, of a graph are an ongoing research topic in computer science and are often based on network flow/minimum cut theory or more recently, spectral clustering. With the large availability of protein interaction networks and microarray data supported, to identify the linear paths that have biological significance in search of a potential pathway is a challenge issue so we use all pairs shortest path.

The second step, molecular community prediction, takes output of connected component graph as a input, seeds a complex with the highest degree vertex and recursively moves outward from the target vertex. If a vertex is included, its neighbors are recursively checked in the same manner to see if they are part of the complex. A vertex is not checked

more than once, since complexes cannot overlap in this stage of the algorithm. This process stops once no more vertices can be added to the complex based on the given distance metric and is repeated for the next highest unseen weighted degree vertex in the network. In this way, the densest regions of the network are identified. The distance metric parameter defines the density of the resulting complex. A distance that is closer to the weight of the target vertex identifies a smaller, denser network region around the seed vertex. Each and every community has some group of proteins.

IV. PERFORMANCE EVALUATION

In section 2 and section 3 defines the basic concepts and methodology procedure. First analyze a curated data set of protein-protein interactions in the yeast Saccharomyces cerevisiae extracted from published papers. This data set comes from with an R package called yeastExpData, which calls the data set as litG.

Collect the dataset from yeastExpData, and retrieve data by using data() function. Figure 4. shows that input dataset which contains 2885 proteins.

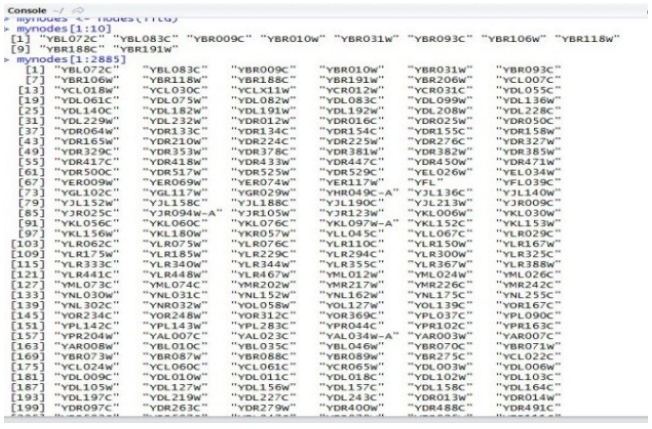


Figure 4. Input Dataset

The dataset contains 2885 yeast proteins, from these select one protein as target protein. here YBR009C as a target protein. Next find out the degree of each and every protein in dataset. The degree of a node in a graph is equal to the number of edges containing that node. The degree for all nodes (i.e. proteins) in the PPI network has been computed using Hellinger Distance. The degree of a vertex (node) in a graph is the number of connections or interactions that it has to other vertices in the graph. The degree distribution for a graph is the distribution of degree values for all the vertices in the graph, that is, the number of vertices in the graph that have degrees of 0, 1, 2, 3, etc.

To calculate the degrees of all the vertices in a graph by using the degree() function in the R graph package. The degree() function returns a vector containing the degrees of each of the vertices in the graph. Remember that there is a degree() function in both the graph and igraph packages, so if you have loaded both packages, you will need to specify that you want to use the degree() function in the graph package, by writing graph::degree().Figure5 shows that degree distribution of each and every proteins.

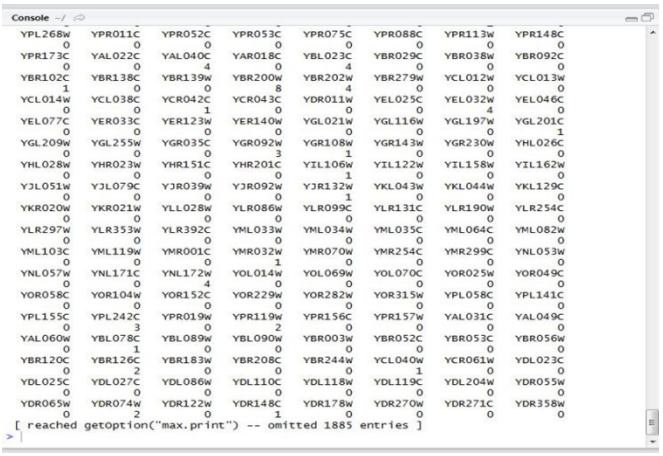


Figure 5. Degree distribution

As to see the above results that the yeast protein YPL268W does not interact with any other protein, while the yeast protein YBR102C interacts with one other yeast proteins. We can sort the vector mydegrees in order of the number of degrees, by using the sort() function. Figure 6. shows that sorting order of each and every protein.

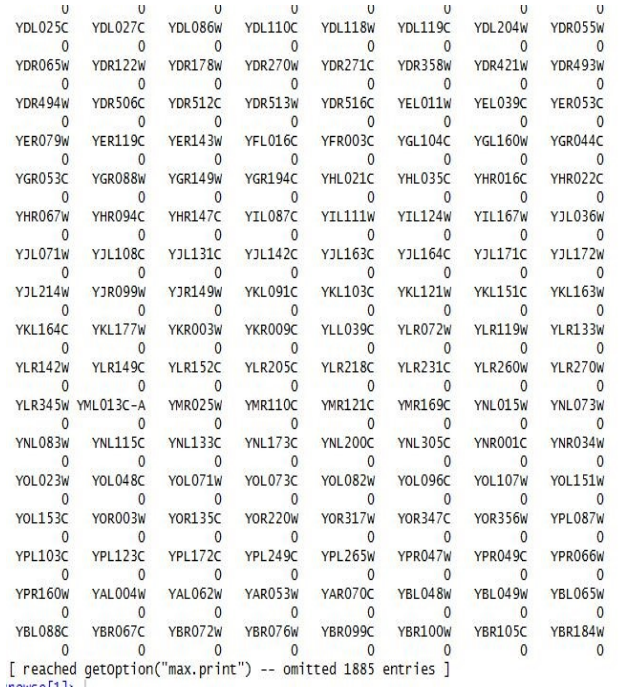


Figure 6. Sorting

V. EXPERIMENTAL RESULTS

This section talks about the outcomes acquired for construction of PPI network using CMD algorithm. Finally to perform two tasks in this research, such as connected components and Communities detection.

After find out the degrees of each and every proteins, we have to visualize the degrees information on graphs by using histogram. A histogram is an accurate graphical representation of data. It is a kind of bar graph. To construct a histogram, the first step is to "bin" the range of values that is, divide the entire range of values into a series of intervals and then count how many values fall into each interval. The bins are usually specified as consecutive, non-overlapping intervals of a variable. The bins (intervals) must be adjacent, and are often (but are not required to be) of equal size. The figure 7 shows that Histogram, the x-axis represents mydegrees and y-axis represents frequencies.

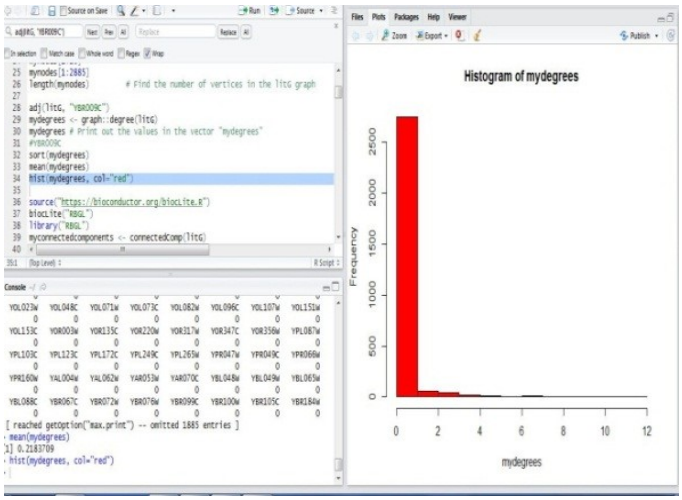


Figure 7. Histogram results for above sorting data set.

For the first process, Analyzing a very large graph, it may contain several sub graphs, where the vertices within each subgraph are connected to each other by edges, but there are no edges connecting the vertices in different sub graphs. In this case, the sub graphs are known as connected components (also called maximally connected sub graphs). Figure 8 shows that connected components.

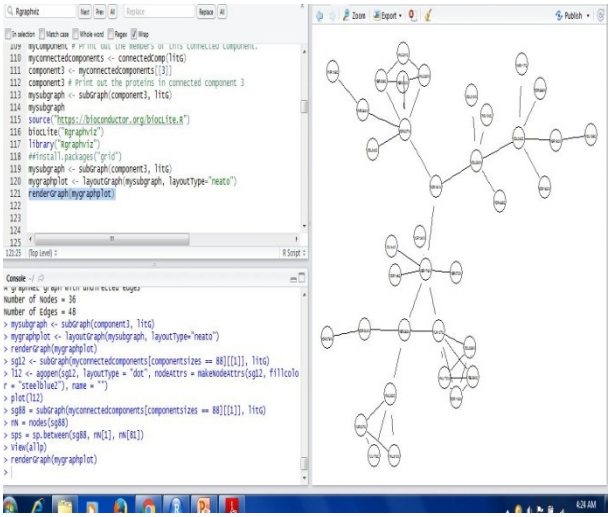


Figure 8. Graphical representation of connected components

There are 2642 different connected components in the litG graph. These are 2642 subgraphs of the graph, where there are edges between the vertices within a subgraph, but no edges between the 2642 subgraphs.It is interesting to know the largest connected component in a graph.

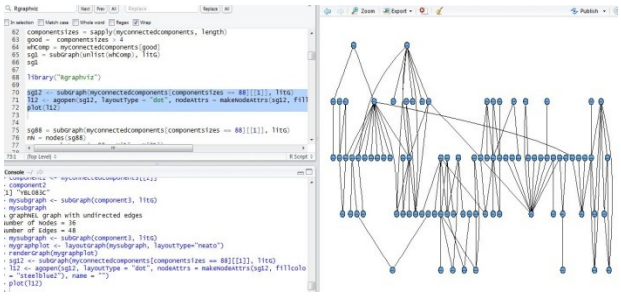
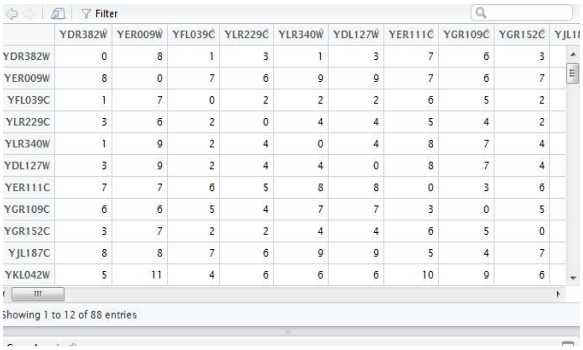


Figure 9. Largest connected component

That is, Figure 9. shows that largest connected component has size 88. There are 2587 singletons (connected components of size 1). plot these largest components using the Rgraphviz package.

To know the diameter of the graph it is defined as the longest shortest path between any two nodes. To compute this function, use johnson.all.pairs.sp. Figure 10 shows that shortest path for each and every protein.



	YDR382W	YER009W	YFL039C	YLR229C	YLR340W	YDL127W	YER111C	YGR109C	YGR152C	YJL1
YDR382W	0	8	1	3	1	3	7	6	3	
YER009W	8	0	7	6	9	9	7	6	7	
YFL039C	1	7	0	2	2	2	6	5	2	
YLR229C	3	6	2	0	4	4	5	4	2	
YLR340W	1	9	2	4	0	4	8	7	4	
YDL127W	3	9	2	4	4	0	8	7	4	
YER111C	7	7	6	5	8	8	0	3	6	
YGR109C	6	6	5	4	7	7	3	0	5	
YGR152C	3	7	2	2	4	4	6	5	0	
YJL187C	8	8	7	6	9	9	5	4	7	
YKL042W	5	11	4	6	6	6	10	9	6	

Showing 1 to 12 of 88 entries

Figure 10. All pairs shortest path

For the second process, By detecting communities within a protein-protein interaction graph, to detect putative protein complexes, that is, groups of associated proteins that are probably fairly stable over time. In other words, protein complexes can be detected by looking for groups of proteins among which there are many interactions, and where the members of the complex have few interactions with other proteins that do not belong to the complex.

There are lots of different methods available for detecting communities in a graph, and each method will give slightly different results. That is, the particular method used for detecting communities will decide how you split a connected component into one or more communities. The function findcommunities() below identifies communities within a graph (or subgraph of a graph). It requires a second function, findcommunities2 ().

In PPI, one of the result can get six different communities in the sub graph corresponding to the third connected component of the litG graph.

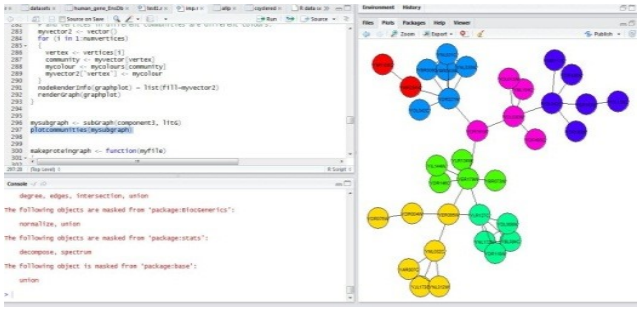


Figure 11. Resultant graph with communities

Figure 11 shows that, the six communities in the third connected component of the litG graph are colored with six different colours.

VI.CONCLUSION

We introduce a novel technique known as Community Molecular Detection algorithm. This algorithm effectively discovers largely connected regions of a molecular connections network, based solely on connection details. Given that this approach to examining proteins connections systems performs well using minimal qualitative details implies that considerable amounts of available details are hidden in huge proteins connections systems. More accurate details exploration methods and systems models could be constructed to understand and predict communications, areas and pathways by considering more existing bio-logical details. In the CMD algorithm, we get a more details about proteins communications and molecular routes known as areas. So by using these details we have to find the particular drug objectives for disease.

VII. REFERENCES

1. David L.González-Álvarez, Miguel A.Vega-Rodriguez,Alvaro Rubio-Largo, "Finding Patterns in Protein Sequences by Using a Hybrid Multiobjective Teaching Learning Based Optimization Algorithm",IEEE,vol .12,issue 3,2015
2. Drees BL, Sundin B, Brazeau E, Caviston JP, Chen GC and Guo W, "A protein interaction map for cell polarity development", 154:549-571, J CellBiol 2001.
3. V Srinivasa Rao, K Srinivas, GN Sunand Kumar & GN Sujin "Protein interaction network for Alzheimer's disease using computational approach" BIOINFORMATION Volume 9(19) ISSN 0973-2063 page number:968-970
4. Gary D Bader and Christopher WV Hogue "An automated method for finding molecular complexes in large protein interaction networks" BMC Bioinformatics 2003,Page Numbers:1-27
5. K Srinivas ,R Kiran Kumar, M Mary Sujatha "A Study on Public Repositories of Human Protein Protein Interaction Data", IJIACS, ISSN2347-8616 ,vol6-issue6, June2017.
6. M Mary Sujatha, K. Srinivas, R. Kiran Kumar "A Review on Computational Methods Used in Construction of Protein Protein Interaction Network" International Journal of Engineering and Management Research Volume-6, Issue-6, Page Number: 71-77 November-December 2016
7. M Wu, X.L. Li, C.K. Kwoh and S.K. Ng, "A Core-Attachment based Method to Detect Protein Complexes in PPI Networks," BMC Bioinformatics, vol. 10, pp. 169, 2009.
8. J Susymary, R. Lawrance ,"Graph Theory Analysis of Protein-Protein Interaction Network and Clustering proteins linked with Zika Virus" Vol. 5, Special Issue 1, pp.100-108, 2017.
9. vsrinivasa Rao,k.srinivas "Protein-Protein Interaction Detection: Methods and Analysis", Hindawi Publishing Corporation International Journal of Proteomics Volume 2014, Article ID 147648, 12 pages
10. M Mary Sujatha, K Srinivas "Pruning Protein Protein Interaction Network in Breast Cancer Data Analysis", International Journal of Computer Science and Information Security (IJCSIS),Vol. 15, No. 7, July 2017.
11. Ben Hur A, Ong CS, Sonnenburg S, Schölkopf B, Rätsch G. Support Vector Machines and Kernels for Computational Biology. PLoS.comput. biol. 2008;4(10):e1000173.
12. Guo Y, Yu L, Wen Z, Li M. Using support vector machine combined with auto covariance to predict protein protein interactions from protein sequences. Nucleic Acids Res. 2008;36(9):3025–3030.
13. Lo S, Cai C, Chen Y, Chung M. Effect of training datasets on support vector machine prediction of protein-protein interactions. Proteomics.2005;5(4):876 – 884.
14. Dohkan S, Koike A, Takagi T. Improving the Performance of an SVM-Based Method for Predicting Protein-Protein Interactions. In. Silico Biol. 2006;6:515–529.
15. Rashid M, Ramasamy S, PS Raghava G. A simple approach for predicting protein-protein interactions.Curr.Pro.Pept. Sci. 2010;11(7):589–6000.
16. Jansen R, Yu H, Greenbaum D, Kluger Y, Krogan N, Chung S, Emili A, Snyder M, Greenblatt J, Gerstein M. A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data. Science. 2003;302(5644):449 – 453.
17. Chen X, Wang M, Zhang H. The use of classification trees for bioinformatics.Wiley Interdisciplinary Reviews. J of Data Mini and Know Disc. 2011;1(1):55–63.

18. EA.Lan Liang, "MS-kNN: protein function prediction by integrating multiple data sources," BMC bioinformatics, 2014.
19. Leicht EA, Holme P, Newman MEJ. Vertex similarity in networks. Physical ReviewE. 2006;73:026120.doi:10.1103/PhysRevE.73.026120. [doi:10.1103/PhysRevE.73.026120].
20. Donoho, D. & Liu, R. (1988). The 'automatic' robustness of minimum distance functionals, Annals of Stat., Vol. 16,(1988), pp. 552-586, ISSN 0090-5364.
21. Giet, L. & Lubrano, M. (2008). A minimum Hellinger distance estimator for stochastic differential equations: an application to statistical inference for continuous time interest rate models, Comput. Stat. & Data Anal., Vol. 52,No. 6, (Feb. 2008), pp. 2945-2965, ISSN: 0167-9473.
22. Azim, G. A, Aboubekeur Hamdi-Cherif, Mohamed Ben Othman and Z.A. Abo-Eleneen" Protein Progressive MSA Using 2-Opt Method" Systems and Computational Biology-Bioinformatics and Computational Modeling 2011 InTech September 2011.