We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



14

Finding Protein Complexes via Fuzzy Learning Vector Quantization Algorithm

Hamid Ravaee[1,2,*], Ali Masoudi-Nejad[1] and Ali Moeini[2]
[1]*Laboratory of Systems Biology and Bioinformatics (LBB), Institute of Biochemistry and Biophysics, University of Tehran, Tehran,*
[2]*Department of Algorithm and Computations, College of Engineering, University of Tehran, Tehran,*
Iran

1. Introduction

Protein–protein interactions (PPI) make up fundamentals of biological processes inside a cell. PPI has most important roles in cells such as post-translational regulation of protein activity, which is occurred by transient protein-protein interactions and participating in enzymatic complexes ensures substrate channelling which drastically increases fluxes through metabolic pathway (Lin et al., 2006). Metabolic pathways, for instance, consist of several proteins, called enzymes, organize a series of chemical reactions with the intent of altering a variety of chemical substance into the other forms, namely products. Proteins interactions happen in signalling pathways where a set of proteins, by an ordered sequence of reactions, try to convert a type of chemical signal to other form, enabling a cell to obtain environmental information quickly. Proteins interactions can be found in any sort of biological processes within cells. Indeed, existence of these interactions makes a cell function, to grow and more importantly survive (Bader & Hogue, a2003).

The objective in PPI network analysis is the discovering dense highly-connected subgraphs that represent functional modules and protein complexes. For understanding the cell function, it is essential first to find all functional modules in protein interaction networks (Bader & Hogue, b2003). Protein complexes are a group of proteins which have more interactions with each other at the same time and place (Chua et al. , 2008). On the other hand, the functional module consists of proteins that participate in a particular cellular process while interacting with each other at different time and place (Mirny & Spirin, 2003) . In order to simplify the terms, we used protein complex and functional modules as same. Since each protein could be involved in several protein complexes, the partitioning of PPI network to some disjoint groups of subgraphs could not explain the true nature of protein complexes occuring in PPI network. Hence, the finding of vertices group with overlapped boundary can be more useful in analyzing PPI network.

* Corresponding Author

In recently years, advances in the high-throughput PPI detection have produced a high volume of PPI datasets freely available to researchers. Therefore many methods and approaches have emerged to analyze experimental PPI data in various organisms. The experimental approaches for discovering protein complexes are more time consuming and expensive. Instead, computational methods which use PPI data are faster and cheaper (Ito et al., 2001).

The most common method of modelling PPI network is using graph theory, which in such a graph G=(V,E) where the nodes correspond to proteins and the edges correspond to interactions. Since the number of proteins and interactions between them in some organism such as yeast or human is remarkably high, the graph modelling PPI is called a complex graph. Partitioning of a complex graph to some disjoint subgraphs is called the graph clustering.

Clustering is the process of grouping data into sets (clusters) which shows more similarity between the objects in the same clusters than they are in different clusters (Schaeffer, 2007). Clustering analysis seeks a set of clusters based on similarity between pairs of elements. Graph clustering is the practice of distribution the vertices of the graph into the clusters taking into consideration the edge connectivity in the graph in such a way that many edges exist within each cluster and relatively few between the clusters. The result of this clustering can define the PPI network's structure and imply functions of proteins in the cluster which were previously uncharacterized (Lin et al., 2006).

Each complex graph modelling a system such as biological systems or social networks has specific properties and characteristics. The properties of graph could be fall into broad categories as the local properties and global properties (Przulj, 2005). The scale-free for distribution of degree and small world properties could be more affective on the result of graph clustering. A scale-free network has a vertex connectivity distribution that follows a power law, with relatively few highly connected vertices and many vertices having a low degree. Most biological networks such as PPI networks have the scale-free property (Pizzuti & Rombo, 2007). In this paper, we convert the normal scale-free PPI network to a non-scale free network by using line graph transformation. In the graph theory, line graph is produced by substituting edges and nodes in the graph. Each interaction is condensed into a node that includes the two interacting proteins. These nodes are then linked by shared protein content.

Important of results of the clustering in PPI network is illustration of structure of the PPI network which can be used to predict the functionality of uncharacterized protein based on other known proteins functions in the same cluster's elements. These clusters correspond to meaningful biological units such as protein complexes and functional modules.

Many clustering approaches (Gao, 2009; Bader & Hogue, b2003; Adamcsek, 2006; Wu et al., 2008 ;Vlasblom, 2009) could not place elements in multiple clusters, which can be unrealistic for biological systems, where proteins may participate in multiple cellular processes and pathways. Since each protein could participate in more than one protein complexes, in the clustering PPI graph, each protein probably have membership to more than one cluster. So in this paper, we present a clustering method that allows to having overlapping founded clusters. Disjoint clusters and overlapping clusters are illustrated in figure 1.

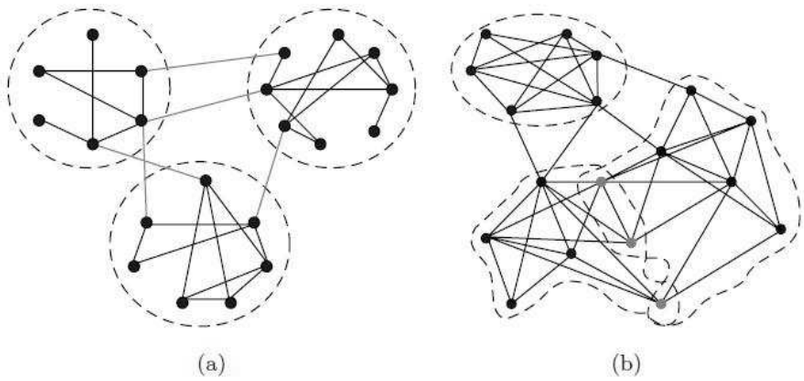


Fig. 1. Illustration of the concept of modules. (a) Disjoint modules; (b) Overlapping modules.

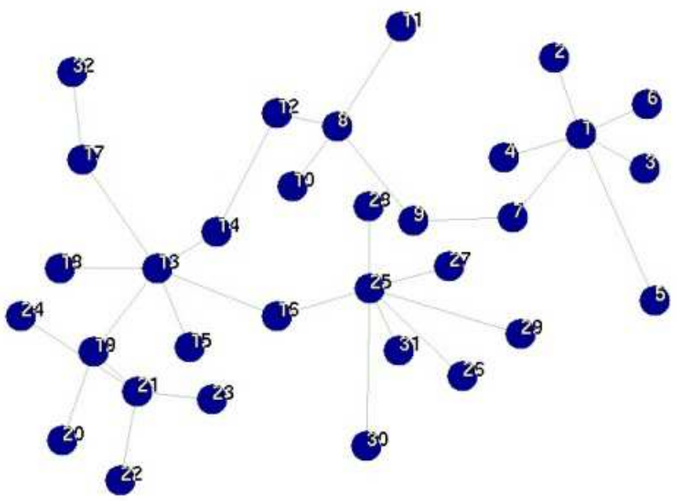
K-means (c-means) clustering (Hartigan, 1975) is applied on unlabeled data by partitioning them on predefined number of groups (k) based on the specifying the centers of groups. After each iteration in the k-means algorithm, the distances between each center of group and other data points are calculated and the center points are updated. Learning Vector Quantization uses k-means idea by defining some codebook vectors each of which represents a cluster for n-dimensional input data. The fuzzy clustering based on fuzzy set theory (Zadeh, 1965) is used to deal with indistinct boundaries between clusters. The most widely used fuzzy clustering method is the fuzzy c-means (FCM) algorithm (Bezdek, 1973) which is generalized from hard c-means algorithm. In this paper, extended FLVQ (Bezdek, 1995) as an intelligent computational method has been used for clustering PPIs. The results of this algorithm can be verified by biological and non-biological criteria and we showed that FLVQ technique is more effective and accurate for finding protein complexes in PPI network.

2. Primary definitions

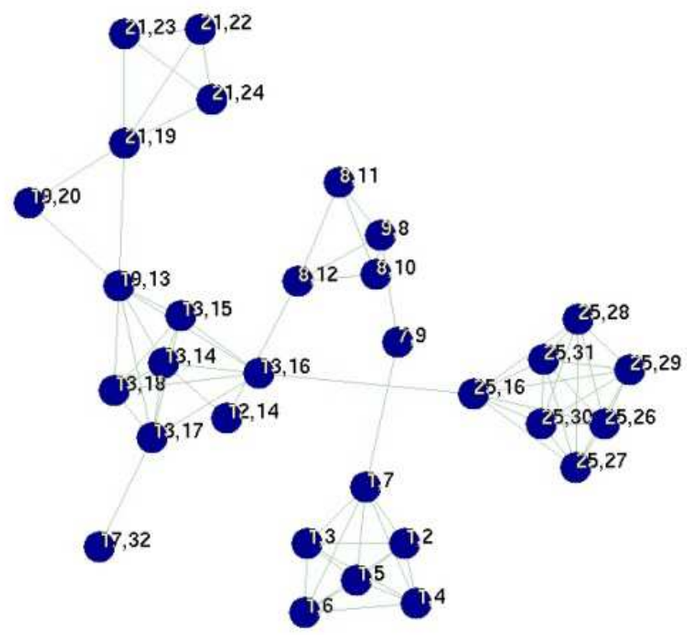
The problem of clustering of PPIs starts with a mathematical representation of PPI networks. A conventional way for representing PPI network is using graph theory concepts. PPI network could be illustrated by a graph G=(V,E) with a set of vertices V and a set of edges E in which each vertex is corresponded by a protein in PPI network and each edge connects to two vertices whose corresponding proteins have physical interaction with each other.

Clusters in the graph could be interpreted as dense subgraphs the number of edges within each subgraph is the maximum number and the number of edges between clusters is the minimum one. Therefore, the PPI clustering is an optimization problem and like other optimization problems, there is a need to an objective function to get optimum point.

PPI networks have scale-free property and finding the dense subgraphs is most difficult task in these networks. So using line graph we eliminate the scale-free property. In each node in the line graph is an edge in original network and every two nodes with common proteins are connected to each other. Figure 2 shows a scale free network and the generated line graph based on original graph.



(a)



(b)

Fig. 2. **a.** Orginal scale-free graph **b.** converted graph by line graph.

2.1 Learning Vector Quantization

Learning Vector Quantization (LVQ) is placed in the competitive learning category and it is closely related to Self-Organizing Map (SOM) (Kohonen, 1990). SOM is a well-developed neural network technique for data clustering and visualization. It can be used for projecting a large data set of a high dimension into a low dimension (usually one or two dimensions) while retaining the initial pattern of data samples. Indeed, SOM has two main principles:

vector quantization and vector projection. Vector quantization makes up a delegate set of vectors called output vectors (codebook vectors) from the input vectors. Let's denote the set of output vectors (codebook vectors) as $Y=\{y_1,y_2,..,y_c\}$ with the same dimension as input vectors. In general, vector quantization reduces the number of vectors, and this can be considered as a clustering process. The maximum number of clusters in a network is defined by user specified value, c. After learning process, it may be possible for some codebook vectors to correspond to empty clusters.

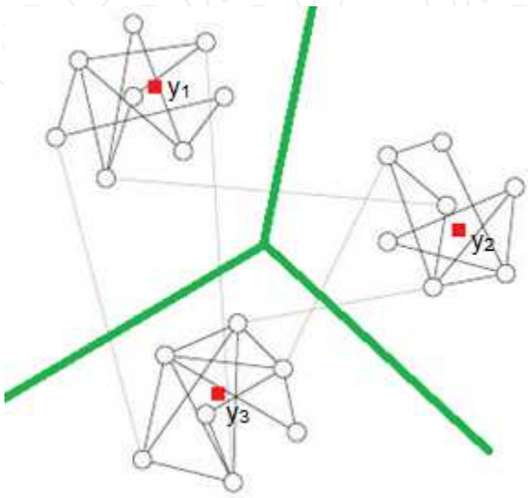


Fig. 3. The red points (y_1,y_2,y_3) corresponded to output vectors indicating a dense subgraph in the sample network.

The LVQ algorithm represents a set of input vectors $x_i \in X \subset \Re^n$ by a set of c prototypes $Y = \{y_1, y_2, .., y_c\} \subset \Re^n$ .. The LVQ is associated with a competitive network which consists of an input layer and an output layer. Each node in the input layer is connected directly to the cells, or units, in the output layer. A weight vector, also referred to as prototype, is assigned to each cell in the output layer (Ravuri & Karayiannis, 1995). The codebook vector having minimum distance with input vector x_i is called winner vector, k, and is defined as:

$$k = \arg \min_l \|y_l - x_i\| \tag{1}$$

Update equation of LVQ algorithm is:

$$y_j(t+1) = y_j(t) + \alpha_t h_{ij,k} \|x_i - y_j(t)\| \tag{2}$$

Here a_t is the scalar-valued learning rate, $0<a_t<1$, and decreases monotonically with time t. The neighborhood function $h_{ij,k}$ denotes the interaction between codebook vector i and j and winner vector k. The simple definition of $h_{ij,k}$ is:

$$h_{ij,k} = \begin{cases} 1 \; if \; j = k \\ 0 \; if \; \; j \neq k \end{cases} \tag{3}$$

In the LVQ algorithm, neighborhood radius is one and only the winner vector could be updated.

2.2 Fuzzy Learning Vector Quantization

While most typical clustering algorithms assigns each data point to exactly one cluster, fuzzy clustering allows for the extent of membership, to which a data point belongs to different clusters. The FLVQ may be seen as a learning fuzzy c-means using a fuzzification index m. Karayiannis et al (Ravuri & Karayiannis, 1995) presented a broad family of FLVQ algorithms, which were initially introduced on the basis of perceptive arguments. This derivation was based on the minimization of the average generalized distance between the input vectors and the prototype vectors. The fuzzy partitioning algorithm, FCM is run into by minimization problem that is solved by reformation of FCM algorithm to FLVQ algorithm (Bezdek, 1995).

The updated equation for the FLVQ involves the membership functions which are used to determine the strength adjacency between each prototype and input vectors.

$$\alpha_{ij,t} = (u_{ij,t})^{m_t} \tag{4}$$

$$u_{ij} = \left[\sum_{l=1}^{c} \left(\frac{D_{ij}}{D_{lj}}\right)^{\frac{2}{m-1}} \right]^{-1} \tag{5}$$

$$D_{ij} = \left\| x_i - y_j \right\| \tag{6}$$

Where $m = m_t = m_0 - \Delta mt$ and $\Delta m = (m_0 - m_f) / MaxItr$ and D_{ij} is the distance and m_0 is some constant value greater than the final value (m_f) of the fuzzification parameter m. *MaxItr* is the constant parameter for limitation of iterations.

3. The FLVQ algorithm

The calculation of distances between network vertices and prototype vectors in the FLVQ is critically challenging. In the following algorithm, we used a new definition of vertices based on n-dimensional vectors and; we representing new scalar distance between input vectors and codebooks (output) vectors. Each vertex in PPI graph is modeled by a vector called input vector. Given *G=(V,E)* represents a PPI network including $|V|$ vertices and $|E|$ edges. An input vector is defined as :

$$x_m = \{x_{m1}, x_{m2}, .., x_{mn}\} \tag{7}$$

$$x_{ij} = \begin{cases} 0 + \varepsilon & if \quad i \neq j \quad and \quad e_{ij} = 0 \\ 1 - \varepsilon & if \quad e_{ij} = 1 \\ 1 - \varepsilon & if \quad i = j \end{cases}$$

Where $n=|V|$, e_{ij} is element *(i,j)* in adjacency matrix corresponding the graph G and ε is a real small value between *(0,1)*.

This definition makes possible to use scalar distance measure such as the dot product is possible. There are some distance criteria in vector space to measure similarity (distance) between two vectors. Correlation is a simple way for measuring distance between two vectors in the same dimension. If x_i and x_j are two vectors with the dimension of n, the equation (8) is the inner product of two vectors:

$$S_{ij} = X_i.X_j = \sum_{k=1}^{n} x_{ik}x_{jk} \tag{8}$$

$$D_{ij} = S_{ij}^{-1} \tag{9}$$

Where D_{ij} is the distance and S_{ij} is the inner product between X_i and X_j.

The FLVQ algorithm performs clustering of the input graph by training process. Training process consists of some iterations. The number of iteration depends on convergence criteria and can be limited by a user specified constant. Each iteration consists some epochs. The number of epochs is equal by c (number of prototype vectors and the maximum number of clusters). In each epoch, an input vector x_i is selected randomly. A selected input vector is not being selected in a same epoch again. The selected input vector x_i is compared with all the prototype vectors with a similarity measure *(ex. dot product)* and the prototype vector y_j with most similarity with x_i known as winner vector.

The implementation of the FLVQ algorithm is described as follows:

- **Step 1. Initialization**
  Initialize the c codebook's vectors $y=\{y_1,y_2,..,y_c\}$ *by randomly assigning each element of codebook vectors by a real number between* $(\varepsilon,1-\varepsilon)$. *Set iteration counter* $t=1$. *Give* $0\leq\varepsilon<1$. t_{max} *is the iteration limit.*
- **Step 2. Learning**
  Repeat until stopping criterion is satisfied:
- **Step 2.1** While there is a unselected input vector
  - *Randomly pick an input* x_i
  - *Compute winner vector based on distance measure of xi and every codebook vectors* $y_j: j=1..k$
  - *update winner vector* y_j *based on input vector* x_i *and learning ratio a*
- **Step 2.2** update learning ratio α

4. Data set

The PPI network is derived from the yeast subset in the Database of Interacting Proteins (DIP) (Xenarios et al., 2002). The dataset of yeast is composed of 4963 proteins and 17570 interactions. Most of these interactions have been derived by yeast two-hybrid screen. For evaluation of finding clusters, we use protein complex data from the MIPS database (Mewes et al., 2004). In the currated complex dataset, there are 404 protein complexes. The protein complex having most proteins is "cytoplasmic ribosomal large subunit" with 88 proteins and there are 169 protein complexes with just two proteins.

5. Experimental result

The FLVQ algorithm is applied on the PPI network of the *Saccharomyces Cerevisiae* (yeast) dataset downloaded from the DIP (Guldener, 2005). After using FLVQ on DIP protein-protein interaction, over than 300 clusters obtained frequency of each based on the number of vertices in is shown in figure (4). As the figure (4) shows most obtained clusters approximately include 9 and 12 vertices. In addition, the number of clusters with size of over 20 are also considerable. This means that the FLVQ algorithm could find larger dense subgraphs in the PPI network. When the cluster size became larger, few graph clustering methods could find these clusters with proper efficiency.

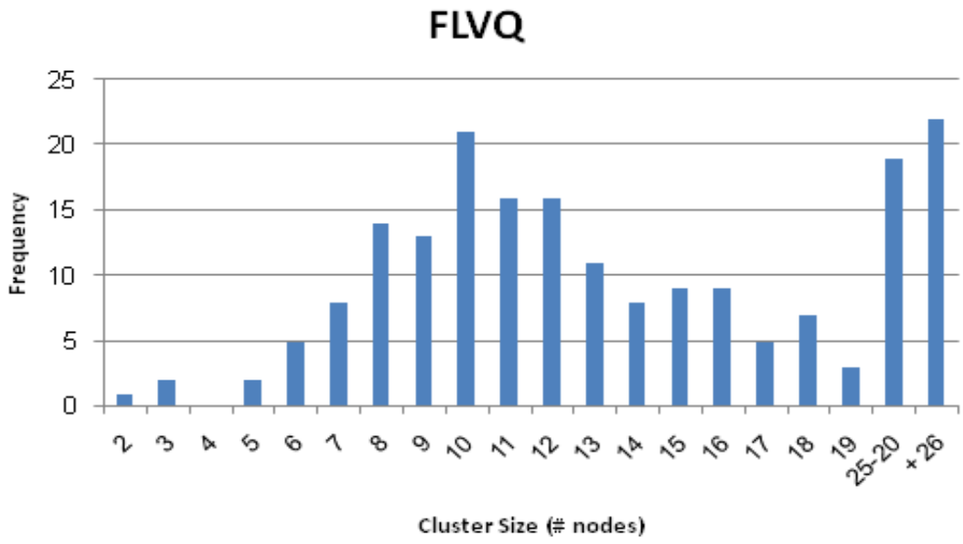


Fig. 4. Number of obtained clusters by FLVQ algorithm based on the cluster size.

The results of the FLVQ algorithm are evaluated by the clustering score used by (Bader & Hogue, a2003; Newman, M. & Girvan. M., 2004). The clustering score for each cluster is defined by the product of size and density of the cluster. The density of cluster is the ratio between number of edges in cluster $|E|$ and maximum number of possible edges in it $|E_{max}|$. The following equation (10) shows clustering score definition.

$$\sigma(\Gamma)=\delta(\Gamma).|V| \tag{10}$$

Where Γ is a cluster in the clustering result and $\delta(\Gamma)$ is the density of given subgraph Γ and is declared by equation (11) and $/V/$ shows the number of vertices in Γ subgraph.

$$\delta(\Gamma)=2|E|/(|V|(|V|-1)) \tag{11}$$

Where E is the set of edges that connects the existing vertices in V in given subgraph of Γ. The clustering score for each clusters is shown in figure (5). The cluster score for bigger

clusters is more elevated than smaller clusters proving that FLVQ is rather successful to find subgraphs with more higher number of vertices and with most density. Highest clustering score shows that the obtained clusters are more compact and larger.

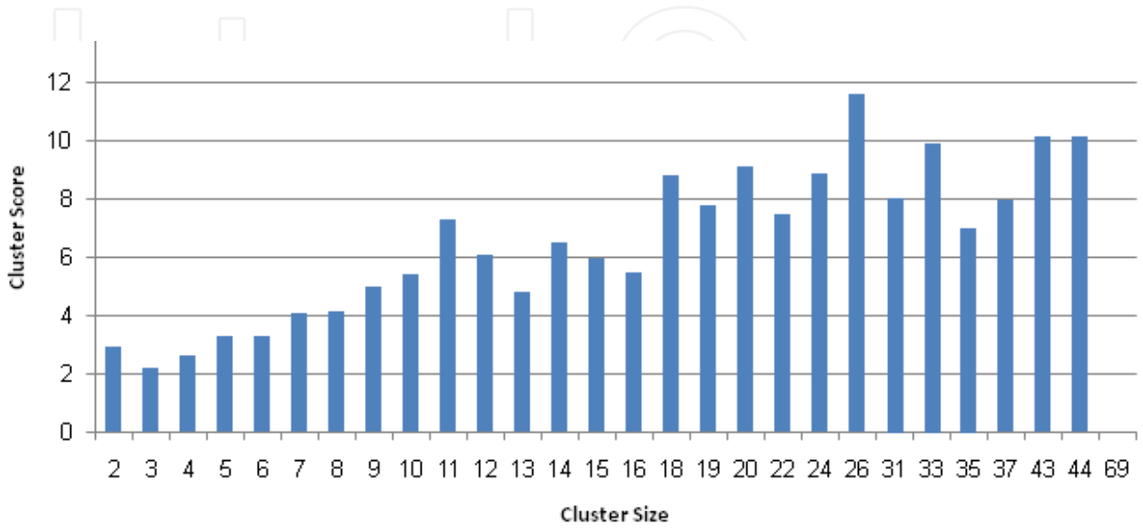


Fig. 5. Amount of clustering score for each obtained cluster in FLVQ algorithm.

The clustering results can be validated by ground truth with Precision and Recall. Assume a module (cluster) X is mapped to a functional module F_i. Recall, also termed the true positive rate or sensitivity, is the proportion of proteins common in both X and F_i to the size of F_i. Precision, which is also termed the positive predictive value, is the proportion of proteins common in both X and F_i to the size of X.

$$precision = \frac{|X \cap F_i|}{|X|} \tag{12}$$

$$recall = \frac{|X \cap F_i|}{|F_i|} \tag{13}$$

The accuracy of clusters is assessed by *f*-measure. The *f*-measure is defined as the harmonic mean of recall and precision:

$$f - measure = \frac{2(precision.recall)}{precision + recall} \tag{14}$$

Figure (6) shows the average of f-measure based of protein complex size for the FLVQ algorithm. In figure (6), the f-measure of each obtained cluster is measured based on experimental protein complexes MIPS. The value of f-measure could be between 0 and 1.

The highest f-measure value indicates the most conformity between experimental protein complex and obtained complex by the algorithm.

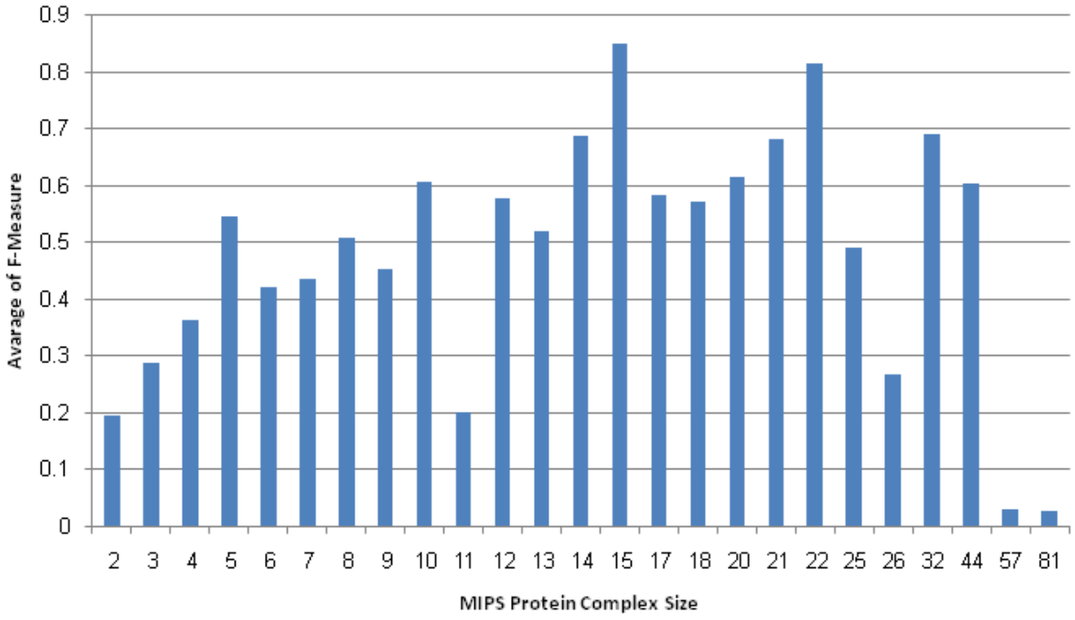


Fig. 6. f-measure between finding subgraphs and experimental protein complexes based on its size.

6. Conclusion

In this paper, we presented a FLVQ algorithm as a robust tolerable method to find dense subgraphs in PPI networks as protein complexes. The algorithm identifies more than 200 dense subgraphs having more overlap among experimentally known protein complexes. By clarifying the structure of protein interactions network, uncharacterized proteins could be predicted by the functions of other known proteins which belong to same clusters. By using line graph transformation, we eliminated the scale-free degree distribution in PPI network which caused larger number of dense highly connected subgraph revealed. There is overlapping between found subgraphs that express the results are more conforming with the reality nature of protein complexes.

7. References

Adamcsek, B. (2006). Cfinder: Locating Cliques And Overlapping Modules In Biological Networks, *Bioinformatics*, Vol. 22, pp. 1021-1023.

Bader, G. & Hogue, C. (2003). An Automated Method For Finding Molecular Complexes In Large Protein Interaction Networks, *BMC Bioinformatics*, Vol. 4

Bader, G. & Hogue, C. (2003). Analyzing Yeast Protein-Protein Interaction Data Obtained From Different Sources, *Nat. Biotechnol*, Vol. 20, pp. 991-997

Bezdek, C. &Hathaway,J. (1995). Optimization Of Clustering Criteria By Reformulation, *IEEE Transactions on Fuzzy Systems*, Vol. 2, pp. 241-246.

Bezdek, C.; (1973). Fuzzy Mathematics In Pattern Classification, *Ph.D. dissertation, Dept. Appl. Math.,Cornell Univ., Ithaca, NY*

Chua, H.; Ning, K.; Sung, W.; Leong. H., (2008). Using Indirect Protein–Protein Interactions For Protein Complex Prediction, *Journal of Bioinformatics and Computational Biology*, Vol. 6., pp. 435-466.

Gao, L.; Sun, p.; Song, j. (2009). Clustering Algorithms For Detecting Functional Modules In Protein Interaction Networks, *Journal of Bioinformatics and Computational Biology*

Guldener, U. (2005). CYGD: The Comprehensive Yeast Genome Database, *Nucleic Acids Res*, Vol. 33, pp. 364–368.

Hartigan, J.A. (1975). Clustering Algorithms. *New York : Wiley*

Ito, T.; Chiba, T.; Ozawa, R.; Yoshida, M. ; Hattori, M.; Sakaki, Y. (2001). A Comprehensive Two-Hybrid Analysis To Explore The Yeast Protein Interactome, *PNAS*, Vol. 98, pp. 4277-4278.

Kohonen, T. (1990). The Self Organizing Map, *IEEE Proc*, Vol. 78

Lin, C.; Cho, Y.; Hwang, W.; Pei, P.; Zhang, A. (2006). Clustering Methods In Protein-protein Interaction Network, In: *Knowledge Discovery in Bioinformatics: Techniques, Methods and Application*. John Wiley & Sons

Mewes, H.W. et al., (2004). MIPS: Analysis And Annotation Of Proteins From Whole Genomes, *Nucleic Acids Res*, Vol. 32, pp. D41-D44

Mirny, V. & Spirin, L. (2003). Protein Complexes And Functional Modules In Molecular Networks, *Proc. Natl Acad. Sci*, Vol. 100(21), pp. 12123–12126

Newman, M. & Girvan. M., (2004). Finding And Evaluating Community Structure In Networks, *Physical Review*, 2004.

Palla, G.; Der´enyi, I.; Farkas, I.; Vicsek, T., (2005). Uncovering The Overlapping Community Structure Of Complex Networks In Nature And Society. , *Nature*, Vol. 435, pp. 814–818

Pizzuti, C.; Rombo, S. (2007). Multi-functional Protein Clustering in PPI Networks, *International Conference on Intelligent Data Engineering and Automated Learning*

Przulj, N. (2005). *Knowledge Discovery in Proteomics Graph Theory Analysis of Protein-protein Interactions*, Department of Computer Science, University of Toronto

Ravuri, N. & Karayiannis, M. (1995). An Integrated Approach To Fuzzy Learning Vector Quantization And Fuzzy C-Means Clustering, *New York : Intelligent Engineering Syslems Through Artificial Neural Networks*, Vol. 4, pp. 247-252.

Schaeffer, S. (2007). Survey Graph Clustering, *Elsevier*

Vlasblom, J. & Wodak., S. (2009). Markov Clustering Versus Affinity Propagation For The Partitioning Of Protein Interaction Graphs, *BMC BIOINFORMATICS*, Vol. 10.

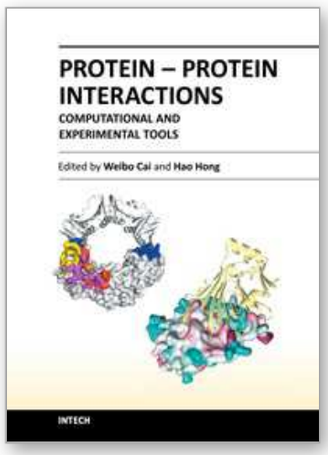
Wu, M.; Li, X.; Kwoh. C. (2008). Algorithms For Detecting Protein Complexes In PPI Networks: An Evaluation Study, *PRIB08*, pp. 135-146.

Xenarios, I.; Salwinski, L.; Duan, X.; Higney, P.; Kim, S.; Eisenberg, D. (2002) DIP, The Database Of Interacting Proteins: A Research Tool For Studying Cellular Networks Of Protein Interaction, *Nucleic Acids Res*, Vol. 30, pp. 303-305.

Zadeh L., A. (1965) Fuzzy Sets, *Inf. Control*, Vol. 8, pp. 338-353.
Zhang, A., (2009). Modularity Analysis of Protein Interaction Networks, *Protein Interaction Networks Computational Analysis* , pp. 66-77

IntechOpen

IntechOpen



Protein-Protein Interactions - Computational and Experimental Tools

Edited by Dr. Weibo Cai

ISBN 978-953-51-0397-4
Hard cover, 472 pages
Publisher InTech
Published online 30, March, 2012
Published in print edition March, 2012

Proteins are indispensable players in virtually all biological events. The functions of proteins are coordinated through intricate regulatory networks of transient protein-protein interactions (PPIs). To predict and/or study PPIs, a wide variety of techniques have been developed over the last several decades. Many in vitro and in vivo assays have been implemented to explore the mechanism of these ubiquitous interactions. However, despite significant advances in these experimental approaches, many limitations exist such as false-positives/false-negatives, difficulty in obtaining crystal structures of proteins, challenges in the detection of transient PPI, among others. To overcome these limitations, many computational approaches have been developed which are becoming increasingly widely used to facilitate the investigation of PPIs. This book has gathered an ensemble of experts in the field, in 22 chapters, which have been broadly categorized into Computational Approaches, Experimental Approaches, and Others.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hamid Ravaee (2012). Finding Protein Complexes via Fuzzy Learning Vector Quantization Algorithm, Protein-Protein Interactions - Computational and Experimental Tools, Dr. Weibo Cai (Ed.), ISBN: 978-953-51-0397-4, InTech, Available from: http://www.intechopen.com/books/protein-protein-interactions-computational-and-experimental-tools/finding-protein-complexes-via-fuzzy-learning-vector-quantization-algorithm

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen