Chapter 3

Protein Complex Finding Methods

3.1 Introduction

Any cellular function in the living body is brought about by the interactions among proteins. These interactions can be static or dynamic in nature. Groups of proteins which dynamically interact in nature are termed as functional modules, whereas those groups of proteins which are assumed to interact irrespective of time are referred to as protein complexes. The task of identifying protein complexes has become much popular compared to functional modules due to the inherent inability of PPI detection methods to report time related information. PPI detecting methods such as Yeast-two-Hybrid and TAP-MS are inadequate in capturing the spatial and temporal information regarding interacting proteins [80]. Therefore, a large fraction of the research community is working in the direction of identifying protein complexes only. Protein complex identification has been targeted as a clustering problem, which deals with identifying densely connected groups of proteins from a PPI network.

Identified complexes can be used to analyze the structural and functional properties of a PPI network. They can also be used in tracking down the evolutionary orthology pattern [131]. For example, a protein complex consisting of cytochrome C protein is known to aid in aerobic respiration in humans. This protein is also found in mice and fish and is associated with the same function. They can also be used for predicting functions of uncharacterized proteins. Identification of protein complexes from PPI network can be modelled in the form of a mathematical problem as follows.

Problem formulation *Given a graph $G = (V, E)$ corresponding to a PPI network (where V represents proteins and E represents interactions among the proteins) and a set of benchmark complexes, B_C (where each element of B_C is a group of proteins), the task is to extract groups of vertices (clusters), C in G using some internal criteria defined on G, such that matching between B_C and C is maximized.*

3.2 Related Work

A number of methods have been proposed in literature tp detect protein complexes as dense subgraphs in a PPI network. *MCODE* [4], a well established protein complex finding method works in three steps-vertex weighting, molecular prediction and post-processing. In the first step, all vertices are weighed using the concept of highest k-core concept of the vertex neighborhood. Weighing the vertices itself requires two more steps - calculating the density of the highest k-core of the immediate neighorhood of v, known as the core clustering coefficient and assigning the final score as the product of the core clustering coefficient with the highest k-core level of the successive neighorhood of the vertex. The second step involves starting with the highest score seed vertex and expanding it recursively by adding those vertices in the cluster whose score is higher than some threshold. This process is repeated until no further nodes are left satisfying this criterion. The next cluster begins with the next highest score which is yet to be seen and cluster expansion continues. The third and final step is an optional post-processing step, which involves some kind of filtering of these clusters. The filtered clusters are returned as complexes in this method. *FAG-EC* [79], which is an agglomerative approach starts with singleton vertices as seed nodes for clusters. These vertices are expanded using a queue. This queue holds the edges in ascending order of their weights. At each iteration, the top of the queue is checked w.r.t. certain criteria such as: if two edges connect the same cluster elements, then they belong to the same cluster otherwise they belong to different clusters. This generates the sequence of complexes. Another method, *FT* [40] is a two step process involving a hierarchical and a transfer procedure for complex generation. During the hierarchical stage, each atomic class is joined with another and a score is calculated. Two classes are joined only if they lead to a maximum score. This process continues until no further fusions can be carried out. The next stage involves a transfer of elements from one class to another. If the value of an element in a class is not maximized,

it is transferred to a different class so as to maximize its contribution. If none of the transfers leads to a higher score, the element is returned as a singleton set. This method leads to the possibility of throwing out certain proteins which do not take part in complex fomation. *TFit* [33], a complex finding method works on the principle of clique finding. In this technique, the vertices of a cluster are transferred among many clusters, until there is a change occurring in the modularity function, defined by the ratio of the outward edges to the inward edges. Once this is done, the group of vertices belonging to the same cluster are merged. A non-exclusive complex finding method called *OCG* [11] is based on a partitioning process which begins with an initial set of overlapping classes. These classes are hierarchically merged among themselves provided the merging leads to an increase in modularity function. The process stops either when the expected number of clusters are formed or the modularity is maximized. However, there is a slight trick involved in constraining the process using maximum modularity. The modularity function may decrease or may not be repetitive w.r.t. the initial class. Therefore, an optimization function is used to assign each element to its proper class depending on its contribution. This step further refines the performance of the method by eliminating loosely coupled elements. A heuristic method called *QCUT* [121] uses the KCUT method to divide the graph into subnetworks. These subnetworks are then compared among themselves. The next stage involves refinement of these subnetworks based on modularity. The modularity of a network can be improved either by adding a new vertex into a community or by transferring an existing vertex to another community. *ClusterONE* [101] is one of the most remarkable methods for complex finding. It is based on a greedy approach of seed selection and expansion. The first step begins with a seed node, which grows by adding or removing vertices to find groups with high cohesiveness. Cohesiveness is decided by the number of edges within and outside the cluster. This process is repeated for multiple seeds. In the next step, the overlap extent among groups is calculated and groups with overlap score greater than some threshold are merged into one. The last step involves eliminating groups with less than three proteins or whose density is below a certain threshold from the list of complexes. PEWCC [169] uses the weighted clustering coefficients to predict complexes. A two step process, TINCD [102] uses the clustering results along with the TAP data to predict complexes. CPredictor2.0 [161] assumes a set of clusters based on functional similarity between proteins. It then uses Markov clustering to discover complexes from the assumed groups.

A number of methods for protein complex finding have been published. These methods are basically seed expanding methods which require at least two criteria-

one for seed selection and another for its expansion. One can use a set of criteria either in sequential or parallel fashion. However, in this domain, much work uses the serial approach for complex finding. Due to widespread use of protein complexes in predicting uncharacterized protein functions and also their disorientation in disease states, it is an utmost priority to accurately identify these complexes. The more accurate is the complex finding process, the higher are the chances of extracting relevant and appropriate disease related information. Experimental studies show that a proper combination of topological and biological properties leads to an increased accuracy of the protein complex finding methods. An optimal subset of properties for this problem can be obtained using the multi-objective optimization approach when using multiple criteria in any problem, the goal is to solving is to get a fair trade-off among the criteria so that the results are not compromised. This logic has led a few researchers to use optimization technique to get biologically significant complexes. The first work called *PROCOMOSS* (Protein Complex Detection using Multi-objective Evolutionary Approach based on Semantic Similarity) [99] uses a combination of density and a number of interconnecting nodes not present in a cluster for the topological feature and semantic similarity for the biological feature. The density and semantic similarity of complexes are maximized so as to get densely connected complexes from the graph, whereas the number of interconnecting nodes not present in the cluster is minimized so as to get well separated clusters. In order to maximize the functional significance of complexes, it uses a combination of Lin, Jiang and Conrath and Kappa measure of semantic similarity between twoproteins. This set of features is optimized using the NSGA II optimization technique. Another recent work by Bandyopadhya et al. [5] used three topological properties–density, contribution of a node to a cluster and closeness centrality, and a functional property which is the semantic similarity among proteins to decide upon the set of complexes. In this method, they used a Relevance Semantic Similarity to calculate the similarity among proteins. They also used the NSGA II optimization technique [24] to get an optimal subset of these parameters based on a fitness score. A summary of some existing complex finding techniques is given in Table A.2.

3.3 Motivation

Various classes of methods can be used to find protein complexes from a PPI network. These methods are either solely based on topological properties or are

Table 3.1 Summary of some protein complex finding techniques

Category	**Method**	**Salient feature**	**Datasets Used**	**Availability**
Serial	MCODE [4]	Works on the highest k-core concept to find complexes	Gavin_2002, MIPS,SGD	ClusterViz (Cytoscape)
	FAG-EC [79]	Based on an agglomerative approach with members stored in queue	DIP	ClusterViz (Cytoscape)
	FT [40]	Based on a hierarchical and transfer approach	Plasmodium	ClustnSee (Cytoscape)
	TFit [33]	Based on fusion and transfer approach on clique partitioning problem	Plasmodium	ClustnSee (Cytoscape)
	OCG [11]	Requires overlapping classes as input	Yeast	ClustnSee (Cytoscape)
	QCUT [121]	based on KCUT method and maximizes modularity	MIPS	CommFinder (Cytoscape)
	ClusterONE [101]	Based on a cohesiveness measure to identify complexes	MIPS, SGD	ClusterONE (Cytoscape)
Parallel	PROCO MOSS [99]	Uses a combination of density, number of interconnecting nodes and semantic similarity as objective functions	DIP, MIPS	-
	Bandyopadhya et al. [5]	Uses a combination of density, contribution of nodes in a cluster, closeness centrality and semantic similarity as objective functions	HPRD	Matlab code

based on a combination of both topological and biological features. An empirical analysis [131] on eight existing methods using four yeast datasets was done to find the most appropriate method that performed well in all situations. Unfortunately none of these existing methods could live up to my expectations. These methods were highly parameter dependent and had to be fine-tuned to obtain good results. Further, no clear conclusion could be drawn on the performance of each of these methods. Therefore, I tried analyzing the PPI network using an obvious feature, connectivity. My aim was to design a method which was based on not more than two parameters and was able to perform consistently well over all datasets. In this process, I designed the CNCM approach. This method uses only two parameters and the results did not show much fluctuations with changing parameters. However, it did not perform well on the HPRD dataset and was beaten by ClusterONE in terms of accuracy. Another line of work discussed in this chapter is based on the use of multi-objective optimization. Bandyopadhya et al. proposed a method based on optimizing both topological and biological properties of a PPI network during complex finding. In their method, they used density, contribution of a node and closeness centrality as the topological measure and Relevance Semantic Similarity for the biological measure. I did an empirical analysis on the different centrality measures [132] and found that none of these measures were effective in analyzing PPI networks. Therefore, I proposed a method called DCRS, which works on the same framework as that of Bandyopadhya's except changing the topological features. I used reachability contribution instead of closeness centrality because reachability of nodes is already established to be effective in identifying the importance of genes in yeast [41]. I also used Wang's semantic similarity for the biological property as it is known to be the best available semantic similarity measure [151]. I successfully used DCRS on the HPRD dataset and found an accuracy of around 48%, which is higher than Bandyopadhya's method. Thus, my target of obtaining quality complexes from human PPI dataset is achieved by my proposed method.

3.4 Contributions

In this chapter, I make the following contributions.

- A method called CNCM is proposed to detect protein complexes from a PPI network. This method is based solely on the topology of the inherent network.
- I have proposed a multi-objective method called DCRS for the same. This

method uses NSGA II to optimize the topological properties of PPI network to get quality complexes.

3.5 Protein complex finding based on topological information: CNCM

A PPI network is a visual representation of nodes (proteins) and their associations. The topology of a network mainly decides the pattern of subgroups which might exist in the network. A careful analysis of this physical appearance might lead to identification of groups of proteins (known as complexes) which coordinate to achieve certain functions.

The methods discussed so far use topological properties of a PPI network to detect protein complexes. However, these methods use a number of parameters which are difficult to tune to obtain quality complexes. Therefore, I have proposed a method called CNCM which uses just two parameters for effective complex detection. Following are the notable features of CNCM.

- CNCM uses a combination of topological properties of a PPI network to detect complexes.
- Unlike other methods, it uses only two parameters for complex finding. These parameters are simple and easy to use.
- CNCM detects overlapping complexes, which are characterized by the multi-functional nature of proteins.
- This method detects sparse complexes which are otherwise difficult to to detect.

CNCM is a graph-based approach for identifying protein complexes. This method ensures detection of effective complexes w.r.t. benchmark dataset. It also ensures detection of sparse and overlapping complexes. This technique has been used with four yeast datasets and the results are satisfactory in terms of precision, recall and f-measure as well as in biological terms of co-localization score and p-value.

3.5.1 Proposed Method : CNCM

Clustering is an effective way of grouping elements based on similarity. We propose a method called CNCM (Connectivity based Network Clustering Method) for complex finding. It uses two properties, connectivity and clustering coefficients of nodes in the PPI network. Following definitions are used in this algorithm.

Definition 1 (Neighborhood). *A node $v_i \in V$ is said to be a neighbor of node $v_j \in V$ if v_i and v_j are connected by an edge e.*

Definition 2 (Degree of a node). *The degree d of a node $v_i \in V$ represents the number of partners (connected nodes) v_i has in G.*

Definition 3 (Clustering coefficient). *The clustering coefficient of a node $v_i \in V$ is the ratio of the number of links among the neighbors of v_i, i.e., l_{v_i} to the total number of possible links among its neighbors, k_{v_i}.*

$$CCf(v_i) = \frac{2l_{v_i}}{k_{v_i}(k_{v_i} - 1)} \tag{3.1}$$

Definition 4 (Connectivity). *The connectivity of a node $v_i \in V$ to a subgraph G' is defined as the ratio of the number of links, l_{v_i} shared between the v_i and the members of the subgraph G' to the degree of the node, d_{v_i}.*

$$Connt(v_i, G') = \frac{l_{(v_i,G')}}{d_{v_i}} \tag{3.2}$$

Definition 5 (Core). *A core of a complex C_i is defined as a node (or protein) $v_i \in V$ such that the clustering coefficient, $CCf(v_i) \geq CCfT$, where $CCfT$ is a user defined threshold and $CCf(v_i)$ is the maximum among all nodes in C_i.*

Definition 6 (Core complex). *A core complex, $C_i = \{v_1, v_2, v_3\}$ is a set of three nodes (one is the core, v_1 and two other nodes, v_2, v_3) whose connectivity is highest among themselves compared to other nodes in complex C_i.*

Definition 7 (Periphery). *A node v_i is considered a periphery node (or protein) or a border node in a complex, C_i, if it is loosely attached to C_i, i.e., $d_{v_i} = 1$ and $Connt(v_i, C_i) = 1$.*

Definition 8 (Protein Complex). *A protein complex is a subgraph $G' \subseteq G$ with at least a core complex attached to other nodes v_k such that $Connt(v_k, G') \geq \alpha$ (α is a user defined threshold) which is also greater than connectivity of all other nodes $v_l \in V$.*

Definition 9 (Neighbor of Complex)**.** *A node v_i is a neighbor of a complex C_i, if $\exists v_m \in C_i$ such that $Ng(v_m) = v_i$.*

CNCM is a seed expanding technique for finding complexes from a PPI network. It takes $CCfT$ and α as user defined thresholds. It starts with assigning all its elements to a data structure called *RemList*. The first step involves finding the clustering coefficient of each element in the *RemList*. The cluster finding process then begins with the node v_h having the highest clustering coefficient among all other elements in *RemList*. It then checks if $CCf(v_h) \geq CCfT$ and calls it the core protein, if successful. This element is inserted into pC, another data structure to hold track of cluster elements. The next two elements expand the core protein depending on connectivity values. At this stage, these three nodes in pC gives rise to a core complex. The assumed core complex size is three because finding good intra-node and inter-cluster connectivity values for clusters with less than three elements is very difficult. The next step involves expansion of the core complex with other nodes in *RemList* in terms of decreasing connectivity values w.r.t. the pC. This process is constrained by the use of another user defined threshold, α. A node, v_n gets added to the pC only if $Connt(v_n, pC) \geq \alpha$. This process iterates until no new node satisfies the α criterion. The elements in the pC are returned as a *Cluster* if pC has more than three elements. The next cluster formation begins by choosing a new seed node from the set of remaining nodes in *RemList* and the whole process is repeated to get another set of clusters. This process continues until no more nodes are left satisfying the $CCfT$ cut off. The algorithm for CNCM is given in Algorithm 1.

To analyze the effectiveness of CNCM, we present the following three propositions.

Proposition 1. *CNCM detects overlapping complexes.*
Explanation: CNCM expands a cluster using a node-addition approach. Node v_m gets added to the core complex, C_i, if $Connt(v_m, C_i) \geq \alpha$. A number of clusters are formed in a similar fashion. If any node $v_a \in C_i$ also satisfies *connectivity* criteria w.r.t. C_j (where both C_i and C_j are core complexes), it is added into the cluster C_j . Hence, clusters identified by CNCM may overlap.□

Proposition 2. *CNCM can detect sparse (or small) complexes effectively.*
Explanation: Sparse or smaller complexes are ones with two to three nodes [137]. CNCM starts with a seed core complex containing three nodes. The core complex is then expanded to include more nodes depending on the connectivity threshold (α)

Input : $G = \{V, E\}$, (PPIN); $CCfT$, (Clustering coefficient threshold);
α, (Connectivity threshold)
Output: $Cluster = \{C_1, C_2, \cdots, C_N\}$, (set of N complexes)

Initialize $\mathsf{RemList} = V, \mathsf{Cluster} = NULL, ccount = 1$;
foreach $v_i \in V$ **do**
| Compute $\mathtt{CCf}(v_i)$;
end
while $|\mathsf{RemList}| \geq 3$ **do**
$\mathsf{pC} = NULL$;

//Find Core Protein, v_i
Choose $v_i \in \mathsf{RemList}$ such that $\forall\ v_j \in \mathsf{RemList}, \mathtt{CCf}(v_j) \leq \mathtt{CCf}(v_i)$;
if $\mathtt{CCf}\ (v_i) < \mathsf{CCfT}$ **then**
| Exit;
end
$\mathsf{pC} = v_i$; $\mathsf{RemList} = \mathsf{RemList} - v_i$;

//Find Core Complex of v_i
Choose $v_j \in \mathsf{RemList}$ such that $\forall v_m \in \mathsf{RemList}$ and
$\mathtt{Connt}(v_j, \mathsf{pC}) \geq \mathtt{Connt}(v_m, \mathsf{pC})$;
$\mathsf{pC} = \mathsf{pC} \bigcup v_j$;
$\mathsf{RemList} = \mathsf{RemList} - v_j$;
Choose $v_k \in \mathsf{RemList}$ such that $\forall v_m \in \mathsf{RemList}$ and
$\mathtt{Connt}(v_k, \mathsf{pC}) \geq \mathtt{Connt}(v_m, \mathsf{pC})$;
$\mathsf{pC} = \mathsf{pC} \bigcup v_k$;
$\mathsf{RemList} = \mathsf{RemList} - v_k$;

//Expand Core Complex of v_i
Choose $v_n \in V - \mathsf{pC}$ and $v_n \in \mathtt{NgCp}(\mathsf{pC})$ such that
$\forall v_l \in V - \mathsf{pC}, \mathtt{Connt}(v_n, \mathsf{pC}) \geq \mathtt{Connt}(v_l, \mathsf{pC})$;
while v_n *exists and* $\mathtt{Connt}(v_n, \mathsf{pC}) \geq \alpha$ **do**
| $\mathsf{pC} = \mathsf{pC} \bigcup v_n$;
| $\mathsf{RemList} = \mathsf{RemList} - v_n$;
| Choose next v_n;
end
Mark pC as C_i;
$\mathsf{Cluster}(ccount) = \mathsf{Cluster} \bigcup C_i$;
ccount++;
end
Return $\mathsf{Cluster}$;

Algorithm 1: Steps involved in CNCM Algorithm

set by the user. The minimum size of a core complex is set to three because smaller size clusters unnecessarily add to the processing time and are usually redundant in nature. If no new node exists satisfying the connectivity criterion of getting into the pC, the core complex is returned as the cluster. Hence CNCM is capable of detecting small or sparse complexes. □

Proposition 3. *CNCM detects periphery proteins.*
Explanation: A periphery protein is a node in the PPIN with a very low connectivity with the complex. Low connectivity generally implies one link between a node and the pC. In CNCM, a node which is connected to a complex with a single link has *connectivity* value one and hence considered a member of the complex and therefore detected. □

Computational Complexity In a given network comprising of n nodes, computing the clustering coefficient requires $O(n_{unique} \times {n'_{unique}}^2)$ time, where n'_{unique} is the average number of neighbors of a node and n_{unique} is the number of unique elements in the graph. To expand the clusters using connectivity value each time takes $O(n_{unique} - 1)$ time. This is an iterative process and requires $O(n^2_{unique})$ time. Thus the overall complexity is $O(n_{unique} \times {n'_{unique}}^2) + O(n^2_{unique}) \equiv O(n^2_{unique})$.

3.5.2 Experimental Results

CNCM has been implemented in MATLAB running on an HP Z800 workstation with two 2.4 GHz Intel(R) Xeon (R) processors and 12 GB RAM, using the Windows 7 operating system. The method has been evaluated on four yeast datasets– Gavin_2002, Gavin_2006, Krogan_2006 and Tong_2004. A detail description of these datasets is given in Subsection 2.1.6.1 of Chapter 2. Results of this method is compared with few existing methods like MCODE [4], FAG-EC [79], FT [40], TFit [33], OCG [11], QCUT [121], ClusterONE [101] and GMFTP [171] using MIPS as benchmark dataset. The details of the benchmark set is given in Subsection 2.1.7.1 of Chapter 2.

Performance measures based on Precision, Recall and F-measure In order to compare the effectiveness of the method with others, indices such as precision, recall and f-measure are used. A predicted cluster cannot exactly match a benchmark complex. So an overlapping threshold between the predicted cluster and the benchmark set is used to decide upon the effectiveness of prediction. Two overlapping schemes are commonly used in the literature– Bader's scheme [4] and Wang's scheme [153]. The details of these schemes are discussed in Subsection 2.1.9

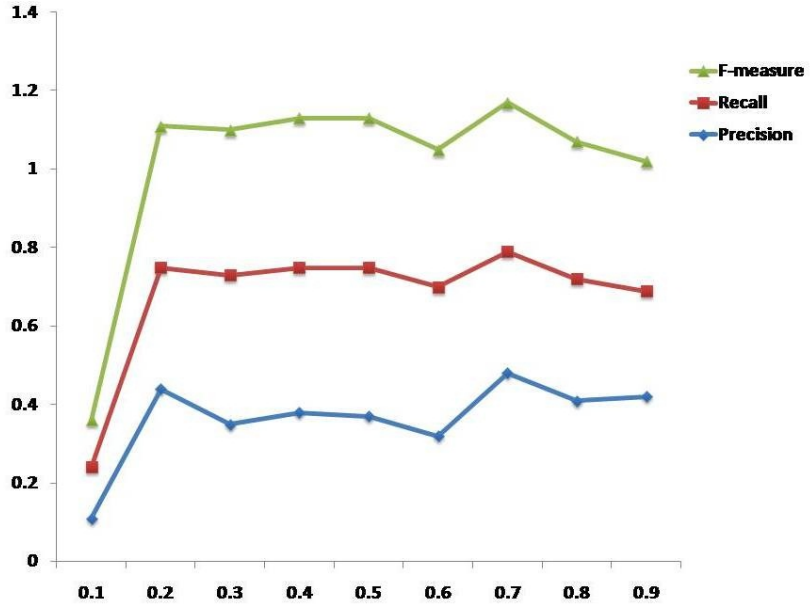
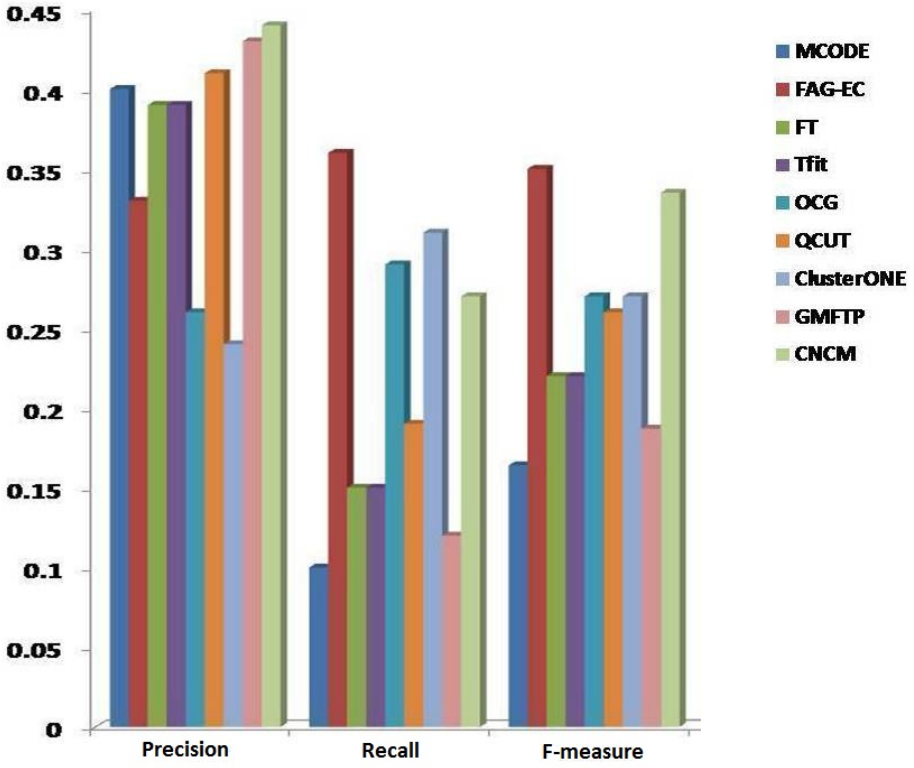


Figure 3.1. Performance of CNCM for various α threshold values for Gavin_2002 dataset.

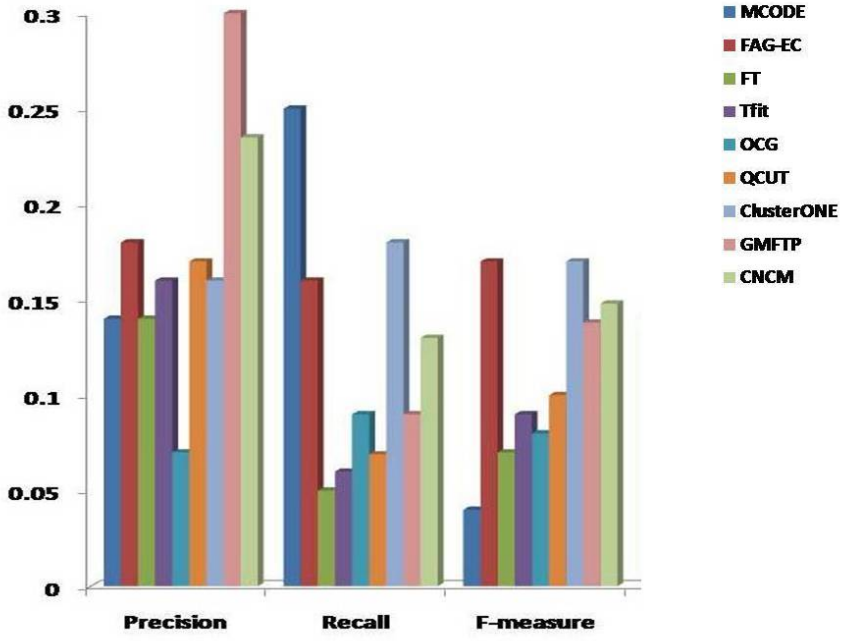
of Chapter 2. During validation, most techniques [4, 153] use threshold values of 0.2 and 0.6 for Bader's scheme and Wang's scheme, respectively. I therefore use the same threshold values to make a fair comparison of my method with the other methods. In order to fix the threshold value for reporting the results of CNCM, I varied the α value in the range 0.1-0.9 as shown in Figure 3.1. In Figure 3.1, we see that CNCM shows stable performance for threshold value 04-0.7. However, the performance has shown a decline at 0.6 and a rise again at 0.7. Therefore, for further comparison, the results obtained at the $\alpha = 0.4$ are used.

Graphs 3.2 - 3.5 show the precision, recall, and f-measure values for CNCM and other methods for the four yeast datasets. From Figures 3.2 - 3.5, we conclude that no single method performs well in terms of these indices over all datasets. MCODE performs well in some cases but it detects very large size cluster, thereby misses smaller significant groups. FT and TFiT both rely on the modularity calculation during complex formation. ClusterONE is based on a greedy approach using cohesiveness whereas GMFTP uses a complex combination of both functional and topological information for complex detection. Considering the complexity of these algorithms, CNCM is based on a simple concept of connectivity between nodes to detect complexes. Using this trivial analysis, it has been found to show better performance than some of these existing methods.

The remarkable performance of CNCM is shown as higher average f-measure

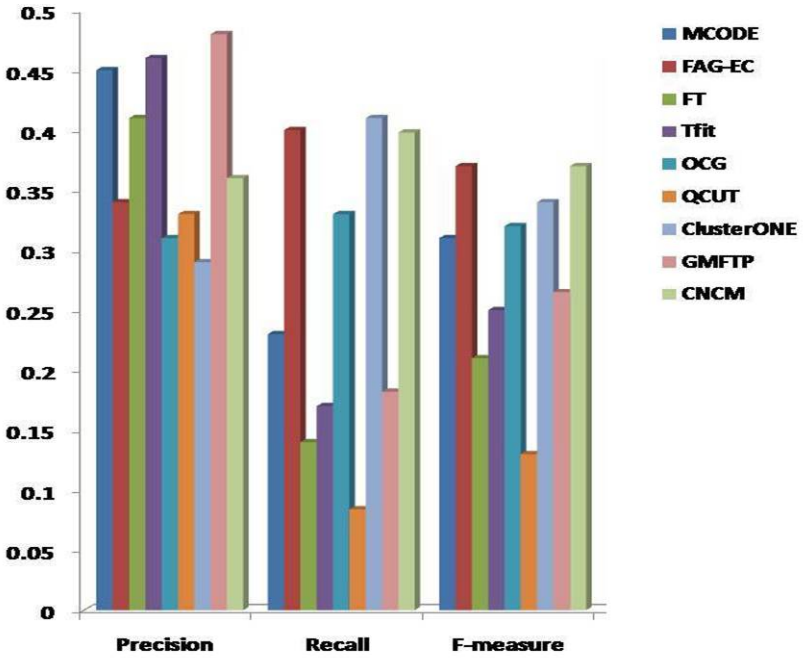


(a) At Bader's threshold=0.2

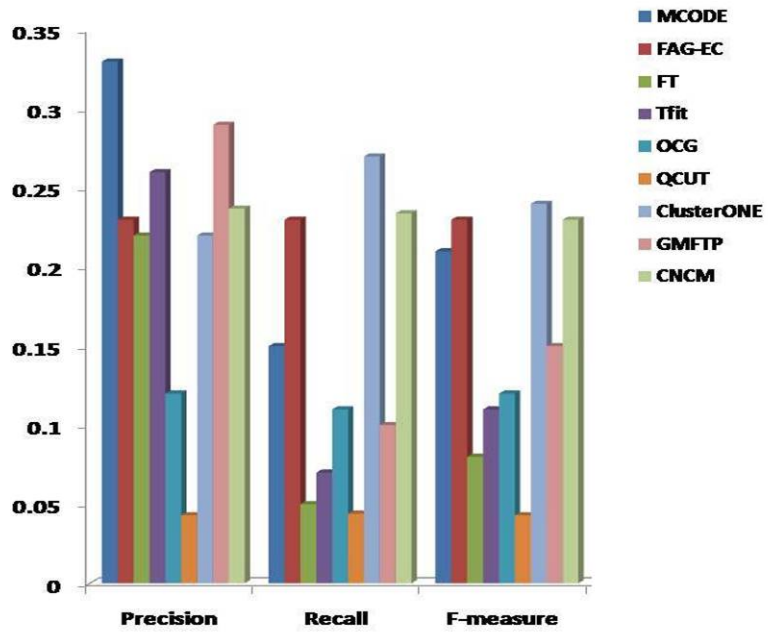


(b) At Wang's threshold=0.6

Figure 3.2. Precision, Recall and F-measure of CNCM and other algorithms on the Gavin_2002 dataset using MIPS as benchmark.

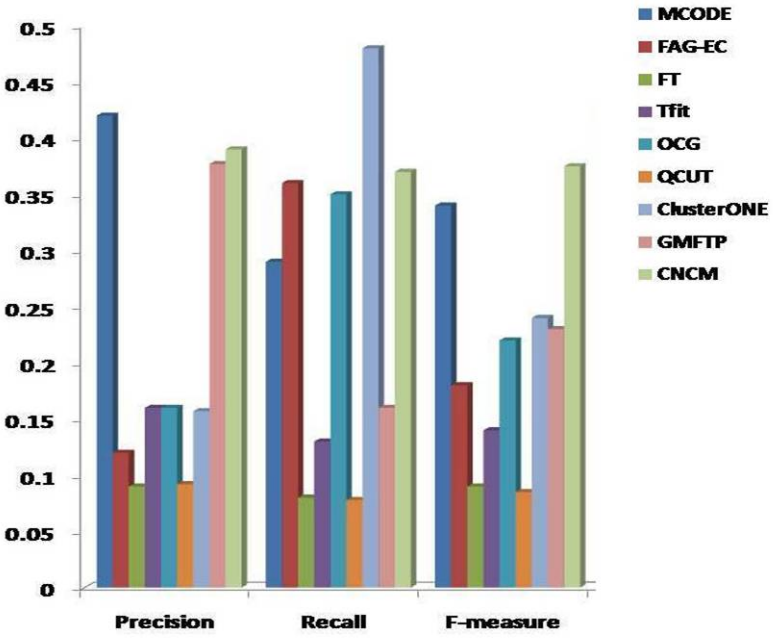


(a) At Bader's threshold=0.2

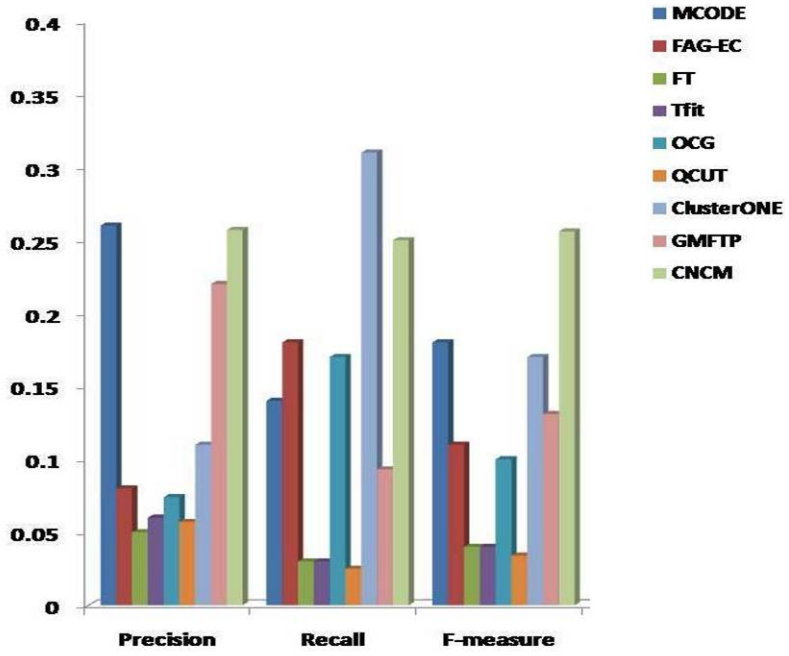


(b) At Wang's threshold=0.6

Figure 3.3. Precision, Recall and F-measure of CNCM and other algorithms on the Gavin_2006 dataset using MIPS as benchmark.

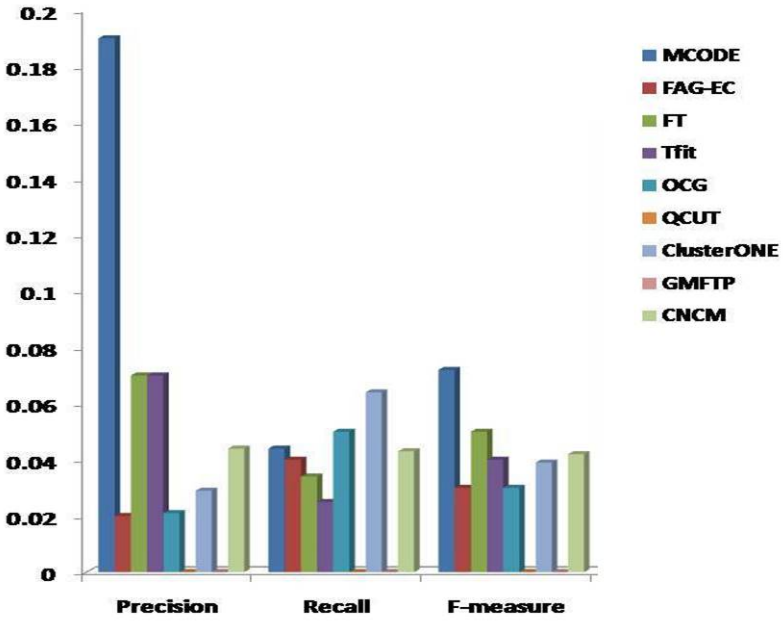


(a) At Bader's threshold=0.2

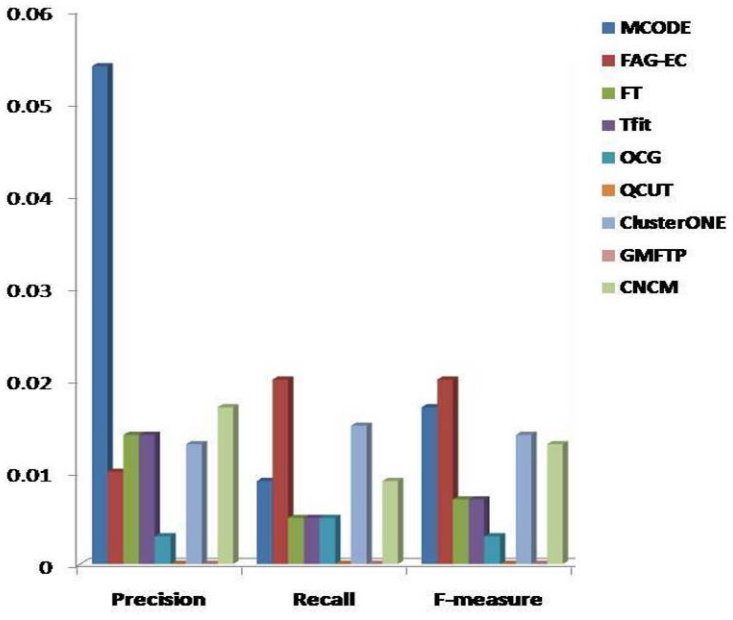


(b) At Wang's threshold=0.6

Figure 3.4. Precision, Recall and F-measure of CNCM and other algorithms on the Krogan_2006 dataset using MIPS as benchmark.



(a) At Bader's threshold=0.2



(b) At Wang's threshold=0.6

Figure 3.5. Precision, Recall and F-measure of CNCM and other algorithms on the Tong_2004 dataset using MIPS as benchmark.

compared to all other methods as seen in Figure 3.6. A high value of f-measure indicates the superiority of the method in detecting biologically significant clusters. In Figure 3.6, we see that CNCM is at the top followed by FAG-EC and MCODE for Krogan_2006 dataset. Hence, we can say that CNCM gives significant results and is more suitable for protein complex detection than the other algorithms.

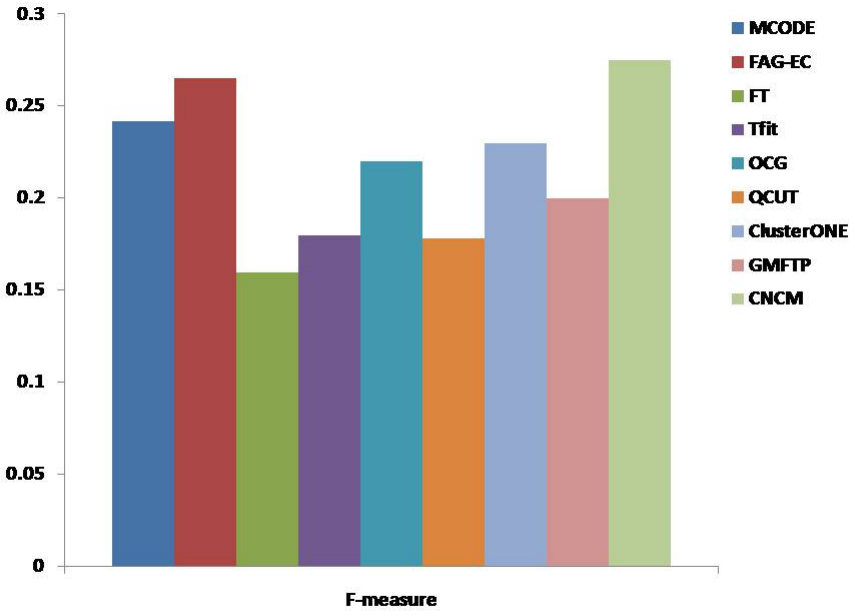
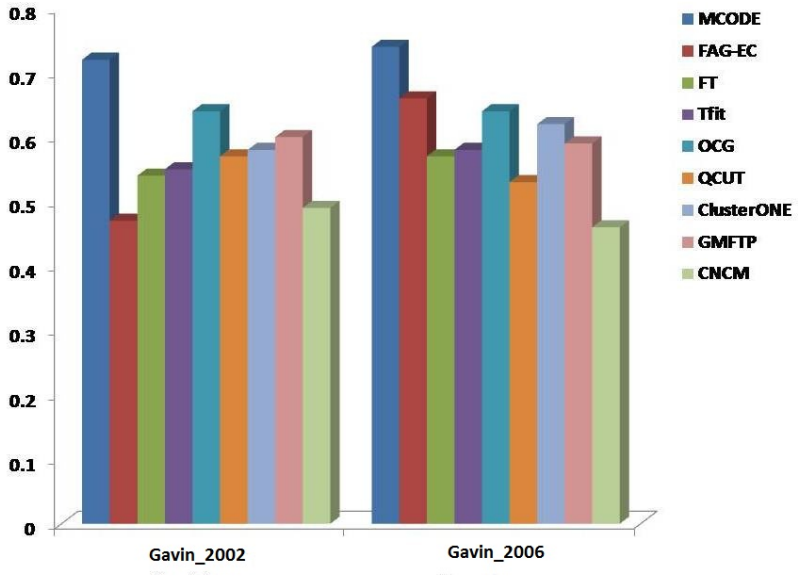


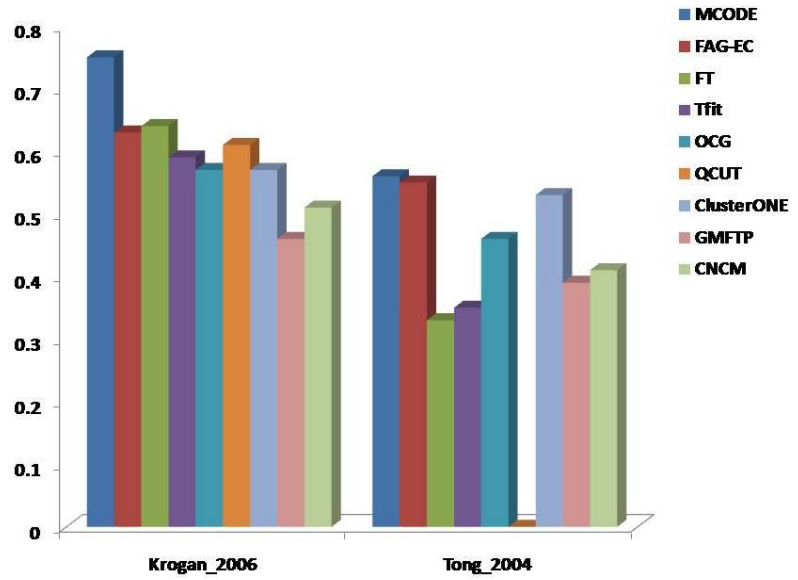
Figure 3.6. Average F-measure using Bader's overlapping scheme with MIPS as benchmark on Krogan_2006 dataset (Higher value implies better performance)

Performance measure in terms of co-localization score Co-localization score is used to evaluate the effectiveness of complexes found by CNCM and and other algorithms. The co-localization measure computes the proximity between proteins in a complex. Details are discussed in Subsubsection 2.1.9.3 of Chapter 2. ProCope tool is used to calculate this score over two localization datasets- Huh et al. and Kumar et al., details of which are discussed in Subsection 2.1.7.2 of Chapter 2. Figures 3.7 and 3.8 shows the co-localization scores of the four datasets using the two localization data. In Figure 3.7, we see that MCODE has the best co-localization score among all the existing methods for the Huh et al. benchmark dataset. In Figure 3.8, CNCM is the winner among all four datasets using Kumar et al. co-localization set.

Validation using Gene Ontology Gene Ontology is a repository which stores genes, its associated GO terms and their associations with other GO terms corresponding to other genes in the form of a tree. GO can be used to find how two genes



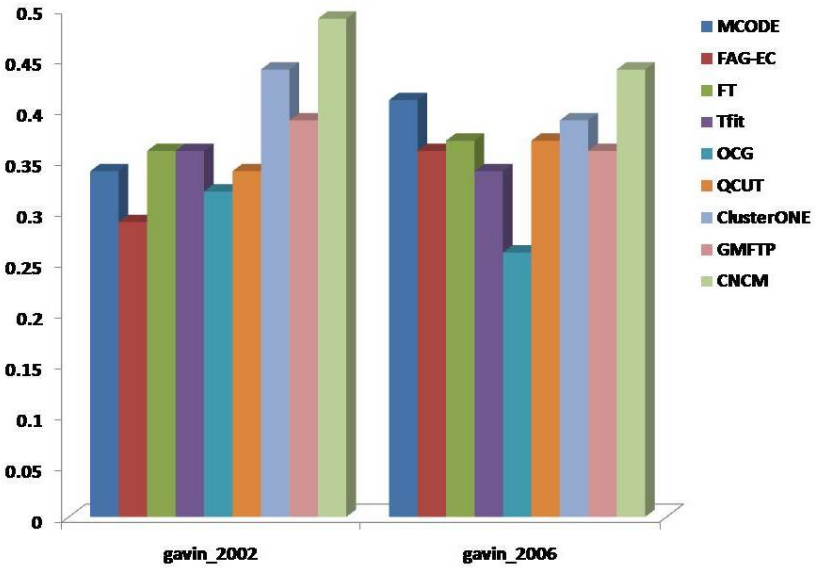
(a) Co-localization score over Gavin_2002 and Gavin_2006 datasets using Huh et al. localization data



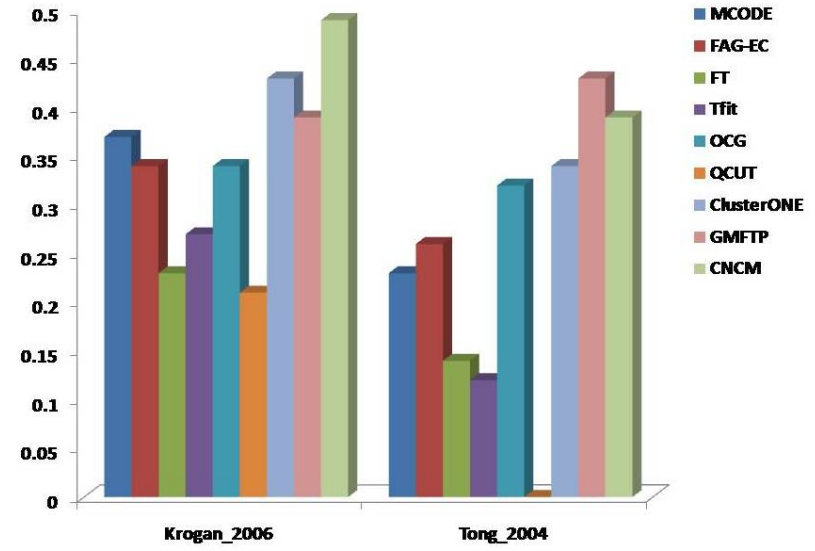
(b) Co-localization score over Krogan_2006 and Tong_2004 datasets using Huh et al. localization data

Figure 3.7. Co-localization score for four yeast datasets using Huh et al., localization dataset (Higher value implies better performance).

(proteins) are related semantically. This is the functional enrichment of genes, i.e., how many genes work together to achieve a certain biological function, which is calculated by p-value. p-value and its calculation in described in Subsubsection 2.1.9.4 of Chapter 2. We use the BinGO plugin [89] of Cytoscape to compute p-



(a) Co-localization score over Gavin_2002 and Gavin_2006 datasets using Kumar et al. localization data



(b) Co-localization score over Krogan_2006 and Tong_2004 datasets using Kumar et al. localization data

Figure 3.8. Co-localization score over four yeast datasets using Kumar et al. localization data.

values of proteins in each complex. Table 3.2 - 3.5 report the p-values of some common enriched terms found by CNCM and other algorithms.

3.5.3 Discussion

CNCM is a topology based technique that ensures high intra-cluster connectivity by adding nodes with higher connectivity to other nodes in the complex. The effectiveness of CNCM was validated for four yeast datasets using statistical and biological measures. The next work is based on the use of a multi-objective optimization to detect protein complexes.

3.6 Finding Protein Complexes using Multi objective Approach : DCRS

Complex finding has been broadly studied as a clustering problem. However, not all methods work well for all datasets. Inefficiency of these methods may be attributed to the presence of different patterns of complexes. For example, not all complexes are dense and some complexes may not be overlapping at all. Therefore, it requires an optimal combination of topological as well as biological properties to identify the complexes. In this section, a method called DCRS has been proposed, which is based on the use of an optimization technique called NSGA II to obtain the optimal set of parameters, which can lead to the formation of biologically significant complexes.

3.6.1 Proposed Method:DCRS

Analyzing protein complex finding as an optimization problem allows us to use a number of topological properties, which are otherwise difficult to regulate. During the course of analysis, we found that centrality measures are not reliable for graph-based problems, i.e., one centrality measure is suitable for one type of network while another measure works well for a different network. So, we used a new measure called reachability contribution in addition to the two other features used by Bandyopadhya et al. The following are the notable features of my method.

- DCRS uses a set of topological and biological features in a parallel fashion for complex extraction.

Table 3.2 Comparison with MCODE, FAG-EC, FT, TFit, OCG, QCUT, ClusterONE and GMFTP in terms of p-value for Gavin_2002 dataset.

GO-ID	**CNCM**	**MCODE**	**GO-ID**	**CNCM**	**FAG-EC**
GO:051123	2.27E-26	1.72E-07	GO:0006378	5.36E-39	1.70E-30
GO:000398	3.32E-24	4.31 E-06	GO:031124	6.86E-35	2.79E-27
GO:000377	3.79E-24	4.52E-06	GO:006379	2.06E-32	4.12E-25
GO:070897	5.23E-25	4.53E-07	GO:070271	2.93E-15	2.89e-10
GO:000375	1.05E-23	6.52E-06	GO:006461	2.93E-15	2.89E-10
GO-ID	**CNCM**	**FT**	**GO-ID**	**CNCM**	**TFit**
GO:031124	6.86E-35	128E-28	GO:070271	2.93E-15	7.68E-09
GO:006399	2.34E-15	2.61E-09	GO:006461	2.93E-15	7.68E-09
GO:006378	5.36E-39	1.96E-33	GO:031124	6.86E-35	1.28E-28
GO:070271	2.93E-15	4.82E-10	GO:OO6399	2.34E-15	3.59E-09
GO:006461	2.93E-15	4.82E-10	GO:006378	5.36E-39	1.96E-33
GO-ID	**CNCM**	**OCG**	**GO-ID**	**CNCM**	**QCUT**
GO:006378	5.36E-39	1.35E-22	GO:071038	3.78E-27	2.86E-18
GO:009304	1.29E-28	5.79E-14	GO:016078	3.78E-27	2.86E-18
GO:042797	1.29E-28	5.79E-14	GO:071042	4.73E-28	2.43E-19
GO:006379	2.06E-32	4.70E-18	GO:071047	4.73E-28	2.43E-19
GO:031124	6.86E-35	5.28E-22	GO:071025	3.05E-25	4.43E-18
GO-ID	**CNCM**	**Cluster ONE**	**GO-ID**	**CNCM**	**GMFTP**
GO:006378	5.36E-39	1.99E-32	GO:031124	1.39E-23	5.66E-14
GO:043631	4.50E-36	3.81E-30	GO:000447	2.13E-21	7.35E-09
GO:043631	4.50E-36	3.81E-30	GO:000447	2.13E-21	7.35E-09
GO:006379	2.06E-32	8.74E-27	GO:000398	1.73E-16	7.15E-08
GO:031124	6.86E-35	1.66E-29	GO:006364	6.39E-10	2.65E-06
GO:006357	4.06E-11	1.87E-06	GO:019941	3.48E-10	2.29E-08

Table 3.3 Comparison with MCODE, FAG-EC, FT, TFit, OCG, QCUT, ClusterONE and GMFTP in terms of p-value for Gavin_2006 dataset.

GO-ID	**CNCM**	**MCODE**	**GO-ID**	**CNCM**	**FAG-EC**
GO:042273	8.77E-41	3.28E-09	GO:042766	6.89E-14	1.62E-05
GO:000462	4.77E-34	5.56E-05	GO:006356	1.61E-15	8.87E-11
GO:030490	2.07E-33	7.90E-05	GO:051329	5.65E-13	2.93E-09
GO:042255	1.79E-21	5.83E-07	GO:051325	9.19 E-13	4.69E-09
GO:000447	3.52E-18	1.98E-04	GO:000349	1.91E-11	8.56E-08
GO-ID	**CNCM**	**FT**	**GO-ID**	**CNCM**	**TFit**
GO:006413	5.11E-18	1.26E-09	GO:006413	5.11E-18	4.10E-12
GO:032986	1.99E-32	3.44 E-25	GO:051325	9.19E-13	5.05E-07
GO:006476	1.42E-23	5.83E-17	GO:051329	5.65E-13	3.07E-07
GO:016479	5.83E-18	8.40E-12	GO:000086	1.43E-12	2.18E-07
GO:031938	1.49E-16	6.06E-11	GO:022411	2.45E-21	1.37E-16
GO-ID	**CNCM**	**OCG**	**GO-ID**	**CNCM**	**QCUT**
GO:006476	1.42E-23	9.44E-15	GO:006476	1.42E-23	8.67E-16
GO:006333	4.79E-32	1.00E-23	GO:006413	5.11E-18	4.70E-11
GO:006413	5.11E-18	7.94E-12	GO:016479	5.83E-18	2.85E-11
GO:006354	1.05E-26	5.73E-22	GO:031938	1.49E-16	3.06E-10
GO:034728	6.33E-32	9.59E-28	GO:032986	1.99E-32	3.32E-26
GO-ID	**CNCM**	**Cluster ONE**	**GO-ID**	**CNCM**	**GMFTP**
GO:006356	1.61E-15	2.86E-12	GO:000398	7.81E-78	3.86E-56
GO:000349	1.91E-11	9.93E-09	GO:042254	1.34E-58	4.01E-50
GO:006376	5.44E-09	1.14E-06	GO:043044	1.61E-50	2.53E-36
GO:000390	3.85E-07	4.71E-05	GO:006378	2.47E-48	6.76E-31
GO:032988	3.85E-07	4.71E-05	GO:042273	8.19E-19	6.69E-19

Table 3.4 Comparison with MCODE, FAG-EC, FT, TFit, OCG, QCUT, ClusterONE and GMFTP in terms of p-value for Krogan_2006 dataset.

GO-ID	**CNCM**	**MCODE**	**GO-ID**	**CNCM**	**FAG-EC**
GO:034661	1.37E-36	1.05E-16	GO:006351	2.93E-31	1.55E-45
GO:016075	1.37E-36	1.05E-16	GO:032774	3.55E-31	2.19E-45
GO:071029	3.72E-35	1.22E-15	GO:032986	8.93E-40	1.00E-31
GO:043633	3.72E-35	1.22E-15	GO:034661	1.37E-36	9.30E-34
GO:071046	3.72E-35	1.22E-15	GO:016075	1.37E-36	9.30E-34
GO-ID	**CNCM**	**FT**	**GO-ID**	**CNCM**	**TFit**
GO:006508	2.46E-28	6.12E-14	GO:034661	1.37E-36	9.36E-21
GO:034661	1.37E-36	3.03E-22	GO:16075	1.37E-36	9.36E-21
GO:016075	1.37E-36	3.03E-22	GO:031123	3.93E-28	7.49E-14
GO:031124	1.51E-26	1.94E-13	GO:071029	3.72E-35	3.53E-21
GO:006353	6.68E-30	5.78E-17	GO:043633	3.72E-35	3.53E-21
GO-ID	**CNCM**	**OCG**	**GO-ID**	**CNCM**	**QCUT**
GO:006334	7.98E-25	5.64E-15	GO:006366	4.39E-34	2.80E-16
GO:000459	1.18E-34	6.32E-25	GO:034661	1.37E-36	1.10E-19
GO:031125	2.16E-33	8.04E-24	GO:016075	1.37E-36	1.10E-19
GO:031497	1.33E-23	3.76E-14	GO:031123	3.93E-28	9.43E-13
GO:000469	3.58E-26	2.41E-17	GO:071029	3.72E-35	3.46E-20
GO-ID	**CNCM**	**Cluster ONE**	**GO-ID**	**CNCM**	**GMFTP**
GO:051568	2.73E-23	1.58E-02	GO:000398	1.78E-46	4.80E-39
GO:043486	2.92E-18	2.10E-04	GO:006351	1.91E-44	5.08E-38
GO:007059	3.75E-11	5.59E-04	GO:043044	3.00E-40	7.77E-34
GO:006403	5.15E-10	3.55E-03	GO:051123	5.48E-36	3.44E-28
GO:006348	2.27E-10	2.62E-04	GO:034661	5.48E-31	3.47E-22

Table 3.5 Comparison with MCODE, FAG-EC, FT, TFit, OCG, QCUT, ClusterONE and GMFTP in terms of p-value for Tong_2004 dataset.

GO-ID	**CNCM**	**MCODE**	**GO-ID**	**CNCM**	**FAG-EC**
GO:071555	1.02E-31	1.88E-06	GO:032197	1.57E-34	2.73E-04
GO:045229	1.02E-31	1.88E-06	GO:000375	2.08E-47	2.18E-28
GO:007067	2.89E-34	3.26E-09	GO:000377	2.32E-48	2.36E-29
GO:000087	3.92E-34	3.70E-09	GO:051656	1.17E-22	1.14E-04
GO:000398	1.74E-48	1.40E-24	GO:006397	1.06E-39	6.02E-22
GO-ID	**CNCM**	**FT**	**GO-ID**	**CNCM**	**TFit**
GO:007062	4.24E-22	2.11E-10	GO:000398	1.74E-48	2.25E-34
GO:000375	2.08E-47	4.87E-36	GO:000377	2.32E-48	2.92E-34
GO:000377	2.32E-48	5.06E-37	GO:000375	2.08E-47	2.12E-33
GO:000398	1.74E-48	3.76E-37	GO:008380	3.02E-44	1.53E-30
GO:007059	4.21E-29	2.82E-18	GO:006397	1.06E-39	1.93E-26
GO-ID	**CNCM**	**OCG**	**GO-ID**	**CNCM**	**QCUT**
GO:000398	1.74E-48	2.64E-35	-	-	-
GO:000377	2.32E-48	3.23E-35	-	-	-
GO:000375	2.08E-47	1.51E-34	-	-	-
GO:008380	3.02E-44	2.60E-32	-	-	-
GO:006397	1.06E-39	4.60E-29	-	-	-
GO-ID	**CNCM**	**Cluster ONE**	**GO-ID**	**CNCM**	**GMFTP**
GO:000087	3.92E-34	1.43E-20	GO:051248	1.56E-37	4.62E-33
GO:007067	2.89E-34	1.05E-20	GO:070882	6.23E-36	4.14E-17
GO:007059	4.21E-29	3.06E-16	GO:006351	1.38E-30	2.39E-22
GO:000280	1.28E-33	2.07E-22	GO:006260	2.13E-25	1.31E-24
GO:009892	6.07E-19	1.04E-09	GO:045229	6.17E-15	9.57E-12

*No significant clustering results were found when we use QCUT algorithm on tong_2004 dataset, so we could not perform the validations on them. Hence the entries for this field in Table 3.5 were NIL.

- It is based on the elitism property of NSGA II which guarantees its improved performance.
- The performance of DCRS is dependent on the initial population set and the number of iterations performed.

DCRS uses a set of objectives to find quality complexes. Handling multiple objectives in parallel gives rise to a number of solutions, generally known as the pareto optimal solutions. One cannot make a comparison among these solutions as making it better for one objective makes it worse for another objective. In order to get the best set of solutions with diversity, we use an optimization technique called NSGA II. The steps involved here are the same as those of any multi-objective optimization problem. These are - Population initialization, choosing the objective functions, intermediate population generation and the final population generation. These steps are discussed in detail next.

A. Population initialization: The process begins with declaring a set of chromosomes as the initial population set. A protein complex is a set, $C_i = \{v_1, v_2, ...v_n\}$, where $v_i \in V$ is a chromosome. The initial population can be derived using any clustering technique on the PPI dataset. We use complexes from the first method, CNCM as the initial population set. The technique used in this method is discussed in Subsection 3.5.

B. Choosing the set of objective functions: The ultimate target of any complex finding technique is to get biologically enriched complexes. In line with this aim, we use both topological and biological properties of the PPI network. From the perspective of topology, *density, contribution of a node into a cluster* and *reachability contribution* are used while from the biological perspective, *semantic similarity* between nodes in the network is used. The overall framework is similar to that of Bandyopadhya et al.'s work [5]. The first two features–density and contribution of a node in a cluster are already known to be useful for complex finding in PPI networks. An additional feature called reachability contribution is used for better topological representation of the PPI network. Reachability contribution is the aggregate sum of the reachability indices of nodes in a cluster. The literature [41] suggests using degrees of adjacent nodes to determine essential proteins in yeast. We therefore use this notion in the form of reachability index to improve the complex finding process.

1. ***Objective functions based on topology:*** The physical interactome of the PPI network defines its inherent properties. The literature [4] suggests the

presence of a dense structure for protein complex formation. We therefore use density as the first objective function. In order to obtain a compact and functionally coherent structure, density of such subgroups needs to be maximized. For a complex C_i, density is given by Definition 21.

The second objective function is calculated as the aggregate contribution $contbn_{v_i}$ of all nodes within a cluster. Maximizing the contribution of each cluster results in the formation of well-separated clusters. The overall contribution of a cluster, $Clcontbn$ is given as the sum of the individual contribution, $contbn_{v_i}$ of its nodes, i.e.,

$$Clcontbn_{C_i} = \sum_{i=1}^{k} contbn_{v_i} \tag{3.3}$$

where $C_i = \{v_1, v_2 ... v_k\}$ is the cluster with members, v_i and the individual node contribution is

$$contbn_{v_i} = \frac{DN_{c_{v_i}}}{d_{v_i}} \tag{3.4}$$

where $DN_{c_{v_i}}$ represents the direct neighbors of v_i within cluster C_i and d_{v_i} is the total degree of node v_i.

The third objective function relies on the reachability of the direct neighbors of a node within a cluster. This function when maximized leads to the formation of coherent complexes. The reachability contribution of a cluster, C_i is given as the sum of the reachability of its member nodes, $Rbty_{v_i}$, i.e.,

$$RbyContbn(C_i) = \sum_{i=1}^{k} Rbty_{v_i} \tag{3.5}$$

where $Rbty_{v_i}$ is the reachability of a node v_i in a cluster C_i and is given as the ratio of the total number of links its direct neighbors have within C_i to the total number of edges in the cluster. Mathematically,

$$Rbty_{v_i} = \sum_{DN_{c_{v_i}}} \frac{l_{wC}}{tedges_C} \tag{3.6}$$

where $DN_{c_{v_i}}$ is the set of direct neighbors of node v_i within cluster C_i, l_{wC} is the number of links each node $v_x \in DN$ has within the cluster, where $DN = \sum_{i=1}^{x} DN_{c_{v_i}}$ and $tedges_C$ is the total number of edges in the cluster, C_i.

2. ***Objective function based on biological characterisitics:*** In addition to the three topological functions, we use a biological measure called semantic similarity for complex finding. The use of a biological feature enhances the

chance of obtaining more biologically relevant complexes. We use Wang's semantic similarity [151]. The overall semantic similarity of a cluster, C_i, is given as the sum of the semantic similarity of the combination of all its member nodes, $C_i = \{v_1, v_2...v_k\}$. Mathematically,

$$SmSim_{C_i} = \sum_{i=1, j \neq i}^{k} semsim(v_i, v_j) \tag{3.7}$$

, where $semsim(v_i, v_j)$ is given by Definition 12.

The semantic similarity criterion is maximized in addition to the other three topological objectives for a cluster so as to get functionally enriched clusters corresponding to biologically relevant complexes. The method is called as *DCRS* as it uses **D**ensity, **C**ontribution, **R**eachability contribution and **S**emantic similarity as its objective functions.

C. Intermediate Population Generation: The next step is the generation of an intermediate population for a user specified number of iterations. This is accomplished using the genetic operators–selection, crossover and mutation. However, we do not use crossover as it produces disconnected components. The selection operation is based on the traditional crowding distance metric and uses the objective function space to prioritize the solutions. Mutation is performed either by adding new nodes or deleting a few nodes. A perturbed node is chosen with probability, $p = 0.9$ for the mutation to take place. For addition of new nodes, a set of random nodes is chosen around the perturbed node, then their direct neighbors are added to produce a new chromosome. For deletion, we remove randomly selected nodes to get a new chromosome. During this step, there are chances of generating ambiguous populations, which is taken care of by using the non-domination sorting method.

D. Final population generation: The final set of chromosomes is returned to the user once the number of iterations are over. The result set is also arranged using non-domination sorting method.

Computational Complexity In a given network consisting of n nodes, let the initial number of complexes generated be M'. Let n' be the number of members in each complex. Therefore, for calculating density, i.e., the first objective function, it requires $O(n')$ time. The calculation of the other two topological function requires $O(M'n'^2)$ each since both of them needs to traverse the whole n' members for each complex to identify the number of interacting partners. In order to get the semantic similarity for n' elements in each complex, it takes $O(2n'^2 - n')$ since $2n'^2 - n'$

combinations of elements need to be analyzed. Once the objective functions are done, the role of NSGA II comes, which is a three step process- non dominated sorting requires $O(M'2N_i^2)$ since it involves M' individual sortings of N_i initial set of populations. The next step is the calculation of crowding distance which takes $O(M(2N_i)log2N_i)$. The final step is sorting the candidates in the front, which requires $O(2N_ilog(2N_i))$. This process is repeated for user defined number of iterations. Thus, the overall time complexity is given by $O(n') + O(M'n'^2) + O(2n'^2 - n') + O(M'2N_i^2) + O(M(2N_i)log2N_i) + O(2N_ilog(2N_i)) \equiv O(n'^2)$.

3.6.2 Experimental Results

The DCRS method is implemented in MATLAB running on an HP xw6600 workstation. Analysis of the proposed method was carried out on the HPRD dataset [111] consisting of 39,237 interactions and 9088 proteins. Wang's semantic similarity used in the objective function is computed using R's GOSemSim package [168], whose details have been discussed in Subsection 2.1.5 of Chapter 2. For the initial population, we use the results obtained using the first method, CNCM and the number of iterations are fixed at 5. Another constraint is that we use only the top 50 complexes from CNCM as the initial population set. The performance analysis was performed using Sensitivity, Positive Predictive Value and Accuracy. The performance of DCRS is compared to a few well-known methods such as MCODE [4], FAG-EC [79], FT [33], TFit [33], OCG [11], QCUT [121], ClusterONE [101], GMFTP [171], CNCM [130] and Bandyopadhya et al.[5] in Figure 3.9.

In Figure 3.9, we observe that DCRS takes the top rank in terms of sensitivity, however, it is beaten by Bandyopadhya's method and ClusterONE in terms of PPV. In order to draw the final conclusion, Accuracy of DCRS with other methods is compared. It is found that DCRS is the clear winner among all methods.

Effect of initial population on the clustering results DCRS was also analyzed by varying the initial population set. This analysis reports the performance of DCRS using a proper complex set obtained from any complex finding method w.r.t. any random set obtained using any clustering method. The performance of DCRS using k-means and CNCM as the initial population set on the HPRD dataset is given in Figure 3.10. In this figure, we see that DCRS gives better performance when using clustering results from CNCM as compared to that of k-means. Moreover, k-means is not suitable in this domain as it requires the number of clusters as input, which is difficult to decide. The performance improvement of our method using CNCM as the initial population set can be attributed to the fact

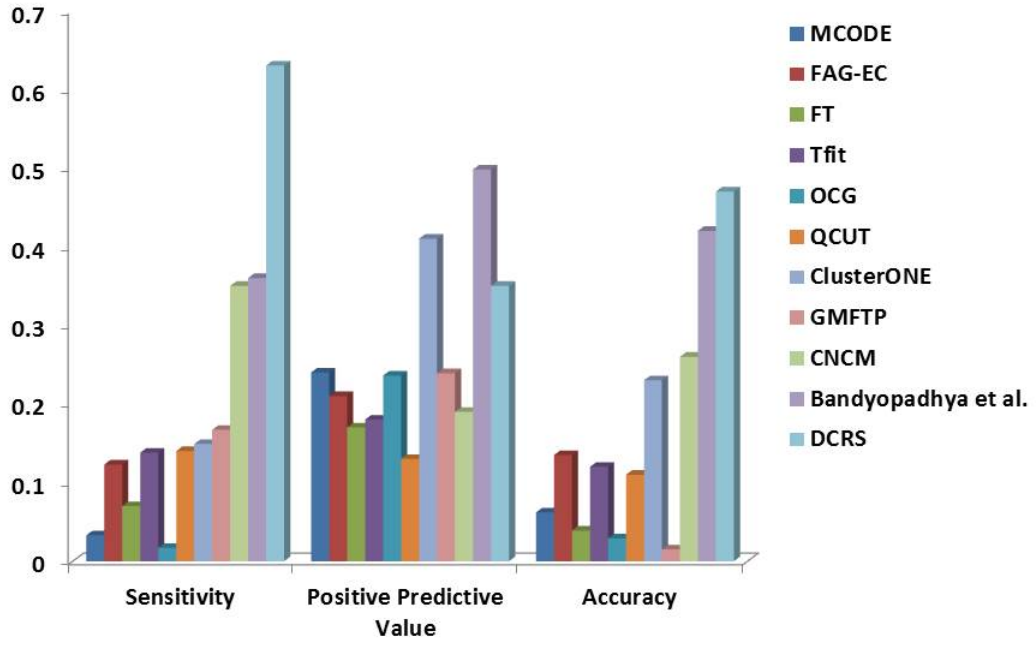


Figure 3.9. Performance measures in terms of Sn, PPV and Acc of DCRS compared with other methods over HPRD dataset.

that k-means is a partitoning approach whereas CNCM is a graph-based method. Moreover, graph-based methods such as MCODE and IPCA have already been shown to be effective in PPI analysis [131].

3.6.3 Discussion

This section introduced a complex finding approach based on the parallel evaluation and optimization of a set of features. We observe that by careful selection of the initial population and the number of iterations, one can obtain high quality complexes. This method performs well in terms of accuracy using ten different methods, which are either sequential or parallel, when tested on the HPRD dataset. This work can be further extended to see its performance at different iterations.

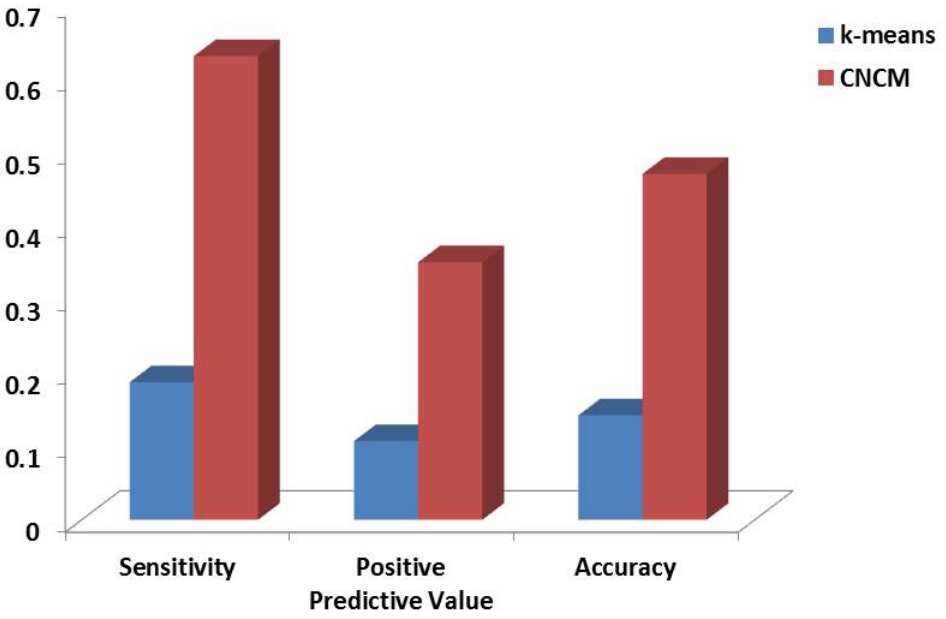


Figure 3.10. Performance measures of DCRS compared with other methods over HPRD dataset and variations in results using two different set of initial population.

3.7 Conclusion

We explore a number of possibilities to identify quality complexes from a PPI Network. We proposed a method called CNCM, which completely relies on topological characteristics of the PPI network. The number of parameters used in this method was limited to two as compared to methods such as MCODE and ClusterONE that depend heavily on five or six parameters each. CNCM is able to detect overlapping as well as sparse complexes, the two major issues in any complex finding technique. We also analyzed the parallel evaluation of the objective functions involved during complex finding and developed a method called DCRS. This method used NSGA II as the optimization algorithm to find an optimum solution set. The performance of this method is better than other methods for the human PPI dataset. The results of the proposed method are reported for the best threshold. Using this threshold, the maximum set of complexes obtained are analysed. With variations in threshold, other combinations of complexes would be found which will need further analysis. The contribution made in this chapter is shown in the form of publication as listed

down in Publication No. 1 & 2.

To study the association between disease and protein complexes, the prime requirement is to have a biologically enriched set of complexes. Thus, functional information along with topological properties of PPI data can be used to achieve the desired objective. The next chapter discusses two such works, which are based on this concept.