Core and peripheral connectivity based cluster analysis over PPI network

Hasin A. Ahmed a, Dhruba K. Bhattacharyya b,∗, Jugal K. Kalita b

a Tezpur University, Assam, India
b University of Colorado, Colorado Springs, USA

A R T I C L E   I N F O

Article history:
Received 7 April 2015
Received in revised form 31 July 2015
Accepted 18 August 2015
Available online 24 August 2015

Keywords:
Protein–protein interaction network
Biological network
Clustering
Protein complex

A B S T R A C T

A number of methods have been proposed in the literature of protein–protein interaction (PPI) network analysis for detection of clusters in the network. Clusters are identified by these methods using various graph theoretic criteria. Most of these methods have been found time consuming due to involvement of preprocessing and post processing tasks. In addition, they do not achieve high precision and recall consistently and simultaneously. Moreover, the existing methods do not employ the idea of core-periphery structural pattern of protein complexes effectively to extract clusters. In this paper, we introduce a clustering method named CPCA based on a recent observation by researchers that a protein complex in a PPI network is arranged as a relatively dense core region and additional proteins weakly connected to the core. CPCA uses two connectivity criterion functions to identify core and peripheral regions of the cluster. To locate initial node of a cluster we introduce a measure called DNQ (Degree based Neighborhood Qualification) index that evaluates tendency of the node to be part of a cluster. CPCA performs well when compared with well-known counterparts. Along with protein complex gold standards, a co-localization dataset has also been used for validation of the results.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

A protein, which is a very important biomolecule with one or multiple chains of amino acid residues, performs a large number of biological functions in a living organism (Nelson et al., 2008). A protein complex, also called a multi protein complex, is a group of protein molecules, i.e., polypeptide chains attached through non-covalent protein–protein interactions. A protein complex is responsible for a single or multiple biological functions in biological processes (Hartwell et al., 1999).

Proteins bind with each other as a result of biochemical phenomena. There are a number of methods to detect the amount of interaction that exists between a pair of proteins including yeast two-hybrid screening and affinity purification coupled with mass spectrometry (Phizicky and Fields, 1995). Yeast two-hybrid screening also known as the Y2H method, detects interaction between a pair of proteins by attaching a protein to the DNA binding domain of a yeast transcription factor such as Gal4 and attaching the other protein to the activation domain of the transcription factor (Von Mering et al., 2002). The protein attached to the DNA binding domain is called the bait protein and the protein attached to the activation domain is called prey protein. Interaction between the proteins brings the binding domain and the activation domain closer, leading to expression of the associated gene. The presence of the gene product implies interaction between the pair of proteins. A second method extracts protein complexes with a target protein from a sample using affinity purification (Von Mering et al., 2002). Finally, mass spectrometry is used to identify the proteins under study and measure the amount of interaction. A protein–protein interaction network (PPIN) is a network or graph of nodes corresponding to proteins, with edges among the nodes corresponding to interaction among the proteins (Rual et al., 2005). A formal definition of PPIN is presented next.

Definition 1. Protein–protein interaction network: A PPI network is defined as a graph $G = (V, E)$ where $V$ is the set of vertices corresponding to proteins and $E$ is the set of edges. An edge is an unordered pair of vertices, $(v_i, v_j)$ where $v_i, v_j \in V$, representing an interaction between the proteins corresponding to the vertices $v_i$ and $v_j$.

There are a number of public databases that collect published information about protein–protein interactions and prepare curated datasets (Lehne and Schlitt, 2009). Some such widely used databases are Biomolecular Interaction Network Database (BIND) (Bader et al., 2003), Munich Information Center for Protein Sequences, and Database of Interacting Proteins (DIP). A number of other databases have also been used for validation of the results.

The dataset has also been used for validation of the results.
Protein Sequences (MIPS) database (Mewes et al., 2004), BioGRID (Stark et al., 2006), Database of Interacting Proteins (DIP) (Xenarios et al., 2000), IntAct (Hermjakob et al., 2004), Molecular INTeraction database (MINT) (Zanconi et al., 2002), Human Protein Reference Database (HPRD) (Peri et al., 2003), YPD (Costanzo et al., 2001) and Funcoup (Alexeyenko et al., 2012). Some of the widely used sources of protein–protein interaction data for clustering are datasets of Krogan et al. (2006), Gavin et al. (2002), Collins et al. (2007), Ho et al. (2002), Ito et al. (2001), Uetz et al. (2000), Giot et al. (2003) and Li et al. (2004).

Along with numerous PPI network repositories, there are a few databases which store record of protein complexes such as MIPS (Mewes et al., 2004), SGD (Hong et al., 2008) and CYC2008 (Pu et al., 2009). Computer scientists have always been assisting biologists by developing mathematical models that act behind biological phenomena. The current state of PPI and protein complex research in terms of experimental findings and well-managed repositories of such findings present the computer scientist with a problem that seeks to determine mathematical relationships that exist among proteins of the same complex in a PPI network. Formally, this problem can be defined as follows.

**Problem Definition:** Suppose a set of protein complexes is represented by $P$, where each complex $P_i \in P$ is a set of proteins. The problem involves designing an algorithm that accepts a PPI network $G = (V, E)$ and produces a set of clusters $C$ where each cluster $C_i \subseteq C$ is a subset of $V$ i.e., $C_i \subseteq V$ using some graph theoretic criteria applied on $G$ such that similarity between $P$ and $C$ is maximized.

Existing methods of clustering PPI network are reported in Section 2. Section 3 presents motivation behind the work. Section 4 presents the proposed method. Experimental evaluation of the method is presented in Section 5 and finally, Section 6 concludes the paper.

### 2. Related work

A number of techniques have been proposed to extract clusters in a PPI network. COACH (Wu et al., 2009) exploits the core-periphery structure of protein complexes to build clusters. It prepares protein complex cores from neighborhood graphs of the vertices. These cores are further extended to form complete clusters. COACH analyses the neighborhood of each vertex to form cluster cores using the concepts of neighborhood graph and core vertices. Along with the process of core creation, it ensures that density of the core is more than a user defined threshold and the cores are not highly overlapped. In the second phase, more vertices are included in the cores if connectivity of a node is found to be more than 0.5. This connectivity is measured as the ratio of number of edges between the node and the core complex to the total number of nodes in the core. RFC (Wu et al., 2014) prepares a weighted network using information about neighborhood of the nodes. This network is transformed into an unweighted form using fuzzy analysis. Then rough set theory is used to prepare lower and upper approximations of the clusters with the property that proteins in the boundary region may be non-exclusive and these may belong to more than one clusters. Finally, clusters with high overlap are merged.

CORE (Leung et al., 2009) uses the core-periphery structure of protein complexes to extract clusters from a PPI network. It uses probability based $P$-values to determine whether a pair of vertices belongs to core region of a cluster based on direct connectivity and their common neighborhood. They progressively merge protein pairs with the lowest $P$-value into core regions of clusters. Once core clusters are formed, the peripheral region is detected by including proteins that are common neighbors of at least half of the core proteins. MCODE (Bader and Hogue, 2003) detects clusters in a PPI network in three stages: vertex weighting, complex detection and post processing of clusters. In the first stage, vertices are assigned weights using the density of a highest k-core subgraph in their neighborhood. In the second stage, to form a cluster, it starts with the node with the highest weight and extends it by including vertices into the cluster with weight more than a user given threshold. In the post processing stage, clusters without two cores are removed.

Cfinder (Adamcsek et al., 2006) is a tool to cluster nodes of a biological network. It uses the concept of $k$-clique to locate clusters. Two $k$-cliques are considered adjacent if they share $k-1$ nodes. A pair of nodes are put in the same cluster if they are reachable from each other through adjacent $k$-cliques. CMC (Liu et al., 2009) uses an iterative scoring method to assign weights to edges. The weight between a pair of nodes is calculated based on their common neighbors. The method then finds maximal cliques, which are finally removed or merged based on their connectivity. MCL (Dongen, 2000) is a flow analysis technique used to cluster PPI networks. The technique uses random walks to find regions with strong flow in the graph. The technique converges towards a set of regions with high flow that correspond to the clusters.

RNSC (King et al., 2004) partitions vertices in the PPI network using a cost function. It moves the vertices randomly among the clusters to optimize the cost function. Post processing of the partitions is carried out using three criteria: cluster size, density and functional homogeneity. MGClus (Friggs et al., 2013) uses an agglomerative approach to combine clusters starting from singleton clusters consisting of individual proteins. To merge clusters they use a measure named Merge Gain which takes the number of interior and exterior edges into account. MGClus merges two clusters if Merge Gain for the pair of clusters is more than a user defined threshold. GMFTP (Zhang et al., 2014) uses an affinity based approach that supports overlap among clusters. Affinity score of a protein against clusters is computed using both functional and topological similarity. Gene Ontology is used to compute functional similarity while protein interaction network is used to compute topological similarity. DPC (Li et al., 2014) detects dynamic protein complexes using PPI data and gene expression data. Time series gene expression data is used to find proteins which are active over time points to add dynamic characteristic of generated clusters. First, DPC detects highly connected core regions assisted by time series gene expression data. These cores are then extended to complete clusters.

A detailed review of techniques for detecting complexes in PPI networks can be found in a recent survey by Ji et al. (2014). Table 1 displays some widely known techniques for complex detection along with their distinguishing features. An issue that arises at the time of validation of these techniques is how many common proteins a cluster must have to match a protein complex. This matching information is then used to compute evaluation measures such as precision, recall and sensitivity. Most techniques are found to adapt an overlap score used by Bader and Hogue (2003). If $\alpha$ is the set of proteins in a cluster $\alpha$ and $\beta$ is the set of proteins in a complex $\beta$, the overlap score between $\alpha$ and $\beta$ is computed as follows:

$$\text{Overlap Score}(\alpha, \beta) = \frac{|\alpha \cap \beta|^2}{|\alpha| \times |\beta|}$$  \hspace{1cm} (1)

If this overlap score is more than a threshold, the cluster is assumed to match the complex. Some work in the literature uses other schemes for comparison such as the Jaccard index. The Jaccard index between a cluster $\alpha$ and a complex $\beta$ is computed as follows:

$$\text{Jaccard Index}(\alpha, \beta) = \frac{|\alpha \cap \beta|}{|\alpha \cup \beta|}$$  \hspace{1cm} (2)
Table 1
Some well-known PPI clustering techniques.

<table>
<thead>
<tr>
<th>Technique, year</th>
<th>Basic working</th>
<th>#Input parameters</th>
<th>Pre process data</th>
<th>Post process result</th>
<th>Datasets used</th>
<th>Gold standards matching scheme</th>
<th>Threshold</th>
<th>Performance measures</th>
<th>Executable given</th>
</tr>
</thead>
<tbody>
<tr>
<td>COACH (Wu et al., 2009), 2009</td>
<td>Detects core region and then extends these to complete clusters</td>
<td>2</td>
<td>✓</td>
<td>× (post processing of cores is embedded)</td>
<td>DIP, Krogan</td>
<td>Friedel</td>
<td>Bader</td>
<td>0.2</td>
<td>p value, Colocalization score, F measure, Coverage rate</td>
</tr>
<tr>
<td>RFC (Wu et al., 2014), 2014</td>
<td>Uses a fuzzy-rough approach for clustering proteins</td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>Gavin, Collins, Krogan, BioGRID</td>
<td>MIPS, SGD</td>
<td>Bader</td>
<td>0.25</td>
<td>Precision, F-measure, Accuracy, Separation</td>
</tr>
<tr>
<td>CORE (Leung et al., 2009), 2009</td>
<td>Uses probability approach to detect core and peripheral regions of clusters</td>
<td>0</td>
<td>×</td>
<td>✓</td>
<td>DIP, Krogan, Gavin</td>
<td>MIPS</td>
<td>Bader</td>
<td>0.6, 0.7, 0.8</td>
<td>No. of complexes detected</td>
</tr>
<tr>
<td>MCODE (Bader and Hogue, 2003), 2003</td>
<td>Combines neighboring weighted vertices based on a threshold</td>
<td>1</td>
<td>× (embedded)</td>
<td>× (embedded)</td>
<td>Gavin, MIPS, YPD</td>
<td>Gavin, MIPS</td>
<td>Bader</td>
<td>0.2</td>
<td>No. of complexes detected, Sensitivity, Specificity</td>
</tr>
<tr>
<td>CFinder (Adamcsek et al., 2006), 2006</td>
<td>Tool to cluster biological networks using k-clique</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>CMC (Liu et al., 2009), 2009</td>
<td>Finds maximal cliques as when finding clusters in a weighted network</td>
<td>2</td>
<td>✓</td>
<td>×</td>
<td>Ho, Gavin, Krogan, Uetz, Ito</td>
<td>MIPS, Aloy</td>
<td>Jaccard index</td>
<td>0.5</td>
<td>Precision, Recall, Colocalization score</td>
</tr>
<tr>
<td>MCL (Dongen, 2000), 2000</td>
<td>Uses random walk to find regions in the graph with high flow</td>
<td>1</td>
<td>×</td>
<td>×</td>
<td>Uetz, Ito, Gavin, Ho, Krogan</td>
<td>MIPS</td>
<td>N/A</td>
<td>N/A</td>
<td>Sensitivity, PPV, Accuracy (Different from above)</td>
</tr>
<tr>
<td>RNSC (King et al., 2004), 2004</td>
<td>Randomly transfers proteins among clusters to optimize cost function</td>
<td>0 (3 in post processing)</td>
<td>×</td>
<td>✓</td>
<td>Przulj, Giot, Li</td>
<td>MIPS</td>
<td>King</td>
<td>N/A</td>
<td>Matching rate (upon request)</td>
</tr>
<tr>
<td>MGClus (Frings et al., 2013), 2013</td>
<td>Clusters are merged in an agglomerative manner to form final clusters</td>
<td>1</td>
<td>✓</td>
<td>×</td>
<td>BioGRID, FunCoup</td>
<td>CYC2008</td>
<td>N/A</td>
<td>N/A</td>
<td>Average Jaccard Index</td>
</tr>
</tbody>
</table>

* Original work was not meant to be applied on PPI clustering. So data presented are based on the review by Brohee and van Helden (2006).
Various matching schemes along with the thresholds used are reported in the columns named Matching scheme and Threshold, respectively, in Table 1.

3. Motivation

Many methods have been proposed in the literature addressing the problem of clustering PPI network. These techniques mainly differ in the criteria used to extract clusters from the PPI network. The following are some structural characteristics of overlapped protein complexes in PPI networks as observed by researchers.

- Protein complexes in a PPI network correspond to relatively dense regions (Tong et al., 2002; Bader and Hogue, 2003; Spirin and Mirny, 2003; Vanunu et al., 2010).
- A protein complex typically has two regions, viz., core and peripheral (Gavin et al., 2006; Wu et al., 2009; Wang et al., 2010; Luo et al., 2009). The core part is a highly dense central region where proteins are strongly connected with each other and the peripheral region is that part of the complex where proteins are weakly connected with the core.

Most existing methods for PPI clustering do not use the second fact in the detection of clusters. Wu et al. (2009) proposed a method named COACH that leverages this structural property of protein complexes in clustering PPI nodes. The core vertex (as defined in Wu et al. (2009)) dominates the density value of the core region of the cluster being detected and as a result, other elements may be left weakly connected when density of the core subgraph is compared with the density threshold. Another issue in COACH is the density of the vertex to reside at a dense region. Formal definitions are presented next.

4. Proposed method

Though a number of clustering methods have been proposed for PPI network analysis, most are not consistent in generating high precision and recall when tested with existing PPI datasets. Another issue is the computational expense imposed by these methods. We propose a method for detection of clusters in PPI network using a graph density approach. Our approach called CPCA1 is influenced by the recent observation that a complex consists of a core dense region with some proteins weakly connected to the dense region, often called periphery (Della Rossa et al., 2013; Wu et al., 2009; Wang et al., 2010). While extracting a cluster, CPCA detects a core dense region, followed by detection of a weakly connected peripheral region.

CPCA uses two connectivity criterion functions to identify core and peripheral regions of the clusters viz., core connectivity and peripheral connectivity. To locate a dense vertex in the PPI network, we introduce a measure named DNQ index that computes the tendency of the vertex to reside at a dense region. Formal definitions of DNQ index and the connectivity measures are presented next.

Definition 2. DNQ (Degree based Neighborhood Qualification) index: DNQ index of a vertex \(v_i\) in a PPI network \(G\) is defined as follows.

\[
DNQindex(v_i) = \frac{|V|}{|neighbors(v_i)|}
\]

Here, \(neighbors(v_i)\) is the set of neighbors of vertex \(v_i\) in \(G\), and \(V = \{v : v \in neighbors(v_i), degree(v) \geq degree(v_i)\}\), where \(degree(v)\) is the degree of vertex \(v\) in \(G\).

Definition 3. Core connectivity: In a PPI network \(G = (V, E)\), core connectivity between a vertex \(v_i\) and a partial cluster \(\rho\) where \(\rho \subseteq V - v_i\) is defined as follows.

\[
CoreCon(v_i, \rho) = \frac{\rho}{|\rho|}
\]

Here, \(\rho = \{v : v \in \rho, (v, v_i) \in E\}\).

Definition 4. Peripheral connectivity: In a PPI network \(G = (V, E)\), peripheral connectivity between a vertex \(v_i\) and a partial cluster \(\rho\) where \(\rho \subseteq V - v_i\) is defined as follows.

\[
PerCon(v_i, \rho) = \frac{|\rho'|}{|neighbors(v_i)|}
\]

Here, \(\rho' = \{v : v \in \rho, (v, v_i) \in E\}\), and \(neighbors(v_i)\) is the set of neighboring vertices of \(v_i\) in \(G\).

The intuition behind using core connectivity and peripheral connectivity is guided by the structural difference between the two regions of a protein complex, namely core and periphery or attachment. The core region of a protein complex is denser and the nodes are highly connected among themselves, while the peripheral region of a complex is weakly connected to the core. To reflect this notion in our technique, we use core connectivity to include proteins which are highly connected to the rest of the nodes. This fact is supported by Eq. (4) where core connectivity is computed as the ratio of the number of edges between the node and the partial cluster to the total number of nodes in the partial cluster. To enable the technique to include weakly connected nodes in the cluster, we trace the proportion of neighbors of the node that are connected to the partial cluster as presented in Eq. (5) instead of checking the proportion of nodes in the partial cluster the node is connected to.

Considering the entire set of vertices as the set of remaining nodes, CPCA starts finding a node with the highest DNQ index. This node is put in a partial cluster and additional nodes with highest core connectivity are added to the partial cluster later. Inclusion of the nodes is driven by a user defined threshold called NC (neighborhood and connectivity) threshold \((\tau)\). This phase of partial cluster extension is exclusive, i.e., no node which has already been included in a cluster is chosen again for inclusion. We call this phase of cluster detection exclusive core formation. Once this exclusive core is formed, additional nodes are inserted into the cluster in a non-exclusive manner. This phase of cluster detection is called non-exclusive core formation. The only difference between exclusive core formation and non-exclusive core formation is that the later phase can pick up nodes even from clusters which have already been formed. Finally, the peripheral connectivity measure is used to include additional nodes into the partial cluster. This phase of cluster detection is named non-exclusive periphery formation. Non-exclusive core formation and non-exclusive periphery formation are also driven by the threshold \(\tau\). This cluster detection process iteratively takes place until there is no remaining node with DNQ index greater than or equal to \(\tau\). A logical view of the CPCA algorithm is given in Fig. 1. If no node is included into the partial cluster during non-exclusive core formation phase, CPCA does not move to other two phases and drops the partial cluster. The complexity of the algorithm is \(O(n^2)\). The exclusive core formation phase of the algorithm does not explore next candidate from all the nodes,

---

1 CPCA stands for core and peripheral connectivity based Cluster Analysis.
rather it searches through only the nodes which are not included in any cluster. So amortized complexity of the algorithm is less than $O(n^2)$. Following are two theorems on the CPCA method that ensure proper formation of core and peripheral regions in a cluster.

**Theorem 1.** Core extension phase of CPCA involves addition of nodes which are strongly connected to the partial cluster formed so far with respect to user defined threshold $\tau$.

**Proof:** In the core extension phase, CPCA includes nodes which are connected to the partial module formed so far with core connectivity value more that CC threshold ($\tau$). If we look at Eq. (4), the core connectivity of a node with a partial cluster is defined as the ratio of number of edges between the node and the partial cluster to the total number of nodes in the partial cluster. Driven by the threshold $\tau$, a node without sufficient edges to the partial cluster formed so far is not included in the partial cluster. Hence, a node must be strongly connected subject to the threshold $\tau$ to get included in the partial cluster during the phase of core extension.

**Theorem 2.** Peripheral extension phase in CPCA involves inclusion of nodes which are weakly connected to the core region and detached from rest of the nodes.

**Proof:** In the peripheral extension phase, CPCA decides whether a node can be included in the partial cluster based on its peripheral connectivity with the core. Peripheral connectivity between a node and the core is defined as the ratio of edges between the node and the core to the total number of edges leaving the node. As presented in Eq. (5), the denominator in peripheral connectivity computation is the total number of edges leaving the node. The value of peripheral connectivity does not correspond to how much connectivity the node has with the core, instead it determines how much the node is detached from nodes outside the core, again based on...
the threshold $r$. Nodes included in the peripheral extension phase cannot be strongly connected as the peripheral extension phase is carried out after the core extension phase. Termination of the core extension phase ensures that no node is connected to the core strongly in terms of core connectivity with respect to threshold $r$.

5. Experimental results

We implement CPCA in Matlab and executed it on an HP Z800 workstation with two Intel Xeon 2.40 processors and 12 GB RAM. The executable of CPCA can be found at http://agnigarh.tezu.ernet.in/~dkb/CPCA/. We use two different sources of knowledge to evaluate quality of the clusters, viz., protein complex gold standard datasets (Mewes et al., 2004; Hong et al., 2008) and the protein co-localization dataset (Kumar et al., 2002). We compare results of the proposed technique with five network clustering techniques, namely MCODE (Bader and Hogue, 2003), COACH (Wu et al., 2009), ClusterONE (Nepusz et al., 2012), Affinity propagation clustering (Frey and Dueck, 2007) and SCPS (Nepusz et al., 2010). Implementations given by authors and the ClusterMaker package (Morris et al., 2011) are used for these techniques. We use PPI datasets of Gavin et al. (2002), Ho et al. (2002), Ito et al. (2001), Uetz et al. (2000) and Von Mering et al. (2002) to test the quality of clusters.

5.1. Validation using protein complex gold standard datasets

We use MIPS (Mewes et al., 2004), SGD (Hong et al., 2008) and CYC2008 as (Pu et al., 2009) gold standard for evaluation of clusters. Different matching schemes have been used in protein–protein cluster analysis as reported in Section 1. We first adopt the scheme of Brohee and van Helden (2006) where they use sensitivity, PPV (Positive Predictive Value) and accuracy to evaluate clustering results. Sensitivity of a clustering solution $C$ with clusters $c_1, c_2, \ldots, c_n$ is computed as follows:

$$Sensitivity = \frac{\sum_{c_i \in C} |c_i| \times \text{Sens}_{c_i}}{\sum_{c_i \in C} |c_i|}$$ (6)

Here $\text{Sens}_{c_i}$ is the highest overlap score of cluster $c_i$ with any complex in the gold standard, where overlap score between a cluster and a complex is the ratio of the number of common proteins to the total number of proteins in the complex. Similarly, PPV for clustering solution $C$ can be computed as follows:

$$PPV = \frac{\sum_{c_i \in C} |c_i| \times \text{PPV}_{c_i}}{\sum_{c_i \in C} |c_i|}$$ (7)

Here $\text{PPV}_{c_i}$ is the highest overlap score of cluster $c_i$ with any complex in the gold standard, where the overlap score between a cluster and a protein complex, unlike that of sensitivity computation, is the ratio of the number of common proteins to the total number of proteins in the cluster. Accuracy of a clustering solution is defined as the geometric mean of sensitivity and PPV. The PROCOPE (Krumsieck et al., 2008) tool is used to evaluate effectiveness of produced clusters in terms of accuracy.

We have carried out an analysis of the results produced by CPCA to decide on the value of input parameter Neighborhood and connectivity threshold $r$ on all datasets in terms of accuracy. From Figs. 2–8, we observe that CPCA exhibits good performance for $r = 0.7$ across all datasets. All evaluations reported next are carried out with the value of $r$ set to 0.7.

Fig. 5 shows a comparison of CPCA with its counterparts in terms of accuracy. On Gavin et al.’s dataset, CPCA lies close to ClusterONE, the best performer. ClusterONE seems to be quite close to CPCA and performs well over datasets of Gavin et al., Ho et al. and Mering et al. But it lies behind CPCA on the datasets of Uetz et al. and Ito et al. SCPS is found to counterparts on Uetz et al. and Ito et al.’s datasets. On Mering et al.’s dataset, CPCA lies close to ClusterONE, the best performer. ClusterONE seems to be quite close to CPCA and performs well over datasets of Gavin et al., Ho et al. and Mering et al. But it lies behind CPCA on the datasets of Uetz et al. and Ito et al. SCPS is found to
Fig. 5. Comparison using MIPS gold standard. Program crashed while running SCPS on Ito et al.'s dataset.

Fig. 6. Comparison using SGD gold standard. Program crashed while running SCPS on Ito et al.'s dataset.

Fig. 7. Comparison using CYC2008 gold standard. Program crashed while running SCPS on Ito et al.'s dataset.

Fig. 8. Average accuracy of CPCA and its counterparts on all datasets.

perform well on the datasets of Gavin et al., Uetz et al. and Mering et al. But it fails to achieve relatively good accuracy with the remaining two datasets. COACH is found to achieve good accuracy only on Gavin et al.'s dataset.

A similar scenario is observed when comparing CPCA with its counterparts using SGD as gold standard as presented in Fig. 6. While CPCA beats all on Uetz et al. and Ito et al.'s datasets, it stands close to ClusterONE, the winner on rest of the datasets. ClusterONE is again found to fail to achieve relatively good accuracy on the datasets of Uetz et al. and Ito et al. SCPS is also found to achieve good accuracy on datasets of Gavin et al., Uetz et al. and Mering et al. Thus, CPCA achieves consistently good accuracy on all the datasets.

On CYC2008 dataset, as reported in Fig. 8, CPCA generates best accuracy over all the datasets except Gavin et al.'s dataset. Fig. 8 presents average accuracy of CPCA and its counterparts on all datasets. In terms of average accuracy, CPCA beats all its contenders over all datasets.

We also use overlap score (as reported in Eq. (1)) based matching scheme that requires a threshold value (Bader and Hogue, 2003). Based on whether the overlap score is more than the threshold we say whether a cluster matches a protein complex. Using overlap threshold value of 0.2 (as used by most of the existing methods as reported in Table 1), we compute F-measure for set of clusters generated by CPCA and its counterparts. For a clustering solution C with clusters \( c_1, c_2, \ldots, c_n \) and gold standard \( P \) with complexes \( p_1, p_2, \ldots, p_m \), \( \text{precision} \) can be computed as follows (Wu et al., 2014).

\[
\text{Precision} = \frac{|\{|c_i | c_i \in C, \exists p_j \in P, \text{Overlap} \_\text{score}(c_i, p_j) \geq \alpha\}|}{|C|} \tag{8}
\]

Here, \( \text{Overlap} \_\text{score}(c_i, p_j) \) is the overlap score between cluster \( c_i \) and complex \( p_j \) as reported in Eq. 1. \( \alpha \) is the overlap threshold value. Similarly, \( \text{recall} \) can be computed as follows.

\[
\text{Recall} = \frac{|\{|p_j | p_j \in P, \exists c_j \in C, \text{Overlap} \_\text{score}(p_j, c_j) \geq \alpha\}|}{|P|} \tag{9}
\]

F-measure is harmonic mean of \( \text{precision} \) and \( \text{recall} \) and can be computed as follows.

\[
F \text{- measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}
\]

Fig. 9 reports F-measure generated by CPCA and its counterparts using MIPS gold standard. Except Ho et al.'s dataset, CPCA is found to lie close to the best performer. CPCA generates best F-measure values on Uetz et al.'s, Ito et al.'s and Mering et al.'s datasets while
using SGD as the gold standard as reported in Fig. 10. A similar behavior was observed by CPCA while CYC2008 dataset is used as gold standard as presented in Fig. 11. Fig. 12 presents average F-measure value of CPCA and its counterparts on all datasets. In terms of average F-measure value, CPCA beats all its contenders when CYC2008 and SGD dataset is used, but loses to ClusterONE when MIPS dataset is used as gold standard.

5.2. Validation using co-localization datasets

The protein localization dataset of Kumar et al. (2002) is used to evaluate the quality of clusters produced by CPCA and its counterparts. We use co-localization PPV used by Pu et al. (2007) for evaluating how proteins belonging to the same clusters are co-localized. Co-localization PPV of a cluster $c_i$ in clustering solution $C$ with respect to localization categories $l_1, l_2, \ldots, l_n$ is computed as follows:

$$\text{co-localization PPV}_{c_i} = \frac{\max_j \text{Common}_{c_i, l_j}}{\sum_{k=1}^{n} \text{Common}_{c_i, l_k}}$$  \hspace{1cm} (11)

Here $\text{Common}(c_i, l_j)$ is the number of common proteins between cluster $c_i$ and localization category $l_j$. Co-localization PPV of the entire clustering solution $C$ is computed as the weighted average of these cluster values.

Fig. 13 presents co-localization PPV scores of the clusters produced by CPCA and its counterparts on all datasets. CPCA is found
to achieve the highest co-localization PPV on Gavin et al. and Ho et al.'s datasets. On Uetz et al.'s dataset, CPCA is very close to Affinity Propagation clustering, the best performer. Though CPCA fails to achieve a relatively high co-localization PPV on Ito et al.'s dataset, it stands close to COACH and ClusterONE, the winners. COACH is found to consistently achieve good accuracy over all the datasets. CPCA along with ClusterONE achieves good accuracy over all the datasets except the dataset of Ito et al. Affinity Propagation clustering is also found to achieve good accuracy over datasets of Ho et al., Uetz et al. and Ito et al.

6. Conclusion

We propose a technique for detection of clusters that correspond to complexes in a PPI network. We have formulated two criteria to form densely connected core and weakly connected peripheral regions of clusters. The correspondence between clusters produced from five benchmark PPI datasets and protein complexes is evaluated in terms of accuracy and F-measure using MIPS and SGD protein complex datasets. CPCA has also been evaluated using a co-localization dataset. CPCA performs consistently well across all the datasets being the best or very close to the best in most of the cases.

Acknowledgements

This work is supported by Department of Science and Technology, Government of India through INSPIRE program under AORC (Assured Opportunity for Research Careers) scheme.

Appendix A. Supplementary Data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.compbiolchem.2015.08.008.

References


