A Fast Agglomerative Community Detection Method for Protein Complex Discovery in Protein Interaction Networks

Mohammad S. Rahman and Alioune Ngom

School of Computer Sciences, 5115 Lambton Tower, University of Windsor, 401 Sunset Avenue, Windsor, Ontario, N9B 3P4, Canada
{rahman1v,angom}@uwindsor.ca

Abstract. Proteins are known to interact with each other by forming protein complexes and in order to perform specific biological functions. Many community detection methods have been devised for the discovery of protein complexes in protein interaction networks. One common problem in current agglomerative community detection approaches is that vertices with just one neighbor are often classified as separate clusters, which does not make sense for complex identification. Also, a major limitation of agglomerative techniques is that their computational efficiency do not scale well to large protein interaction networks (PINs). In this paper, we propose a new agglomerative algorithm, FAC-PIN, based on a local premetric of relative vertex-to-vertex clustering value and which addresses the above two issues. Our proposed FAC-PIN method is applied to eight PINs from different species, and the identified complexes are validated using experimentally verified complexes. The preliminary computational results show that FAC-PIN can discover protein complexes from PINs more accurately and faster than the HC-PIN and CNM algorithms, the current state-of-the-art agglomerative approaches to complex prediction.

1 Introduction

Proteins are known to interact with each other by forming complexes. Each such complex performs an independent and discrete biological function through the interactions of its member proteins [9]. Single proteins may also participate in more than one complex. Protein complexes correspond to *modules*, which are dense subgraphs within PINs, and hence, they can be discovered by appropriate graph clustering approaches. Generally speaking, modules in PINs refer to highly connected subgraphs which have more internal edges than external edges. Many definitions of modules have been proposed in literature [16], and consequently different community detection algorithms have been proposed based on these different definitions.

Module detection in PINs is a computationally hard task and conventional clustering algorithms are not well suited for this task [15, 20]. Efficient, accurate, robust, and scalable methods are therefore required for mining large PINs. There are generally three classes of modules detection approaches: 1) those based on finding *cliques*, which are fully connected subnetworks [11, 17]; 2) those based on detecting dense subnetworks [1, 2], not necessarily cliques; and 3) those based on uncovering the hierarchical

A. Ngom et al. (Eds.): PRIB 2013, LNBI 7986, pp. 1–12, 2013.

organization of modules within PINs [8, 12]. Clique techniques are not quite scalable to large PINs and the identified modules are too strict in the biological sense of modules since proteins participating in a complex may not all interact with each other. Current density-based algorithms commonly misclassify proteins with low degree into small clusters which could be merged to core protein clusters [13]. Moreover, many biologically meaningful modules are ignored due to their low topological connectivity [13]. Hierarchical clustering methods based on global metric over nodes or edges, such as betweenness centralities, are very time-consuming, and thus do not scale well to large PINs. The few hierarchical approaches based on local metric also have the common problem of classifying vertices with degree one in separate clusters, which does not make sense biologically.

In this paper, we propose a fast agglomerative clustering technique, FAC-PIN, which addresses the limitations discussed above for hierarchical algorithms. FAC-PIN is based on a local premetric of relative vertex clustering value for clustering PINs in a hierarchical manner.

The rest of the paper is organized as follow. In Section 2, we discuss a few hierarchical algorithms to which FAC-PIN is based. Section 3 introduces our proposed method. Computational experiments and discussions of results are given in Section 5 before we conclude with possible directions of research.

2 Related Works

Many hierarchical clustering approaches (both agglomerative and divisive techniques) have been introduced in literature, since the original publication of Girvan and Newman in [7] for clustering networks. See the excellent survey on graph clustering algorithms in [5]. Thus, we will present only the few methods that are directly related to our proposed agglomerative approach.

An effective agglomerative technique for clustering large networks was first proposed by Girvan and Newman in [7]. The Girvan and Newman (GN) algorithm first computes the edge-betweenness centrality value of each edge; this is a global metric over the edges and is defined as the number of shortest paths containing a given edge. Then, GN subsequently sort and then remove edges with large betweenness values in an iterative manner and in order to detect the communities; since such edges correspond to *bridges* connecting two modules whereas low-betweenness edges are internal to modules. To increase the computational speed of GN, Clauset et al. [4] made a simple but non-trivial modification in the computation of the value of the modularity function used in GN. Luo et al. [13] defined the concept of the degree of a subnetwork S as the number the of edges containing one endpoint inside S and the other endpoint outside S. The degree of subnetworks was used along with the edge-betweenness values to devise an agglomerative method for module discovery. Li et al. [12] developed a fast agglomerative approach for community detection based on a global centrality measure, the *vertex clustering coefficient*; which is defined as the ratio of the number of edges between the neighbors of a given vertex v and the total number of possible edges in that neighborhood, it measures the degree of completeness of the subnetwork defined by v and its neighbors [6]. Radicchi et al. [16] designed an agglomerative technique based

on the clustering coefficient of an edge; the *edge clustering coefficient* extends the vertex clustering coefficient and is a global measure defined as the number of triangles to which a given edge $e = (u, v)$ belongs to, divided by the number of triangles that might potentially include (u, v). That is:

$$C^{(3)}_{u,v} = \frac{Z^{(3)}_{u,v}}{\min\{(k_u - 1), (k_v - 1)\}}, \tag{1}$$

where, k_a is the degree of a vertex a, $Z^{(3)}_{u,v}$ is the number of triangles containing edge (u, v), and $\min\{(k_u - 1), (k_v - 1)\}$ is the maximal possible number of triangles containing (u, v). This coefficient has been further generalized to higher-order cycles, $C^{(k)}_{u,v}$, such as squares for $k = 4$, $C^{(4)}_{u,v}$. Edges contained in few or no triangles have low clustering coefficients, and hence, correspond to *bridges* connecting two clusters. The edge clustering coefficient assumes the existence of cycles of length k in a network; which is problematic since a network can have many cycles of different lengths and the length distribution is unknown (e.g., there may be very few or very many short-length cycles). For this reason, Wang et al. [19] defined a local metric over the edges, the *edge clustering value*, which is not based on cycles but on the common neighbors of the two endpoints of edge (u, v). The edge clustering value is defined as:

$$ECV(u, v) = \frac{|N_u \cap N_v|^2}{|N_u| \times |N_v|}, \tag{2}$$

where, N_a is the set of neighbors of a vertex a and its cardinality is defined as $|N_a|$. Here, endpoints vertices of an edge (u, v) with a larger clustering value are more likely to be in the same cluster. Using the edge clustering value, Wang et al. [19] devised an agglomerative technique, the HC-PIN algorithm, for discovering modules of a PIN and which is faster and more accurate than current hierarchical algorithms for network clustering.

In the following section, we introduce a new measure, the *relative vertex-to-vertex clustering value*, which is a premetric combining the ideas behind the vertex clustering coefficient, the edge clustering coefficient, and the edge clustering value. Our analysis of this measure will be based on the *weak sense* definition of a community (i.e., a module); that is: a subgraph S is a community in a weak sense if the sum of all degrees within S (i.e., sum of its internal edges) is larger than the sum of all degrees toward the rest of the network (i.e., sum of its external edges) [16].

3 Relative Vertex-to-Vertex Clustering Value

The edge clustering value, $ECV(u, v)$, used in HC-PIN [19], is a similarity metric between the two vertices u and v of an edge (u, v) and which, roughly speaking, tells how likely u and v lie in the same module (i.e., cluster). This is also true with the edge clustering coefficient, $C^{(3)}_{u,v}$, of [16]. However, in complex networks following the power law (i.e., scale-free networks), it is reasonable to assume that the likelihood of a vertex u to lie in the same module as v (or, to lie in the module containing v), is not

equal to the likelihood of v to lie in the module containing u. This assumption stems from the principle of *preferential attachment* in scale-free networks which states that a new node u is likely to *attach* to a high-degree node v than to a low degree node. This is not reciprocal, and hence, clearly suggesting that the likelihood is not symmetric and that it is larger for u to be in a cluster with v than for v to be in cluster with u (if we assume that v is a high-degree node). The similarity metrics $ECV(u, v)$ and $C^{(3)}_{u,v}$ treat equally both endpoints of edges (u, v) irrespective of their degrees. Also, another issue is that both $ECV(u, v)$ and $C^{(3)}_{u,v}$ require vertices u and v be connected by an edge. This requirement is quite restrictive and we aim to extend to the case in which pair (u, v) is not an edge while still being able to decide if both vertices are in the same cluster. Finally, as stated earlier in previous section, current hierarchical approaches have the common problem of classifying low-degree vertices (peripheral to dense subnetwork modules) into separate clusters rather than merging them with their neighboring modules. In the following paragraph, we present a new measure which aims to address these issues.

Let N_a be the set of neighbors of vertex a in an undirected graph $G = (V, E)$. We define $N^+_a = N_a \cup \{a\}$ as the neighbor set of a augmented with a itself. Given two vertices u and v, we define the clustering value of u relative to v as:

$$R(u \dashrightarrow v) = \frac{|N^+_u \cap N^+_v|}{|N^+_u|} \tag{3}$$

$R(u \dashrightarrow v)$ is a premetric that ranges from 0 to 1; that is, it is a measure which does not satisfy the axiom of symmetry and the triangle inequality but satisfies the axioms of self-similarity and minimality. A vertex u with a larger clustering value given another vertex v is more likely to lie in the cluster containing v. In the following $C(a)$ denotes the cluster containing a given vertex a, and we assume that $C(a)$ satisfies the *weak sense* definition of a community [16] (we use the term ws-cluster, hereafter). The following describe the properties of $R(u \dashrightarrow v)$.

Given an edge (u, v), $R(u \dashrightarrow v)$ is maximal (i.e. equals 1) if and only if $|N^+_u| = |N^+_u \cap N^+_v|$. There are two cases achieving the maximum given edge (u, v): (i) when u has degree one; and (ii) when both u and v have the same degree and $|N^+_u| = |N^+_v|$ that is, they have the same neighbors. In either case, If sub-network $C(v)$ (respectively, the induced sub-network of G for subset N^+_v) is a ws-cluster then $\{u\} \cup C(v)$ (respectively, $\{u\} \cup N^+_v$) is a also a ws-cluster.

Given an edge (u, v), $R(u \dashrightarrow v)$ is minimal when u is the highest degree vertex in G and v has degree 1; that is, $R(u \dashrightarrow v) = \frac{2}{1+deg(u,G)}$ and $deg(u, G)$ is maximal. In such case, $R(v \dashrightarrow u)$ is maximal (i.e. equals 1), and hence, $C(u) \cup \{v\}$ (respectively, $N^+_u \cup \{v\}$) is a ws-cluster if $C(u)$ (respectively, N^+_u) is a ws-cluster.

Given an edge (u, v), assume the degrees of vertices u and v in G are such that $deg(u, G) = deg(v, G) = d$ is maximal and that u and v do not share any other neighbors. Then, we have $R(u \dashrightarrow v) = R(v \dashrightarrow u) = \frac{2}{1+d} \leq 0.5$ assuming $d \geq 3$. In this case, $\{u\} \cup C(v)$ (or N^+_v) is not a ws-cluster, and, $\{v\} \cup C(u)$ (or N^+_u) is not a ws-cluster. Consider the induced subgraph of G on $N^+_u \cup N^+_v$, we define the *local betweenness value* of edge (u, v) as the percentage of paths from vertices in $N_u \setminus N_v$ to vertices in $N_v \setminus N_u$ going through edge (u, v). Given the number of

common neighbors between u and v, $|N_u \cap N_v|$, the local betweenness of edge (u, v) is thus $l(u, v) = 100 \cdot \frac{1}{|N_u \cap N_v|+1}$. Given two connected high-degree vertices u and v, the local edge betweenness value $l(u, v)$ increases when $|N_u \cap N_v|$ decreases, and hence, it corresponds to when both $R(u \dashrightarrow v)$ and $R(v \dashrightarrow u)$ values are small at the same time. Edges with high local betweenness values are edges connecting two clusters, and therefore, vertices u and v should not lie in the same cluster.

Finally, our relative vertex clustering values implements the ideas behind the edge clustering coefficient, $C^{(k)}_{u,v}$, of [16], since for a given vertex v and a neighbor u the number of triangles given edge (u, v) is exactly $|N_u \cap N_v|$; and u will be included into $C(v)$ whenever most of the neighbors of u (excluding v) are in $N_u \cap N_v$. This is also true even when (u, v) is not an edge; in such case, $|N_u \cap N_v|$ relates to the number of squares containing vertices u and v. On the other hand, we break through the limitations of [16] as in the edge clustering value, $ECV(u, v)$ of [19], by not assuming the existence of closed loops in a networks, such as triangles or high-order loops. The relative vertex clustering value $R(u \dashrightarrow v)$ also improves $ECV(u, v)$ since neighbors u of v which have most of their neighbors forming a triangle with v are selected for inclusion in $C(v)$. Searching for vertices u which form a cluster with v is also more efficient than searching for edges (u, v) that makes a cluster since the number of edges is larger than the number of vertices in dense subgraphs.

In summary, the values $R(u \dashrightarrow v)$ and $R(v \dashrightarrow u)$ for edge (u, v) can be used as a quick test for deciding whether u (respectively, v) should be merged with the cluster $C(v)$ (respectively, $C(u)$) such that $\{u\} \cup C(v)$ (respectively, $\{v\} \cup C(u)$) remains a ws-cluster.

4 The FAC-PIN Algorithm

Our proposed fast agglomerative clustering algorithm for protein interaction networks, FAC-PIN in Algorithm 1, goes as follows. Given a PIN $G = (V, E)$, we initially consider each vertex as a singleton cluster, and sort the vertices $v \in V$ in decreasing order of their degrees $deg(v, G)$ in G. Then, in an iterative manner, we select the next highest-degree vertex v from the sorted list, and compute the values $R(u \dashrightarrow v)$ and $R(v \dashrightarrow u)$ for each neighbor u of v, and then decide depending on these two values and a threshold $\alpha, 0 \leq \alpha \leq 1$, whether u should be included in $C(v)$ or not.

In the FAC-PIN algorithm, a neighbor u of vertex v is added to the current $C(v)$ when the majority of the neighbors of u are in $N_u \cap N_v$, that is when: 1) $R(u \dashrightarrow v) = 1$, in which case either u has degree 1, or u and v have the same degree and the same set of neighbors; 2) $R(u \dashrightarrow v) > R(v \dashrightarrow u) > \alpha$, in which case u have smaller degree than v and most of the neighbors of u are in the intersection; and 3) $R(u \dashrightarrow v) = R(v \dashrightarrow u)$ and the size of the intersection is larger than the total set of neighbors of u and v which are not in the intersection.

Computational Complexity of FAC-PIN: Let $n = |V|$ be the number vertices, $m = |E|$ be the number of edges, and $\bar{d}$ be the average degree of all vertices, that is $\bar{d} = \frac{1}{n}\sum_{v \in V} deg(v, G)$. The complexity of sorting the vertices by their degree is $O(n)$ by using the *counting sort* method, and the complexity of computing the partition after

Algorithm 1. The *FAC-PIN* Algorithm

Input: $G = (V, E)$: undirected PIN graph
α: threshold parameter
Output: $P_k = \{C_1, \ldots, C_k\}$: identified collection of modules
{**Initialization phase**}
for every $v_i \in V$ **do**
$C(v_i) \leftarrow \{ \{v_i\}, \emptyset \}$; {*each vertex is a singleton cluster*}
end for
Sort all vertices to a priority-queue H in non-increasing order of their degrees;
{**Community detection phase**}
repeat
$v \leftarrow H$; {*select next highest-degree vertex in* H}
for all $u \in N_v$ not yet merged into a cluster **do**
if $[R(u \dashrightarrow v) = 1]$ Or $[R(u \dashrightarrow v) > R(v \dashrightarrow u) > \alpha]$ **then**
$C(v) \leftarrow C(v) \cup \{ \{u\}, \{(u, v)\} \}$;
$C(u) \leftarrow C(v)$;
else
if $[R(u \dashrightarrow v) = R(v \dashrightarrow u)]$ And $[deg(u, G) + deg(v, G) - 1 \leq |N_u \cap N_v|]$ **then**
$C(v) \leftarrow C(v) \cup \{ \{u\}, \{(u, v)\} \}$;
$C(u) \leftarrow C(v)$;
end if
end if
end for
until $H = \emptyset$
$U \leftarrow V$;
$i \leftarrow 1$;
{**Compute the partition** P_k}
while $U \neq \emptyset$ **do**
$v \leftarrow$ randomly select a vertex from U;
$C_i \leftarrow C(v)$;
$U \leftarrow U \setminus \{u \mid C(u) = C(v)\}$;
$i \leftarrow i + 1$;
end while
return $P_k \leftarrow \{C_1, \ldots, C_k\}$;
Evaluate modularity $Q(P_k)$ of partition $P_k = \{C_1, \ldots, C_k\}$;

the community detection phase is also $O(n)$. Let the maximum node degree in G be $d_{\max} = \max_{v \in V} deg(v, G)$. The complexity of computing $R(u \dashrightarrow v)$ given vertices u and v in the "for-loop" of FAC-PIN is $O(d_{\max})$. The complexity of the "for-loop" is then $O(d_{\max}^2)$, and hence, the total complexity of the "repeat-loop" (and thus of FAC-PIN) is $O(nd_{\max}^2) \ll O(n^3)$. Since PINs are power-law networks then the majority of the proteins interact with only very few proteins, and thus the average degree $\bar{d}$ is generally small and can be considered a constant [19]; that is, we can use $\bar{d}$ as the principal variable for measuring the complexity of community detection methods. As such, then the complexity of FAC-PIN is $O(n\bar{d}^2) \ll O(nd_{\max}^2) \ll O(n^3)$. The complexity of the HC-PIN algorithm of [19] is $O(m\bar{d}^2)$ and is larger than that of FAC-PIN since

$n \lll m$ in PINs. We note that HC-PIN is currently the fastest hierarchical method described in literature for clustering PINs, as far as we know.

5 Computational Experiments and Discussions

We have carried out several computational experiments on the PIN data of eight different species using our proposed FAC-PIN algorithm. For each PIN, we performed the following steps sequentially: (1) we arbitrarily set the threshold parameter, α in FAC-PIN, to values $0.5, 0.25, 0.125, 0.0625$ and 0.03125, (2) applied FAC-PIN to the given PIN, with each of these values, (3) evaluated the modularity (i.e., the goodness) of the resulting partition P_k for a value α, and finally (4) we reported the partition result for the value α (among all given values) which gives the best modularity value. The PINs and the modularity evaluation functions are discussed below.

PIN Data: The PINs data of eight different species were obtained from the PINALOG site[1] and the BioGRID database[2]. The eight species given along with their number of proteins and interactions in parenthesis are: *E. coli* (2817, 13841), *D. melanogaster* (Fruit fly, 8366, 25611), *A. thaliana* (Flowering plant, 2651, 5236), *M. musculus* (House mouse, 2888, 4372), *H. sapiens* (Human, 8994, 34935), *R. norvegicus* (Street rat, 1148, 1307), *C. elegans* (Round worm, 4303, 7747), and *S. cerevisiae* (Bakers yeast, 5672, 49830). In all these PINs, the number of edges is much larger than the number of vertices.

Modularity Functions: Given a clustering result (i.e. a partition) $P_k = \{C_1, \ldots, C_k\}$ with k clusters, we used the popular modularity function introduced by Newman and Girvan [4], defined as

$$Q(P_k) = \sum_{i=1}^{k} (e_{ii} - a_i^2), \tag{4}$$

where, e_{ii} is the fraction of edges with both end vertices in the same community i, and a_i is the fraction of edges with at least one end vertex in community i. Larger values of Q correspond to more distinct community structures in PINs. Though Q is widely used, it is known to have serious limitations which has been discussed at length in [5]. The second partition scoring function we used has been introduced in [10] and is defined as

$$w\text{-}\log\text{-}v(P_k) = \sum_{i=1}^{k} (e_{ii} - \log a_i). \tag{5}$$

Function $w\text{-}\log\text{-}v$ allows for more diverse cluster sizes than function Q, and smaller values corresponds to better modularity structures.

[1] `http://www.sbg.bio.ic.ac.uk/~pinalog/downloads.html`
[2] `thebiogrid.org`

Computational Results: As said above, we applied FAC-PIN many times on a given PIN data but with a different threshold value α in each run, then evaluated the resulting partition for that value α, and then retained the best partition P_k obtained for the PIN among all values α. The best partition is that which has the best modularity value. In order to study and compare the performance of FAC-PIN, we downloaded the CNM code from `http://cs.unm.edu/~aaron/research/fastmodularity.htm`[4] and implemented the HC-PIN algorithm [19]. The HC-PIN and CNM methods were applied on the same PIN data as the FAC-PIN approach. For HC-PIN, we set the two parameters λ and s as in [19] (CNM has no parameters). The modularity results of the three methods are given in Tables 1 and 2, and their running times are shown in Table 3. The PINs are sorted in increasing order of their number of proteins (that is, *Street rat*'s PIN being the smallest is on first column and *Human*'s PIN being the largest is on the last column).

Table 1. Q results of *FAC-PIN*, *CNM* and *HC-PIN*

Algorithms	Street rat	Flowering plant	E. Coli	House mouse	Round worm	Baker's yeast	Fruit fly	Human
FAC-PIN	0.7897	0.9422	0.1492	0.7644	0.7484	0.5110	0.6486	0.7827
CNM	0.5457	0.7861	0.0587	0.4781	0.4057	0.1412	0.3116	0.2858
HC-PIN	0.4502	0.7819	0.0023	0.5015	0.2928	0.0387	0.0086	0.0126

Table 2. w-$\log$-v results of *FAC-PIN*, *CNM* and *HC-PIN*

Algorithms	Street rat	Flowering plant	E. Coli	House mouse	Round worm	Baker's yeast	Fruit fly	Human
FAC-PIN	-2.252	-3.603	-0.262	-2.634	-2.094	-0.521	-1.517	-1.941
CNM	-1.699	-2.866	-0.192	-1.530	-1.819	-0.481	-1.233	-1.269
HC-PIN	-1.558	-3.071	-0.019	-1.805	-1.809	-0.028	-0.072	-0.113

As we see in both Tables 1 and 2, FAC-PIN outperformed both the HC-PIN and CNM methods in all given PINs. We note that as the size of the PINs increases, in terms of either the number of proteins or the number of interactions, the difference between the performances of FAC-PIN and HC-PIN (or CNM) also increase greatly. This is also true in Table 3 showing the execution times, in seconds, of the three algorithms. Clearly FAC-PIN is much faster than the other two methods, and again, the difference in performance increases as either the number of proteins or the number of interactions increases. All experiments were performed on an Intel machine (Core TM i7-2600, 3.400 GHz, CPU with 8 GB RAM).

Table 3. Time results *FAC-PIN*, *CNM* and *HC-PIN*

Algorithms	Street rat	Flowering plant	E. Coli	House mouse	Round worm	Baker's yeast	Fruit fly	Human
FAC-PIN	1.00	4.77	3.66	7.44	22.25	25.12	54.85	72.59
CNM	8.46	119.40	144.94	155.33	484.25	645.03	1428.98	1753.28
HC-PIN	2.78	14.68	55.02	13.99	34.52	663.50	234.69	372.31

6 Protein Complex Discovery

We validated our results by comparing the communities detected by FAC-PIN with a list of protein complexes obtained from the MIPS database, which we consider as a *gold standard* data. Our validations were done only for four species which we could download corresponding complexes from MIPS. For *Baker's yeast*'s PIN, we obtained complexes from the MIPS *Comprehensive Yeast Genome Database-CYGD*[3]. For the PINs of *Street rat, House mouse, and Human*, the corresponding complexes were downloaded from the MIPS *Comprehensive Resource of Mammalian Protein Complexes- CORUM* [4]. We could not find complexes for the remaining species in due time.

We proceeded similarly to Laarhoven et al. [10] and considered only the known complexes (i.e., not those obtained by computational means) containing at least three proteins. Since FAC-PIN generates non-overlapping communities, we considered only complexes which are at the bottom of the MIPS hierarchy of complexes and subcomplexes. The unconfirmed complexes, that is those in category 550, were excluded.

The validation proceeds by determining the degree of overlap between the communities identified by FAC-PIN and the protein complexes; that is, we want to determine how effectively a community matches a known complex. We used the *overlapping score* function given in [2, 3, 10, 19]. The overlapping score, $O(C, K)$, between a community C and a known complex K is defined as:

$$O(C, K) = \frac{|C \cap K|^2}{|C| \times |K|}, \tag{6}$$

A community C is considered to match a known complex K whenever $O(C, K) \geq \tau$; where, $0 < \tau \leq 1$ is the matching threshold. We have a perfect match only when $O(C, K) = 1$. Threshold value $\tau = 0.2$ was used in [2, 3, 19] whereas [10] used $\tau = 0.25$. We used both values of τ in our complex validation. After computing the overlapping scores between all pairs (C, K) of communities and known complexes for a given PIN, we then determined the ability of FAC-PIN to correctly classify the known complexes. The reason for doing this is that a given complex K_1 may match many communities but with different degrees of overlap, while another complex K_2 may match with a single community only. Hence, we calculated the *Specificity*, the *Sensitivity*, and the F-*score*, as our measures of accuracy; they are defined as follow:

[3] `ftp://ftpmips.gsf.de/yeast/catalogues/complexcat/`

[4] `http://mips.helmholtz-muenchen.de/genre/proj/corum/`

$$Sensitivity = \frac{TP}{TP + FN} \tag{7}$$

$$Specificity = \frac{TP}{TP + FP} \tag{8}$$

$$F\text{-score} = \frac{2 \times specificity \times sensitivity}{specificity + sensitivity} \tag{9}$$

where, *TP* (true positive) is the number of the identified communities C matched by the known complexes K, *FN* (false negative) is the number of known complexes that are not matched by the communities, and *FP* (false positive) is the total number of the identified communities C minus *TP*. Table 4 shows the comparison results on the protein complexes of the *Specificity*, the *Sensitivity*, and the F-*score* of FAC-PIN, HC-PIN and CNM. The results are shown for the two values of threshold τ (discussed above) and for the modularity scoring function Q. For HC-PIN, results are shown for two values of its parameter λ ([19] showed validation results with these two values of λ).

In Table 4, we see that FAC-PIN identifies communities whose average sizes (column 8) are closer to the average sizes of the known protein complexes (column 4), whereas HC-PIN and CNM yield larger averages of cluster sizes. The consequence of this is that smaller FAC-PIN communities produce higher accuracy (*Specificity*, *Sensitivity* or F-*score*) in the great majority of cases. This is because, most of the known complexes are small, and thus the accuracy increases as the size of a complex decreases. In particular, we obtain a larger number of perfectly matched complexes to communities with FAC-PIN than with HC-PIN or CNM.

7 Conclusion

In this paper, we devised a new agglomerative clustering approach, FAC-PIN algorithm, for detecting the communities of a given PIN networks, and then compared our method with two fast hierarchical techniques discussed in literature. Our approach is based on a the use of new measure, the *relative vertex clustering value* which helps decide whether a given vertex u should be included within the cluster of another vertex v depending on how many of the neighbors of u form a triangle with u and v. Our approach is very fast since we are examining only the vertices and in an efficient manner, unlike the two compared algorithms which examine edges. Thus our method is appropriate for PINs, which in general contain more interactions than proteins. More study needs to be done and we plan to perform validations based (1) on random networks, in order to analyze the robustness of FAC-PIN, and (2) on gene ontology annotations. Comparisons with other methods which are not necessarily hierarchical will also be important. Non-agglomerative clustering methods based on the relative vertex clustering value will be investigated. Finally, we plan to validate FAC-PIN through *functional enrichment* in order to evaluate how well the identified communities match with know protein functions.

Table 4. Comparison of the *Specificity*, *Sensitivity* and F*-score FAC-PIN*, *CNM* and *HC-PIN*

Species	Number of Proteins	Number of Complexes	Average Complex Size	Threshold τ	Algorithms	Computed results					
						Number of Clusters	Average Cluster Size	Perfectly Matched K	*Sensitivity*	*Specificity*	F*-score*
Baker's yeast	1237	267	4.63	0.2	*FAC-PIN*	285	4.34	12	0.092	0.78	0.164
					CNM	300	4.12	5	0.010	0.33	0.013
					HC-PIN($\lambda = 0.5$)	153	8.08	5	0.090	0.69	0.159
					HC-PIN($\lambda = 1.0$)	111	11.14	3	0.010	0.51	0.019
				0.25	*FAC-PIN*	285	4.34	12	0.090	0.82	0.162
					CNM	300	4.12	5	0.010	0.33	0.013
					HC-PIN($\lambda = 0.5$)	153	8.08	5	0.090	0.55	0.154
					HC-PIN($\lambda = 1.0$)	111	11.14	3	0.008	0.50	0.015
Human	2555	575	4.44	0.2	*FAC-PIN*	607	4.21	8	0.005	0.74	0.010
					CNM	639	3.99	5	0.004	0.40	0.007
					HC-PIN($\lambda = 0.5$)	129	19.80	3	0.005	0.39	0.009
					HC-PIN($\lambda = 1.0$)	119	21.47	3	0.004	0.44	0.007
				0.25	*FAC-PIN*	607	4.21	8	0.005	0.74	0.010
					CNM	639	3.99	5	0.004	0.31	0.008
					HC-PIN($\lambda = 0.5$)	129	19.80	3	0.005	0.39	0.009
					HC-PIN($\lambda = 1.0$)	119	21.47	3	0.004	0.44	0.007
Street rat	557	328	1.69	0.2	*FAC-PIN*	389	1.42	7	0.250	0.43	0.316
					CNM	475	1.17	3	0.109	0.36	0.248
					HC-PIN($\lambda = 0.5$)	117	4.76	1	0.160	0.33	0.214
					HC-PIN($\lambda = 1.0$)	117	4.76	1	0.160	0.33	0.214
				0.25	*FAC-PIN*	389	1.42	7	0.170	0.29	0.214
					CNM	475	1.17	2	0.150	0.27	0.192
					HC-PIN($\lambda = 0.5$)	117	4.76	1	0.110	0.22	0.143
					HC-PIN($\lambda = 1.0$)	117	4.76	1	0.110	0.22	0.143
House mouse	935	460	2.03	0.2	*FAC-PIN*	568	1.64	13	0.230	0.59	0.327
					CNM	605	1.54	6	0.120	0.56	0.198
					HC-PIN($\lambda = 0.5$)	241	3.87	3	0.180	0.48	0.265
					HC-PIN($\lambda = 1.0$)	151	6.19	3	0.110	0.50	0.182
				0.25	*FAC-PIN*	568	1.64	13	0.212	0.55	0.306
					CNM	605	1.54	6	0.120	0.56	0.198
					HC-PIN($\lambda = 0.5$)	241	3.87	3	0.153	0.41	0.222
					HC-PIN($\lambda = 1.0$)	151	6.19	3	0.110	0.50	0.182

References

1. Altaf-Ul-Amin, M.: Development and Implementation of an Algorithm for Detection of Protein Complexes in Large Interaction Networks. BMC Bioinformatics 7(207) (2006)
2. Bader, G.D., Hogue, C.W.: An Automated Method for Finding Molecular Complexes in Large Protein Interaction Networks. BMC Bioinformatics 7(2) (2003)
3. Chua, H.N., Ning, K., SUng, W.-K., Leong, H.W., Wong, L.: Using Indirect Protein-Protein Interaction for Protein Complex Prediction. Journal of Bioinformatics and Computational Biology 6(3), 435–466 (2008)
4. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Nature 453(7191) (2005)
5. Fortunato, S.: Community detection in graphs. Elsevier Physics Reports 486, 75–174 (2010)
6. Friedel, C., Zimmer, R.: Inferring Topology from Clustering Coefficients in Protein-Protein Interaction Networks. BMC Bioinformatics 7(519) (2006)

7. Girvan, M., Newman, M.E.: Community Structure in Social and Biological Networks. Proceedings of Natural Academy of Science USA 99, 7821–7826 (2002)
8. Hartuv, E., Shamir, R.: A Clustering Algorithm Based on Graph Connectivity. Information Processing Letters 76(4-6), 175–181 (2000)
9. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: From molecular to modular cell biology. Nature 402, C47–C52 (1999)
10. van Laarhoven, T., Marchiori, E.: Robust Community Detection Methods with Resolution Parameter for Complex Detection in Protein Protein Interaction Networks. In: Shibuya, T., Kashima, H., Sese, J., Ahmad, S. (eds.) PRIB 2012. LNCS (LNBI), vol. 7632, pp. 1–13. Springer, Heidelberg (2012)
11. Li, X.L., Tan, S., Foo, C., Ng, S.: Interaction Graph Mining for Protein Complexes Using Local Clique Merging. Genome Informatics 16, 260–269 (2006)
12. Li, M., Wang, J.X., Chen, J.E.: A Fast Agglomerative Algorithm for Mining Functional Modules in Protein Interaction Networks. In: Proceedings of First International Conference on BioMedical Engineering and Informatics (BMEI), pp. 3–7 (2008)
13. Luo, F.: Modular Organization of Protein Interaction Networks. BMC Bioinformatics 23(2), 207–214 (2007)
14. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review 69(066133) (2003)
15. Pei, P., Zhang, A.: A 'Seed-Refine' Algorithm for Detecting Protein Complexes from Protein Interaction Data. IEEE Transcations of Nanobioscience 6(1), 43–50 (2007)
16. Radicchi, F., Castellano, C., Cecconi, F.: Defining and Identifying Communities in Networks. Proceedings of Natural Academy of Sciences USA 101(9), 2658–2663 (2004)
17. Spirin, V., Mirny, L.A.: Protein Complexes and Functional Modules in Molecular Networks. Proceedings of Natural Academy of Science USA 100(21), 12123–12128 (2007)
18. Wang, R.-S., Zhang, S., Wang, Y., Zhang, X.-S., Chen, L.: Clustering complex networks and biological networks by non-negative matrix factorization with various similarity measures. Elsevier Neurocomputing 72 (2008)
19. Wang, J., Li, M., Chen, J., Pan, Y.: A Fast Hierarchical Clustering Algorithm for Functional Modules Discovery in Protein Interaction Networks. IEEE/ACM Transaction on Computational Biology and Bioinformatics 8(3) (2011)
20. Yook, S., Olvai, Z., Barabsi, A.L.: Functional and Topological Characterization of Protein Interaction Networks. Protenomics 4, 928–942 (2004)