# DEEN: a Simple and Fast Algorithm for Network Community Detection

Pavol Jancura, Dimitrios Mavroeidis, Elena Marchiori*

Radboud University Nijmegen,
Intelligent Systems, Institute for Computing and Information Sciences,
Postbus 9010, 6500GL Nijmegen, The Netherlands
{jancura,D.Mavroeidis,elenam}@cs.ru.nl
http://www.cs.ru.nl/~jancura
http://sites.google.com/site/mavroeid/
http://www.cs.ru.nl/~elenam

**Abstract.** This paper introduces an algorithm for network community detection called DEEN (Delete Edges and Expand Nodes) consisting of two simple steps. First edges of the graph estimated to connect different clusters are detected and removed, next the resulting graph is used for generating communities by expanding seed nodes.
DEEN uses as parameters the minimum and maximum allowed size of a cluster, and a resolution parameter whose value influences the number of removed edges. Application of DEEN to the budding yeast protein network for detecting functional protein complexes indicates its capability to identify clusters containing proteins with the same functional category, improving on MCL, a popular state-of-the-art method for functional protein complex detection. Moreover, application of DEEN to two popular benchmark networks results in the detection of accurate communities, substantiating the effectiveness of the proposed method in diverse domains.

**Keywords:** Community detection, protein interaction networks, graph sparsification, heuristic search.

## 1 Introduction

Many real world phenomena can be modeled as complex interaction networks. Community detection in these networks amounts to detecting groups of nodes such that the nodes belonging to the same group are more connected to each other than to nodes in the rest of the network. Community detection is of high practical relevance in domains as diverse as protein complex detection, community discovery in social networks and many others.

There is a large amount of methods for finding communities in a network (see for instance [9], a recent survey on this subject). Recently, an alternative view on this problem has been proposed in [25], based on the observation that a

* Corresponding author

network consists of communities which are embedded in a background, that is, a set of nodes that do not belong to any community. The authors proposed a sequential method for detecting communities (and background nodes). Specifically, they formalized the task of finding one community by means of an optimization problem, which was tackled using an heuristic local search algorithm based on tabu-search, and use a parameter specifying the maximum size of a cluster as termination criterion when the desired number of communities is not specified a priori. A drawback of this algorithm is that it requires a number of random starting values and random orders of nodes for finding one community.

In this paper we present an alternative method for community (and background) detection, called `DEEN` (Delete Edges and Expand Nodes), consisting of the following two steps. First, edges considered to link different clusters are detected and deleted from the network. Next, the resulting network is used for generating clusters in a sequential way, each time starting from one seed node having highest degree.

The main advantages of `DEEN` with respect to other methods for community detection, like the one above mentioned, are its simplicity and efficiency. Indeed, `DEEN` uses a simple probabilistic local criterion for scoring edges. Edges with score bigger than a given threshold value are deleted. Communities with minimum and maximum size specified by the user are sequentially generated from seeds using a simple node expansion procedure.

In order to test the effectiveness of `DEEN` we applied it to two benchmark networks for which a "true" community structure is known. Moreover we applied `DEEN` to the protein-protein interaction (PPI) network in budding yeast, previously analyzed in [6].

Results of these experiments show that `DEEN` is capable of detecting communities with high accuracy, substantiating its competitiveness with state-of-the-art methods.

A preliminary version of this paper appeared in [13].

## 2 The Algorithm

Before introducing the proposed algorithm, we describe the notation and terminology used throughout the paper. A network is represented by an undirected graph $G = (V, E)$, where $V$ is the set of nodes denoted by $s, t, u, v, \ldots$ and $E$ the set of edges connecting pairs of nodes, denoted by $e, g \ldots$. An edge $e = uv \in E$ represents the relation between nodes $u$ and $v$ (for instance, in a PPI network, an interaction between two proteins).

Given a graph $G = (V, E)$, nodes joined by an edge are called *adjacent*. A *neighbor* of a node $u \in V$ is a node adjacent to $u$. We denote by $N(u)$ the the set of neighbors of $u$. The degree of $u$, denoted by $k_u$, is the number of elements in $E$ containing the vertex $u$. In the sequel we assume that $G$ has $n$ nodes and $m$ edges.

The size of a set $S$ is denoted by $|S|$ and the set-theoretical difference between two sets $S, T$ is denoted by $S \setminus T$.

### 2.1 DEEN: Delete Edges and Expand Nodes

The proposed algorithm consists of two steps, called "Delete Edges" and "Expand Nodes", respectively. These steps are explained in detail in the sequel.

**Delete Edges.** The procedure for deleting edges is based on a local scoring function $w$ that assigns a weight to each edge in $E$. Edges with weight bigger than a given threshold are deleted.

The scoring function $w : E \to \Re$ maps edges to real numbers, such that for $e = st$

$$w(e) = \begin{cases} \frac{(|N(s) \setminus N(t)|-1)(|N(t) \setminus N(s)|-1)}{(m-2|N(t) \cap N(s)|-1)} \frac{(m-1)}{(|N(s)|-1)(|N(t)|-1)} & \text{if } |N(s)| \cdot |N(t)| > 0 \\ 1 & \text{otherwise} \end{cases}$$

The weight $w(e)$ quantifies the strength of the signal induced by $e$ in a local neighborhood of that edge. Specifically, we define the *local signal induced by* $e$ to be the set of other edges occurring in triangles passing through $e$, that is,

$$signal(st) = \{sr, tr \mid r \in N(s) \cap N(t)\}.$$

Then we can show that $w(e)$ is the quotient of the expected number of edges between $s$ and $t$ under the *configuration null model* (see, e.g., [18]) between the two following graphs: $G$ where both $e$ and the signal induced by $e$ have been removed, and $G$ where only $e$ has been removed.

Indeed, in the configuration null model, a vertex can be attached to any other vertex of the graph and the probability that vertices $s$ and $t$, with degrees $k_s$ and $k_t$, are connected, can be calculated directly as follows. In order to form an edge between $s$ and $t$ one needs to join two stubs (i. e., half-edges), incident with $s$ and $t$. The probability $p_s$ to pick at random a stub incident with $s$ is $\frac{k_s}{2m}$, as there are $k_s$ stubs incident with $s$ out of a total of $2m$. The probability of a connection between $s$ and $t$ is then given by the product $p_s p_t$, since edges are placed independently of each other. The result is $p_s p_t = \frac{k_s k_t}{4m^2}$, which yields an expected number $P_{st} = 2m p_s p_t = \frac{k_s k_t}{2m}$ of edges between $s$ and $t$.

One can verify that

$$w(e) = \frac{P_{st}^{G_1}}{P_{st}^{G_2}},$$

where $G_1$ is the subgraph of $G$ with the same set of nodes and with set of edges equal to $E \setminus (\{e\} \cup \text{signal}(e))$ and $G_2$ is obtained from $G$ by removing only the edge $e$.

If $s$ and $t$ have no common neighbors, that is, $N(s) \cap N(t) = \emptyset$, then $w(e) = 1$. If either $s$ or $t$ have degree 1 then $w(e) = 0$ reaches its minimum value.

The scoring function $w$ is used to perform edge deletion: edges with score greater than a threshold $\gamma$ are removed from $G$. The parameter $\gamma$ influences the resolution of the communities to be found. In general, using a low value of $\gamma$ will result in the removal of more edges, hence a larger amount of nodes will become background.

**Expand Nodes.** After deleting all edges of $G$ with $w(e) > \gamma$ we obtain a new graph, say $G'$. We build clusters by applying to $G'$ the following local clustering procedure that iteratively selects and expands a seed node.

Specifically, a clustering is generated as follows: a node $u$ with highest degree is selected (ties are broken at random) and used as first element of a cluster. This cluster is expanded by adding the set $N(u)$ of all neighbors of $u$ in $G'$. Such expansion step is applied to all the elements added to the cluster. The procedure is iterated until either no neighbor can be added, or a given upper bound ($max\_size$) on the maximum size of a cluster is reached, or when there are more edges with both nodes in the cluster than edges with only one node in the cluster (this latter condition corresponds to the notion of weak community as defined in [19]). At that point the nodes of the constructed cluster are removed from $G'$ and a new seed (node with highest degree) is selected and expanded. The process continues until all nodes have been assigned to a cluster. The set of nodes occurring in clusters with size smaller than a given lower bound ($min\_size$) form the background.

**Time Complexity.** The time complexity of `DEEN` is dominated by the computation of the score of each edge. Assuming the adjacency lists for each node to be pre-sorted, intersecting the adjacency lists for two nodes takes number of operations proportional to the sum of their degrees. Since a node $s$ of degree $k_s$ requires $k_s$ intersections, the total number of operations is proportional to $\sum_s k_s^2$. Nevertheless, using fast approximation techniques (see [4, 20]) one can compute an estimate of the score of all edges in linear time on the number $|E|$ of edges.

## 3 Related Work

Our approach is related to other methods for community detection and clustering in networks based on seed expansion, like those used for clustering PPI networks, `SPICi` [14], `DPClus` [2], and `SCAN` [17], our previous work on modular network comparative analysis [12], and other methods not specifically developed for analyzing biological networks (e.g., [1]). Their appropriateness essentially depends on the input graph properties. For instance, for graphs with small diameter a breadth-first search (BFS) style expansion strategy may be more heavily influenced by the graph's small world properties (i.e., small shortest paths between all the graph nodes) rather than the node community structure; while in the cases of graphs with large diameter a BFS approach may present a simple and easy way to implement a strategy for detecting the desired clusters. While these methods employ a more involved criterion for seed selection and expansion, `DEEN` acts on a sparsified graph which allows one to use a very simple cluster expansion criterion based on node degree.

`DEEN` is also related to approaches that transform the graph prior to clustering. An example of this approach is the algorithm proposed in [21]: hub proteins (with degree greater than a given threshold) are first selected and their

neighborhood graphs are subsequently constructed. A hub-duplication strategy is then applied to detect dense subgraphs in these neighborhood graphs with multi-functional hub proteins assigned to multiple clusters. While the goal of this method is to better identify overlapping complexes with multi-functional proteins, DEEN aims at identifying boundaries between clusters, characterized in terms of edges.

Another example of a related approach is the popular Girvan-Newman algorithm for community detection [11], applied to detect functional complexes in PPI network, e.g., in [7]. This clustering algorithm is based on a measure that tells us which edges are more likely to be "between" communities. The communities are detected by progressively removing edges from the original graph, in a greedy fashion, until a criterion measuring the modularity of the resulting network partition is satisfied. The measure employed in the algorithm is the so-called edge betweeness, defined as the number of shortest paths between pairs of nodes that pass through that edge. If there is more than one shortest path between a pair of nodes, each path is assigned equal weight such that the total weight of all of the paths is equal to one. Edges connecting communities will have high edge betweenness (at least one of them). By removing these edges, the groups are separated from one another and so the underlying community structure of the network is revealed.

DEEN differs from the Girvan-Newman algorithm in three main aspects: (a) it uses a different *local* measure for quantifying the likeliness of an edge to be between clusters, which employs information on the nodes adjacent to the two nodes of that edge; (b) it removes edges in one step, instead of iteratively; (c) it detects communities by performing seed expansion on the network resulting from the application of the edge deletion step. Thus DEEN is a fast local heuristic method, which uses graph transformation as a pre-processing instead of as a core step of a clustering algorithm.

Many variants of the Girvan-Newman algorithm have been proposed, including algorithms based on local measures for edges or nodes (for an overview see [9]). In particular, the measure used in DEEN for edge removal is related to the notion of edge-clustering coefficient used in the local method for community detection presented in [19]. Edge-clustering coefficient, is defined as the number of triangles (or other higher order cycles) to which a given edge belongs plus one, divided by the number of triangles that might potentially include it, that is, the minimum of the degrees of the edge nodes minus one.

Conceptually, the edge deletion component of our framework is also related to graph sparsification techniques that remove graph-edges with the general goal of approximating certain graph properties of interest. Most relevant to our work are the graph sparsification approaches that aim in preserving cuts (up to a multiplicative error) in the reduced, "sparsified" graph (i.e. [10] and references therein). As opposed to these approaches our work defines a probabilistic local model for removing the edges and does not directly aim in approximating cluster-structure properties of the full graph. Other related work is local graph

sparsification for scalable clustering [20], which tries to remove edges in order to enable faster graph clustering.

## 4 Experimental Evaluation

### 4.1 Benchmark Networks

To illustrate the proposed method and test its performance, we apply it to two popular benchmark networks: the karate club network and the US College football network. We measure the quality of discovered communities in terms of cluster purity, that is, the fraction of all pairs of nodes that are assigned to that cluster and belong to the same true cluster. The average purity of the discovered communities is reported as quality measure of the performance of `DEEN`.

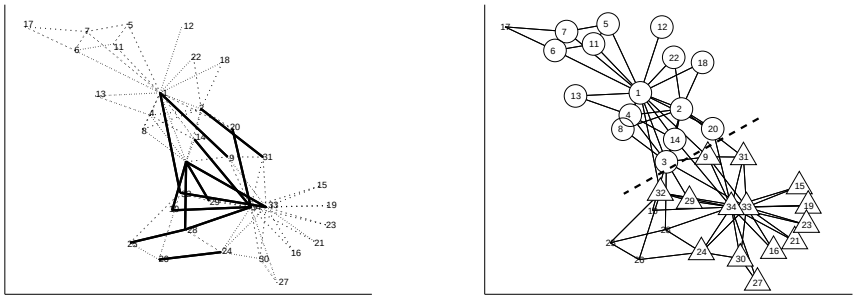

**Fig. 1.** (Application of `DEEN` to the Zachary social network (with $\gamma = 0.6$, minimum and maximum size of clusters equal to 3 and 15, respectively). Left-hand side figure: deleted edges are plotted using thick lines. Right-hand side figure: the two clusters detected by `DEEN` with cluster identifiers denoted by a circle and a triangle symbol, respectively. Nodes with no identifier are not assigned to any cluster and form the background. The dash line separates the two true communities.

**The karate club network.** The karate club network describes the friendship relation between members of a karate club studied by Zachary [24]. During the course of the study, the administrator and the club's instructor quarreled, and the club split into two factions. We consider these fractions as a true clustering for testing the performance of `DEEN`. The network contains 34 nodes and 156 edges. Figure 1 (left-hand side) shows the network after the application of the edge deletion step, while the plot on the right-hand side shows the clustering generated by `DEEN` (obtained using $\gamma = 0.6$, $min\_size = 3$ and $max\_size = 15$). The instructor and the administrator are represented by nodes 1 and 34, respectively.

The results show that DEEN correctly identifies the presence of two communities, each of them containing only elements belonging to the same true community. Therefore the average purity of DEEN applied to this network is equal to 1. Five nodes are not assigned to clusters: these nodes are weakly connected to the identified communities, and are considered by DEEN to be background.

When varying the value of $\gamma$ a different number of nodes is not assigned to any cluster, while the average purity of the resulting clusters remains equal to 1 except for the case $\gamma = 1$ (no edge removal). In this case DEEN detects four clusters, node 3 is mapped to the wrong community, and 8 nodes are assigned to the background.

**The US College football network.** The US College football network [11] describes the schedule of Division 1 games for the 2000 season: the nodes are the teams and each edge represents a game between two teams. The network contains 115 nodes and 499 edges. The teams are divided into so-called conferences which provide a true clustering that we use for evaluating the results of DEEN.

Applying DEEN to this network, we find that it identifies the conference structure with a high degree of purity. Almost all teams are correctly grouped with the other teams in their conference. The average purity for different values of $\gamma$, the number of deleted edges and the number of nodes assigned to the background are plotted in Figure 2.

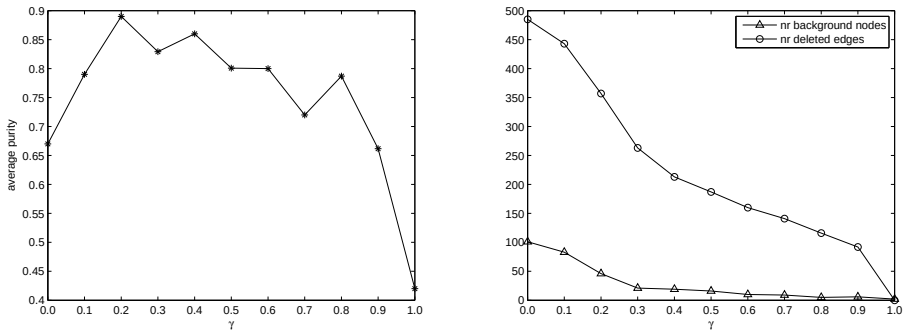


**Fig. 2.** Application of DEEN to the US College football network (with minimum and maximum size of clusters equal to 3 and 15, respectively. Left-hand side plot: average purity for different values of $\gamma$. Right-hand side plot: number of deleted edges and number of nodes assigned to the background for different values of $\gamma$.

### 4.2 Protein Complex Detection in the Budding Yeast PPI Network

With the exponential increase of data on protein interactions obtained from advanced technologies, data on thousands of interactions in human and most model species have become available [3, 23]. PPI networks offer a powerful representation for better understanding modular organization of cells, for predicting biological functions and for providing insights into a variety of biochemical processes. In particular, PPI networks can be used for detecting protein functional modules and complexes and for assigning function to yet uncharacterized proteins.

Among the interactions produced by high-throughput methods there could be many false positives. In [16] the accuracy and the biases of 80 000 physical interactions among 5400 yeast proteins reported previously were assessed and a confidence value was assigned to each interaction.

In order to reduce the interference by false positives, the authors extracted interactions with high and medium confidence. This network was first used in [6] for detecting complexes. We retrieved a version of this network from `http://vlado.fmf.uni-lj.si/pub/networks/data/bio/Yeast/Yeast.htm`: it contains 6646 interactions among 2361 proteins.

**Assignment of Annotation and p-values to Clusters.** As an isolated cluster may involve different functional categories, here we p-values are used as criteria to assign each cluster a main function, as done previously e.g. in [6]. Specifically, hypergeometric distribution was applied to model the probability of observing at least $k$ proteins from a cluster of size $l$ by chance in a category containing $C$ proteins from a network containing $n$ proteins. The resulting p-value $P$ is computed as

$$P = 1 - \sum_{i=1}^{k-1} \frac{\binom{C}{i}\binom{n-C}{l-i}}{\binom{n}{l}}.$$

The above test measures whether a cluster is enriched with proteins from a particular category more than would be expected by chance. If the p-value of a category is near 0, the proteins of the category in a cluster will have a low probability of being chosen by chance. Here, we assigned each cluster the main function with the lowest p-value in all categories.

For each cluster we calculated its p-value and annotated it based on the Munich Information Center (MIPS) hierarchical functional categories, using a set of functional categories provided in previous studies of this dataset [16] (see Table 1).

We call *significant cluster* a cluster with p-value smaller than 0.05.

Moreover, in order to analyze the more pure clusters generated by the algorithm, we considered the subset of significant clusters satisfying the following two conditions: (a) their assigned functional category is different from "U" and (b) they contain only proteins of that category except possibly some uncharacterized proteins. We call these clusters *homogeneous*.

**Table 1.** MIPS functional categories from [16].

| id | function |
|---|---|
| 10 | E Energy production |
| 7 | G Amino-acid metabolism |
| 2 | M Other metabolism |
| 6 | P Translation |
| 1 | T Transcription |
| 12 | B Transcriptional control |
| 5 | F Protein fate |
| 9 | O Cellular organization |
| 13 | A Transport and sensing |
| 11 | R Stress and defence |
| 8 | D Genome maintenance |
| 4 | C Cellular fate/organization |
| 3 | U Uncharacterized |

**Results.** In our analysis, we considered only clusters containing at least 3 elements (thus we set $min\_size = 3$), that is, clusters of a higher complexity than just a single interaction. We set the bound for the maximum size of a complex to be equal to 15, thus focussing on small functional complexes. Prior knowledge on the size of functional complexes could be used to select a more tight upper bound.

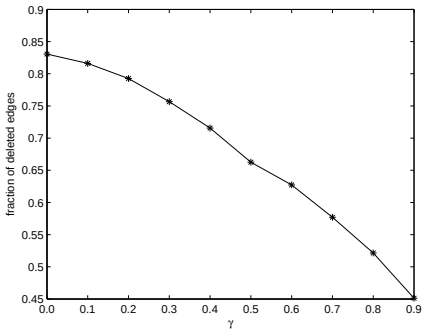


**Fig. 3.** Application of DEEN to the Budding Yeast PPI network. Fraction of edges deleted by DEEN for different values of $\gamma$.

Figure 3 plots the number of edges deleted by DEEN when varying the value of the parameter $\gamma$. The plot shows that a large number of the edges in the considered PPI network is removed, ranging from 45% to more than 80% of the total number of edges.

Figure 4 (the left plot) shows the fraction of significant clusters (that is, with p-value smaller than 0.05) and the number of homogeneous ones, respectively,

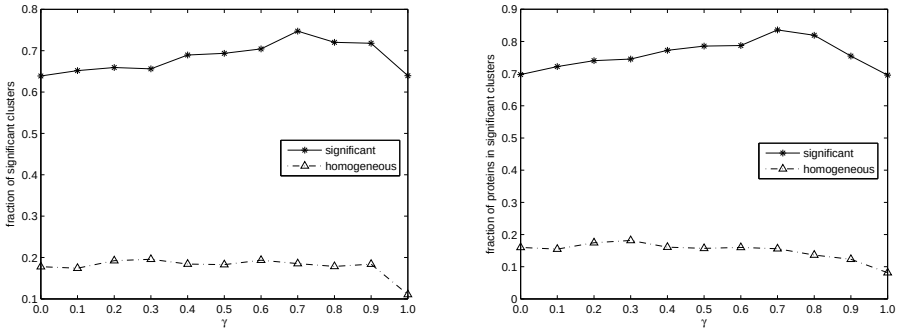


**Fig. 4.** Application of DEEN to the Budding Yeast PPI network. Left: the fraction of significant clusters and of homogeneous ones, respectively, for different values of $\gamma$. Right: the fraction of proteins covered by significant clusters and by homogeneous ones, respectively, for different values of $\gamma$.

for different values of $\gamma$. The right plot shows the behavior of DEEN when varying the value of $\gamma$ with respect to the fraction of the proteins covered by significant clusters and by homogeneous ones, respectively. One can see that for any value of $\gamma$ more than 70% of the clustered proteins are assigned to significant clusters. However, as shown on Table 4.2 (see values in the row with identifier *tp*) the number of proteins assigned to significant clusters tends to increase for higher values of $\gamma$, with a pick for $\gamma = 0.7$.

**Table 2.** Application of DEEN to the Budding Yeast PPI network. Results for different values of $\gamma$. sc = number of significant clusters, hsc = number of homogeneous clusters, tc = total number of clusters, sp= total number of proteins in significant clusters, hp= total number of proteins in homogeneous clusters, tp = total number of proteins in detected clusters, de= number of deleted edges.

| $\gamma$ | 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| sc | 115 | 116 | 120 | 124 | 131 | 129 | 131 | 133 | 121 | 117 | 110 |
| hsc | 32 | 31 | 35 | 37 | 35 | 34 | 36 | 33 | 30 | 30 | 19 |
| tc | 180 | 178 | 182 | 189 | 190 | 186 | 186 | 178 | 168 | 163 | 172 |
| sp | 654 | 691 | 747 | 796 | 846 | 895 | 915 | 1009 | 974 | 846 | 879 |
| hp | 150 | 148 | 176 | 194 | 176 | 179 | 186 | 188 | 162 | 138 | 103 |
| tp | 938 | 957 | 1009 | 1068 | 1095 | 1139 | 1162 | 1207 | 1189 | 1121 | 1264 |
| de | 5520 | 5424 | 5267 | 5029 | 4756 | 4404 | 4169 | 3835 | 3466 | 2999 | 0 |

From the summary of the results contained in Table 4.2 we can draw the following observations. The best performance of DEEN is achieved for $\gamma = 0.7$, with the highest number of proteins assigned to significant and homogeneous

clusters. When no edge deletion is performed (that is, for $\gamma = 1$) very few homogeneous clusters are detected covering about 100 proteins, while using edge deletion with any other value of $\gamma$ (in the considered set) results in the discovery of a larger number of homogeneous clusters covering more proteins. The number of significant clusters generated by DEEN with $\gamma = 1$ is less than that obtained using the other values of this parameter, while the number of proteins covered by these clusters is smaller than the one obtained using $\gamma \in \{0.6, 0.7, 0.8\}$ and similar or bigger than the one obtained using the remaining values of $\gamma$. These results indicate that deleting edges improves the quality of the "Expand Nodes" clustering procedure.

In order to test also the benefits of the type of edge deletion criterion used in DEEN, we replaced it with a random edge deletion procedure. Specifically, we run 10 experiments of DEEN with the randomized procedure for deleting 3835 edges, that is, using the number of edges deleted by the "Delete Edges" algorithm with $\gamma = 0.7$, since this value produced the best results of DEEN. The mean and standard deviation of the results are shown in Table 4.2. These results indicate superior performance of the proposed criterion for deleting edges over random removal.

**Table 3.** Application of DEEN with random edge selection to the Budding Yeast PPI network. hsc = mean number of homogeneous clusters, hp = mean total number of proteins in homogeneous clusters, sc = mean number of significant clusters, sp = mean total number of proteins in significant clusters, tc = mean total number of clusters, tp = mean total number of proteins in detected clusters. Standard deviation is reported between brackets.

| hsc | hp | sc | sp | tc | tp |
|---|---|---|---|---|---|
| 16.9 (3.5) | 80.1 (20.3) | 96.7 (8.1) | 801.7 (6.26) | 144.7 (6.0) | 1115 (24.3) |

**Comparison with MCL.** We compare the results of DEEN (using $\gamma = 0.6$) with those of a state-of-the-art clustering method, called Markov Clustering (MCL) [22, 8].

We briefly describe MCL and refer the reader to [22] for a detailed presentation of this method, and to [8] for its first use for detecting protein families. MCL finds clusters in graphs by a mathematical bootstrapping procedure. The process deterministically computes (the probabilities of) random walks through the graph, and uses two operators transforming one set of probabilities into another. MCL simulates a flow on the graph by calculating successive powers of the associated adjacency matrix. At each iteration, an inflation step is applied to enhance the contrast between regions of strong or weak flow in the graph. The process converges towards a partition of the graph, with a set of high-flow regions (the clusters) separated by boundaries with no flow. The value of the inflation parameter is used to control the granularity of these clusters, hence it indirectly

influences the number of clusters detected by the method. We set this parameter to the value 1.8 as reported in [5] and obtained by tuning for accuracy and separation. MCL with this inflation parameter value was shown to achieve best performance in a comparative experimental assessment of four state-of-the-art algorithms for detecting functional modules in PPI networks. Another recent comparative analysis of algorithms for protein complex detection [15] further substantiated the very good performance of MCL.

DEEN with $\gamma = 0.6$ detected 186 clusters, containing a total of 1162 proteins; 131 of these clusters have p-value less than 0.05, resulting in a percentage of 70.4 high quality clusters covering a total of 915 proteins.

MCL detected 272 clusters with at least 3 elements, containing a total of 2005 proteins; 168 of these clusters have p-value less than 0.05, resulting in a percentage of 61.76 high quality clusters.

The average p-values of MCL and DEEN were 0.0692 (standard deviation equal to 0.0862) and 0.0574 (standard deviation equal to 0.0851), respectively. Figure 5 shows the sorted p-values of the clusters found by DEEN and by MCL.

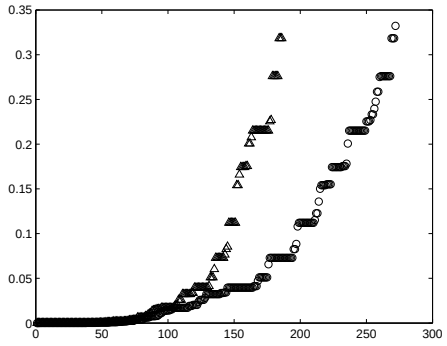


**Fig. 5.** Application of DEEN to the Budding Yeast PPI network. Sorted P-values of the clusters found by DEEN (triangle symbol) and by MCL (circle symbol).

However, the above results should be considered with care due to the bias on the maximum size of clusters considered by DEEN, because larger clusters discovered by MCL may tend to be less significant and homogeneous. Thus, to minimize the effect of this bias, we further focus only on significant and on homogeneous clusters.

MCL discovered 32 homogeneous clusters with 166 proteins in total while DEEN detected 36 homogeneous clusters containing a total of 186 proteins.

The average purity of the significant clusters generated by DEEN for varying values of $\gamma$ and of those discovered by MCL are shown in Figure 6. The plot shows that the significant clusters detected by DEEN are more pure than those

generated by MCL for all values of $\gamma$ except when no edge deletion is performed ($\gamma = 1$).

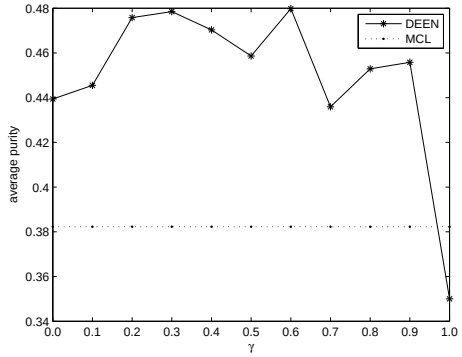


**Fig. 6.** Application of DEEN and MCL to the Budding Yeast PPI network. Average purity of MCL and of DEEN when varying $\gamma$.

These results show that DEEN is more conservative and more accurate than MCL for the following reasons: (1) it found less clusters and clustered less proteins than MCL, (2) it detected a higher number of homogeneous clusters (and covered proteins) as MCL, (3) it detected clusters having average p-value smaller than that of the clusters detected by MCL, and (4) it generated significant clusters with higher average purity than that of those generated using MCL.

## 5 Conclusion

We proposed a new efficient algorithm called DEEN for detecting high quality communities and background in a network.

We assessed experimentally the effectiveness of this algorithm on two benchmarks networks and we compared its performance with that of MCL, a state-of-the-art method for functional complex detection, on a PPI network. Results indicated that DEEN is capable of detecting accurate functional protein modules and is competitive with a current state-of-the-art methods for functional complex detection in PPI networks.

Possible interesting topics to be addressed in future work include extensions of the proposed method to analyze weighted and directed networks, as well as the development of methods for tuning the parameters of DEEN.

## References


1. Alamgir, M., von Luxburg, U.: Multi-agent random walks for local clustering on graphs. In: Proceedings of the 2010 IEEE International Conference on Data Mining. pp. 18–27. ICDM '10, IEEE Computer Society, Washington, DC, USA (2010)
2. Altaf-Ul-Amin, M., Shinbo, Y., Mihara, K., Kurokawa, K., Kanaya, S.: Development and implementation of an algorithm for detection of protein complexes in large interaction networks. BMC bioinformatics 7(1) (2006)
3. Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F.F., Pawson, T., Hogue, C.W.V.: Bind–the biomolecular interaction network database. Nucleic Acids Res 29(1), 242–245 (2001)
4. Becchetti, L., Boldi, P., Castillo, C., Gionis, A.: Efficient algorithms for large-scale local triangle counting. TKDD 4(3) (2010)
5. Brohee, S., van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics 7(1), 488+ (2006)
6. Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., Lu, H., Zhang, J., Sun, S., Ling, L., Zhang, N., Li, G., Chen, R.: Topological structure analysis of the protein-protein interaction network in budding yeast. Nucl. Acids Res. 31(9), 2443–2450 (2003)
7. Dunn, R., Dudbridge, F., Sanderson, C.: The Use of Edge-Betweenness Clustering to Investigate Biological Function in Protein Interaction Networks. BMC Bioinformatics 6(1), 39+ (2005)
8. Enright, A.J., Van Dongen, S., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research 30, 1575–1584 (2002)
9. Fortunato, S.: Community detection in graphs. Physics Reports 486, 75–174 (2010)
10. Fung, W.S., Hariharan, R., Harvey, N.J., Panigrahi, D.: A general framework for graph sparsification. In: Proceedings of the 43rd annual ACM symposium on Theory of computing. pp. 71–80. STOC '11, ACM, New York, NY, USA (2011)
11. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America 99(12), 7821–7826 (2002)
12. Jancura, P., Marchiori, E.: Dividing protein interaction networks for modular network comparative analysis. Pattern Recognition Letters 31(14), 2083–2096 (2010)
13. Jancura, P., Marchiori, E.: Detecting high quality complexes in a PPI network by edge deletion and node expansion. In: CIBB (2011)
14. Jiang, P., Singh, M.: SPICi: a fast clustering algorithm for large biological networks. Bioinformatics (Oxford, England) 26(8), 1105–1111 (2010)
15. Li, X., Wu, M., Kwoh, C.K., Ng, S.K.: Computational approaches for detecting protein complexes from protein interaction networks: a survey. BMC Genomics 11(Suppl 1), S3+ (2010)
16. von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S.G., Fields, S., Bork, P.: Comparative assessment of large-scale data sets of protein-protein interactions. Nature 417, 399–403 (2002)
17. Mete, M., Tang, F., Xu, X., Yuruk, N.: A structural approach for finding functional modules from large biological networks. BMC Bioinformatics 9(S-9) (2008)
18. Molloy, M., Reed, B.A.: A critical point for random graphs with a given degree sequence. Random Struct. Algorithms 6(2/3), 161–180 (1995)
19. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proceedings of the National Academy of Sciences of the United States of America 101(9), 2658–2663 (2004)

20. Satuluri, V., Parthasarathy, S., Ruan, Y.: Local graph sparsification for scalable clustering. In: Proceedings of the 2011 international conference on Management of data. pp. 721–732. SIGMOD '11, ACM, New York, NY, USA (2011)
21. Ucar, D., Asur, S., Çatalyürek, Ü.V., Parthasarathy, S.: Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs. In: PKDD. Lecture Notes in Computer Science, vol. 4213, pp. 371–382. Springer (2006)
22. Van Dongen, S.: Graph Clustering Via a Discrete Uncoupling Process. SIAM Journal on Matrix Analysis and Applications 30(1), 121–141 (2008)
23. Xenarios, I., Salwínski, Ł., Duan, X.J., Higney, P., Kim, S.M., Eisenberg, D.: Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Research 30(1), 303–305 (2002)
24. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)
25. Zhao, Y., Levina, E., Zhu, J.: Community extraction for social networks. Proceedings of the National Academy of Sciences 108(18), 7321–7326 (May 2011)