

TECHNICAL REPORT

Mining the Largest Dense N-vertexlet in a Fuzzy Scale-free Graph

Technical Report No.- MIU/TR-03/08

by

Sanghamitra Bandyopadhyay

and

Malay Bhattacharyya



Machine Intelligence Unit

Indian Statistical Institute

203 B. T. Road, Kolkata - 700108, India.

December 2008

Abstract

The topology of many real-life networks, e.g., the world wide web, social networks, ecological networks, genetic networks, comply with scale-free models. In scale-free models, the vertices of the underlying graph follow a power-law degree distribution. Observably, the graphs corresponding to most of the real-life networks are fuzzy in nature. An important problem of knowledge engineering that has evolved in various real-life networks is to identify the largest group of similar vertices in such networks that are functionally associated. Here, the problem of finding the largest group or association of vertices that are dense (denoted as dense N-vertexlet) in a fuzzy scale-free graph is addressed. Density quantifies the degree of similarity within a group of vertices in a graph. The density of an N-vertexlet is defined in a novel way that ensures significant participation of all the vertices in the N-vertexlet. First, it is established that the problem is NP-complete in nature. An upper bound on the size of the largest dense N-vertexlet in a fuzzy graph, with respect to certain density threshold value, is then derived. Finally, an $O(n^2 \log n)$, n being the number of vertices in the graph, heuristic graph mining algorithm that produces an approximate solution for the problem is presented. The efficacy of the method is shown by applying it on some artificial and biological networks.

Contents

1	Introduction	1
2	Related works	2
3	Problem Formulation	4
3.1	Basic Definitions and Preliminaries	4
3.2	Problem Definition and Complexity Analysis	6
3.3	Deriving an Upper Bound for <i>MAX-DAN</i>	7
4	A Solution Approach to the <i>MAX-DAN</i> Problem	10
4.1	Complexity Analysis	11
4.2	Effectiveness of the Proposed Density Measure	13
5	Results	15
5.1	Artificial Datasets	16
5.2	Biological Datasets	19
6	Discussion and Conclusions	29

List of Figures

1	A fuzzy graph having the set $\{v_1, v_2, v_3, v_4\}$ as the largest, dense N-vertexlet.	3
2	Generation of false DAN.	5
3	(a) The fuzzy weight matrix of an FCG and (b) Construction of <i>NList</i> from the FCG represented in (a) w.r.t. $\delta = 0.7$.	9
4	Updating the connector list of V_{let}^{Nmax} from \tilde{G} .	11
5	Degree distribution in (a) Scale-freeFCG_1 w.r.t. $\delta = 0.7$, (b) Scale-freeFCG_2 w.r.t. $\delta = 0.7$, (c) Scale-freeFCG_3 w.r.t. $\delta = 0.5$, and (d) Scale-freeFCG_4 w.r.t. $\delta = 0.7$.	17
6	Expression graph of largest DAN found in Eisen dataset having (a) $\tilde{\omega}(\tilde{G}) = 308$ w.r.t. $\delta = 0.7$, (b) $\tilde{\omega}(\tilde{G}) = 163$ w.r.t. $\delta = 0.8$, and (c) $\tilde{\omega}(\tilde{G}) = 94$ w.r.t. $\delta = 0.9$. Eisen plot of largest DAN found in Eisen dataset having (d) $\tilde{\omega}(\tilde{G}) = 308$ w.r.t. $\delta = 0.7$, (e) $\tilde{\omega}(\tilde{G}) = 163$ w.r.t. $\delta = 0.8$, and (f) $\tilde{\omega}(\tilde{G}) = 94$ w.r.t. $\delta = 0.9$.	21
7	Gene Ontology biological process level 5 partial results.	22
8	Expression graph of the most dense cluster found in Eisen dataset using (a) K-Means of size 93, (b) K-Medians of size 73, (c) HCL (weighted linkage) of size 162, and (d) CRC of size 41. Eisen plot of the most dense cluster found in Eisen dataset using (e) K-Means of size 93, (f) K-Medians of size 73, (g) HCL (weighted linkage) of size 162, and (h) CRC of size 41.	24
9	Expression graph of largest DAN found in Human Fibroblasts Serum dataset having (a) $\tilde{\omega}(\tilde{G}) = 186$ w.r.t. $\delta = 0.7$, (b) $\tilde{\omega}(\tilde{G}) = 111$ w.r.t. $\delta = 0.8$, and (c) $\tilde{\omega}(\tilde{G}) = 39$ w.r.t. $\delta = 0.9$. Eisen plot of largest DAN found in Serum dataset having (d) $\tilde{\omega}(\tilde{G}) = 186$ w.r.t. $\delta = 0.7$, (e) $\tilde{\omega}(\tilde{G}) = 111$ w.r.t. $\delta = 0.8$, and (f) $\tilde{\omega}(\tilde{G}) = 39$ w.r.t. $\delta = 0.9$.	27
10	Expression graph of the most dense cluster found in Human Fibroblasts Serum dataset using (a) HCL (weighted linkage) of size 49 and (b) CRC of size 55. Eisen plot of the most dense cluster found in Serum dataset using (c) HCL (weighted linkage) of size 49 and (d) CRC of size 55.	28

1 Introduction

The real-life networks such as world wide web, social networks, ecological networks, gene regulatory networks and likewise are best modeled with an underlying graph that abide by a power-law degree distribution [2]. This feature was found to be a consequence of two generic mechanisms: (1) networks expand continuously by the addition of new vertices, and (2) new vertices prefer to attach with those vertices that are already well connected [3]. Networks, where the vertices follow a power-law degree distribution, are referred to as scale-free networks. In brief, the scale-free model of networks are characterized by a graph in which the probability $P(k)$ that a vertex has degree k (i.e., it is connected to k other vertices) decays as a power law, following $P(k) \sim k^{-\gamma}$ ($\gamma > 0$), where γ is a constant [2]. Thus, some of the vertices produce groups or clusters with high connectivity, whereas others remain dispersed. The problem of finding the largest association of vertices that are dense (denoted as largest dense N-vertexlet) from such scale-free graphs is an important problem in real-life networks. Again, this is, in general an essential problem for various networks in the perspective of knowledge mining.

Suppose, a group of network servers exchanging data among themselves. A strict upper bound I_{upper} is specified on the average amount of data these servers can send (or receive). The term average amount formally signifies that at any instant of time t , a server can send (or receive) I_t (where $I_t < n \cdot I_{upper}$) amount of data to (or from) other n servers in parallel. If a set of servers start exceeding this limit (I_{upper}) while sending (or receiving) data between themselves then the entire system may crash or there might be a security problem. Let our goal be to identify the set of those servers associated with such activity. We assume the servers to be the vertices of a graph and the amount of data each of the server pairs exchanges to be the weight between the vertices representing those servers. The weighted graph produced with these assumptions will best describe the aforesaid system. Now, if we can figure out the largest dense N-vertexlet with respect to the threshold value I_{upper} in this graph, that will evidently yield the desired solution to the server problem. Most often similar prerequisites crop up in distributed systems also. The real-time nature of data stream systems and the vast amounts of data they are required to process, introduce new fundamental problems that are not addressed by traditional database management systems [14]. Usefulness of such methods could be also found in various real-life networks or even in spatial data points where the requirement of producing the largest dense clusters, satisfying some threshold, is important. With these motivations, the current article deals with a previously unaddressed problem, referred hereafter as *MAX-DAN*, of mining the largest dense N-vertexlet from scale-free graphs

for a given threshold.

A graph can be considered as fuzzy if there exists a weight function Ω defined over the set of edges (E) of the graph such that $\Omega : E \rightarrow [0, 1]$. It could be thought of as an informal mapping from the definition given in [22]. We consider a fuzzy complete graph as a generalization of a fuzzy graph where all the vertex pairs are connected with an edge. In real-life networks, relationships between the vertices are not always binary, rather they may be fuzzy or probabilistic [12]. In this article a heuristic method for mining the largest Densely Associated N-vertexlet (DAN) in a scale-free fuzzy complete graph for a given threshold is proposed. Since a fuzzy complete graph is essentially a generalization of crisp (unweighted) graphs, the problem being solved is more general in nature. Most importantly, this problem is also unaddressed in the domain of scale-free non-fuzzy graphs. However, the current article encompasses no motivation in this direction of work.

Computational complexity theory is concerned with the time and space complexity of solution models of computational problems. It basically categorizes computational problems into various complexity classes to distinguish their relative complexities [5]. The decision problems (that checks whether a given instance of the problem is a solution or not) verifiable in polynomial time falls in the NP complexity class. The problems in NP that are solvable in polynomial time falls in the P class. Again, a problem that is at least as hard as every problem that is in NP, is defined to be NP-complete. We prove that the reduced decision problem is NP-complete [5].

We have done a comprehensive study on various networks, both real and artificial, to test the performance of our algorithm. A few synthetic datasets have been generated with prior information about the largest DAN. Our algorithm produces promising results as compared to the exact solution. Again, biological and some benchmark datasets have been used to verify the effectiveness of the algorithm. The biological dataset utilized comprises of microarray data [18] of yeast genes. The results obtained from the biological data has been found to support the existing information about this data.

2 Related works

The problem of mining the largest DAN in a fuzzy complete graph has some similarity with that of clustering graphs [30]. Given a graph $G = (V, E)$, where V denotes the set of vertices and E denotes the set of edges, with a slight abuse of notations, we assume E_{V_i} to be the set of edges induced by the set of vertices V_i belonging to G . Then, a c -clustering of the graph G is defined to be a set of clusters $\{G_1, G_2, \dots, G_c\}$ such that i) $G_i = (V_i, E_{V_i}), i = 1, 2, \dots, c$; ii) $V_i \neq \phi, i = 1, 2, \dots, c$; iii) $\cup_{i=1}^c V_i = V$; and

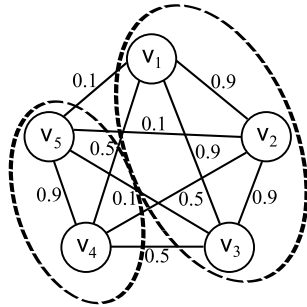


Figure 1: A fuzzy graph having the set $\{v_1, v_2, v_3, v_4\}$ as the largest, dense N-vertexlet.

iv) $V_i \cap V_j = \phi, i \neq j, i, j = 1, 2, \dots, c$. However, the notable difference is, whereas the former one finds groups of vertices supporting a density threshold, the latter one partitions the vertices into coherent groups without taking into account any information about the density threshold. For example, the fuzzy graph of five vertices shown in Fig. 2 will be (and also likely to be) partitioned into the groups $\{v_1, v_2, v_3\}$ and $\{v_4, v_5\}$ by a standard clustering algorithm. However, following our definition provided in the following section, the set of vertices $\{v_1, v_2, v_3, v_4\}$ is the desired largest, N-vertexlet. This is the fundamental perspective where the motivation of both these problems differ. To the best knowledge of the authors, no clustering algorithm addresses the problem of finding the largest dense cluster for a given density threshold. The current state-of-the-art literature reveals the classification of the clustering methods into the following categories: partitional methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods [11] [30]. Although, these approaches by themselves, will be unable to solve the kind of problem aimed at here, it may be possible to modify them suitably to solve the same.

From a different viewpoint, mining the largest DAN in a fuzzy graph can be interpreted as a fuzzification of the Maximum Clique Problem (MCP) [13]. Cliques are by definition complete subgraphs of a graph. Finding the maximum (largest in terms of the cardinality of the vertex set) clique in a non-fuzzy graph is known to be an NP-hard problem [13]. An important study done in the last decade proved that the maximum clique of a graph is hard to approximate within a factor $n^{1-\epsilon}$ (for any $\epsilon > 0$) in polynomial time, where n is the number of vertices in the graph [24]. Various methods for solving MCP e.g., Branch-and-Bound Algorithms [26], Dynamic Programming [4], Constraint Programming [7] [21], Metaheuristic Algorithms [16], Neural Networks [15], Ant Colony Optimization [28], Simulated Annealing [27] etc., have been formulated. An extensive literature survey on the solution methods of MCP is provided in [13]. Most of the approaches reviewed here involved exact algorithms for solving MCP. From a comparative study of the results

of the different methods on the DIMACS benchmark dataset [1], ILOG Solver [21] and SAA [27] are found to be the best among the existing exact and heuristic algorithms, respectively. However, none of these approaches considered the notion of fuzzy logic to define the denseness of a clique. In the present article a new heuristic algorithm is proposed exclusively applicable for the interest of mining the largest DAN in a fuzzy complete graph which is scale-free.

3 Problem Formulation

The formal notations and standard definitions that will be used throughout the paper are first introduced in this section. Then, the necessary proofs appear following the immediate consequences. The term *graph* is used to refer to an undirected labeled simple graph (without self loop or parallel edges). We assume $|S|$ to be the size (cardinality) of a set S and the other notations have their usual meaning, unless specified otherwise.

3.1 Basic Definitions and Preliminaries

Definition 3.1 (Fuzzy Complete Graph) *A Fuzzy Complete Graph (FCG), $\tilde{G} = (V, \tilde{E}, \Omega)$, is defined as a graph in which V denotes the set of vertices, \tilde{E} denotes the set of edges (v_i, v_j) ($v_i \neq v_j, \forall v_i, v_j \in V$) and Ω is a fuzzy edge membership function defined over the set of edges $\Omega : \tilde{E} \rightarrow [0, 1]$.*

Definition 3.2 (N-vertexlet of an FCG) *An N-vertexlet of an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, denoted by V_{let}^N , is defined to be a subset of V (i.e., $V_{\text{let}}^N \subseteq V$) with N ($1 \leq N \leq |V|$) vertices.*

Definition 3.3 (Association Density of a vertex) *Given an FCG $\tilde{G} = (V, \tilde{E}, \Omega)$, the Association Density, $\mu_{v_i/V_{\text{let}}^N}$, of a vertex v_i of \tilde{G} is defined with respect to an N-vertexlet V_{let}^N (such that $v_i \notin V_{\text{let}}^N$) as the ratio of the sum of the fuzzy edge weights between v_i and each of the vertices belonging to V_{let}^N , and the maximum possible sum of edge weights between them. Evidently, the denominator of the aforesaid ratio represents the cardinality of the set V_{let}^N . Thus, the Association Density of a vertex v_i with respect to the N-vertexlet V_{let}^N is computed as,*

$$\mu_{v_i/V_{\text{let}}^N} = \frac{\sum_{v_j \in V_{\text{let}}^N} \Omega_{v_i v_j}}{N}. \quad (1)$$

In Eqn. 1, $\Omega_{v_i v_j}$ denotes the fuzzy weight associated with the edge (v_i, v_j) .

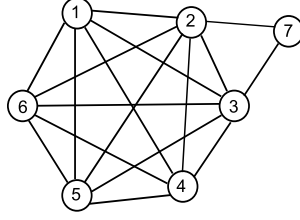


Figure 2: Generation of false DAN.

Definition 3.4 (Dense vertex) We call a vertex, v_i , dense with respect to an Association Density threshold δ and an N -vertexlet $V_{\ell et}^N$ iff $\mu_{v_i/V_{\ell et}^N} \geq \delta$.

Definition 3.5 (Association Density of an N -vertexlet) The Association Density of an N -vertexlet $V_{\ell et}^N$ is defined to be the minimum of the Association Density of every vertex belonging to the N -vertexlet with respect to the remaining $(N-1)$ -vertexlet. So, the Association density of an N -vertexlet $V_{\ell et}^N$ is given by,

$$\mu_{V_{\ell et}^N} = \min_{\forall v_i \in V_{\ell et}^N} \left(\mu_{v_i/V_{\ell et}^N - \{v_i\}} \right). \quad (2)$$

Definition 3.6 (Dense N -vertexlet) We call an N -vertexlet, $V_{\ell et}^N$, dense with respect to an Association Density threshold δ if $\mu_{V_{\ell et}^N} \geq \delta$.

Conventionally, the density of association of an N -vertexlet (or a *graph/subgraph*) is calculated by counting the total number edges it includes. With this measure, it may so happen that an N -vertexlet (or a *graph/subgraph*) becomes dense although it contains some vertices having considerably low degree values. This occurs because the edges are considered in totality, without taking the distribution of edges in the vertex set into account. For example consider the *graph* shown in Fig. 2 where the encircled digits represent the index of the vertices of the *graph*. Here, the N -vertexlet $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ is, in the conventional sense, dense with respect to $\delta = 0.8$. However, it is evident that the association of the vertex v_7 is unreasonable as it has very low connectivity with the remaining vertices. It should be taken care that conceptually a dense association means not only a high overall density but also a high participation density of each member. Moreover, as we are dealing with scale-free networks here, some of their applications require the computation of dense clusters with some minimum density threshold for each vertex [29]. In Def. (3.5) we have therefore incorporated the minimum Association Density, which means a cutoff participation factor, for this purpose.

Definition 3.7 (Fuzzy Scale-free Graph) A fuzzy graph, $\tilde{G} = (V, \tilde{E}, \Omega)$, is said to be scale-free with respect to a weight threshold ω , if the unweighted graph $G = (V, E)$, where

$(V \times V) \rightarrow E$ and $(v_i, v_j) \in E : \Omega_{v_i v_j} \geq \omega, \forall (v_i, v_j) \in \tilde{E}$, follows a power-law degree distribution.

3.2 Problem Definition and Complexity Analysis

With the essential definitions already introduced, we now present the statement of the problem of finding the largest (maximum) DAN formally. At first, we state the fundamental problem of finding the largest DAN, *MAX-DAN*, in an FCG followed by the more general Decision Problem *DAN*. A theoretical analysis of the proper complexity class [9] of *DAN* is also provided.

Problem Statement(MAX-DAN) Given an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, and an Association Density threshold of an N-vertexlet δ , locate a dense N-vertexlet $V_{\text{let}}^{N_{\text{max}}}$ of \tilde{G} that has the maximum cardinality, i.e., $N_{\text{max}} \geq N_i : \forall \mu_{V_{\text{let}}^{N_i}} \geq \delta, \forall N_i = \{1, 2, \dots, |V|\}$.

Now, we put forward the decision version, *DAN*, of the *MAX-DAN* problem.

Problem Statement(DAN) Given an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, and two real numbers $\delta \in [0, 1]$ and $K \geq 2$, determine whether there is an N-vertexlet V_{let}^N of \tilde{G} such that $\mu_{V_{\text{let}}^N} \geq \delta$ and $N \geq K$.

The constraint $K \geq 2$ in the problem definition of *DAN* is clearly justifiable, since an N-vertexlet of size one does not exemplify any suitable meaning about the association of vertices. We now prove that *DAN* is NP-complete.

Theorem 3.1 *DAN is NP-Complete.*

Proof 1 *We first show that the decision problem DAN is in NP. For this, given an N-vertexlet V_{let}^N of an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, along with the supporting parameters δ and K , we should be able to show, in polynomial time, whether $\mu_{V_{\text{let}}^N} \geq \delta$ and $N \geq K$. It is obvious that we can check whether $\mu_{V_{\text{let}}^N} \geq \delta$ by counting the Association Density for all the vertices in $O(n^2)$ time, and whether $N \geq K$ in $O(n)$ time. So, DAN is in NP. To further show that DAN is complete for NP we reduce the Clique Problem (CLIQUE)[5], for a given clique size K for a given graph G , to it.*

Suppose, $G = (V, E)$ is a given unweighted graph. We can construct an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, from G by defining the edge set as $\tilde{E} = \{(v_i, v_j) : \forall v_i, v_j \in V\}$ and the fuzzy weight function as $\Omega_{v_i v_j} = 1$, if $(v_i, v_j) \in E$ and $\Omega_{v_i v_j} = 0$, if $(v_i, v_j) \notin E$. It is apparent from the definitions that an N-vertexlet of an FCG with Association Density 1 is basically a complete graph in an equivalent unweighted graph. So, we can claim that V' is a clique (V' denotes the vertex set of the clique) of G of size K if and only if V' is an N-vertexlet of \tilde{G} of size K with the parameter value $\delta = 1$.

The above reduction is clearly computable in polynomial time. Hence, we have established that CLIQUE is many-one reducible to DAN in polynomial time, i.e., $\text{CLIQUE} \leq_m^P \text{DAN}$ which accordingly proves the NP-completeness of DAN.

3.3 Deriving an Upper Bound for MAX-DAN

Determining the size (number of associated vertices) of the largest DAN in an FCG is again a computationally challenging problem. In the latter part of this section we deal with this important problem, namely, deriving an upper bound for the solution of MAX-DAN problem. The notation $\tilde{\omega}(\tilde{G})$ is used hereafter to denote the size of the largest DAN present in an FCG, \tilde{G} .

Lemma 3.1 *Given an arbitrary vertex v_i of an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, and a set of its neighboring vertices $N = \{v_1, v_2, \dots, v_{|V|-1}\}$ in the descending order of closeness (higher to lower edge weights), the size of the largest DAN in \tilde{G} with respect to the Association Density threshold δ containing the vertex v_i is upper bounded by,*

$$\max_{\forall k \in \{1, 2, \dots, |V|-1\}: \sum_{j=1}^k \Omega_{v_i v_j} \geq \delta k} \binom{k}{k} + 1. \quad (3)$$

Proof 2 *The expression in Eqn. 3 counts the number of neighboring vertices of v_i from the beginning of the set N until the Association Density falls short of δ , plus one for v_i itself. Evidently, no larger DAN than this size can be formed because any further inclusion from (or replacement with) the remaining vertices of N will not succeed in constructing an N -vertexlet having an Association Density more than or equal to δ . This is so because the vertices in N are in the descending order of edge weight with respect to v_i . Hence the lemma.*

Using Lemma 3.1, we now propose an algorithm for determining an upper bound for the size of the largest DAN in an FCG. This is presented in Algorithm 1. In the beginning (Steps 1-3), a multilevel linked list, referred to as the Neighboring List (or *NList*, in short), is constructed from the given FCG. This special-purpose data structure contains, for every vertex, an ordered list of its neighboring vertices in the descending order of edge weights with respect to itself. Fig. 3(b) shows an example of configuration of the *NList* from an FCG whose fuzzy weight matrix is given in Fig. 3(a). For every vertex v_i in *NList*, the maximum number of vertices, $d(v_i)$, that can be associated with v_i to form an N -vertexlet having an Association Density of at least δ is computed (Steps 6-13). In Fig. 3(b), the N -vertexlets that are formed for every vertex using the aforesaid consideration are

demarcated by dotted lines. In Lemma (3.1), it is shown that $d(v_i)$ denotes the maximum size of the largest DAN that could be formed associating the vertex v_i . In the next step (Steps 12-14), the cumulative frequency of vertices with a degree value of at least k for all $k \in \{1, 2, \dots, |V| - 1\}$ is calculated. This is stored in $f(\geq k)$ which denotes the total number of vertices v_i that belongs to V such that $d(v_i) \geq k$. Lastly (Step 15), these values are used in the final step for determining an upper bound, K , of the size of the largest DAN.

Algorithm 1 Determining an upper bound for the size of the largest DAN

Input: An FCG $\tilde{G} = (V, \tilde{E}, \Omega)$ and an Association Density threshold δ .

Output: An upper bound K for the size of the largest DAN $V_{\ell et}^{N_{max}}$ of \tilde{G} .

Data Structure: A multilevel linked list $NList$ which stores the ordered list of neighbors for each vertex. $NList(v_m, n)$ denotes the n^{th} dense neighbor of vertex v_m .

Steps of the algorithm:

- 1: **for** each vertex $v_i \in V$ **do**
 - 2: Set $NList(v_i, n) \leftarrow v_t$, where $v_t \in V - \{v_i\}$, such that $\Omega_{v_i NList(v_i, j)} \leq \Omega_{v_i NList(v_i, k)}$, if $j < k$ and $NList(v_i, j) \neq NList(v_i, k)$, if $j \neq k, \forall n \in \{1, 2, \dots, |V| - 1\}$
 - 3: **end for**
 - 4: **for** each vertex $v_i \in V$ **do**
 - 5: **for** $k=1$ to $|V| - 1$ **do**
 - 6: **if** $\sum_{j=1}^k \Omega_{v_i NList(v_i, j)} < \delta k$ **then**
 - 7: $d(v_i) \leftarrow k - 1$
 - 8: Break out of the closer for loop and proceed to next v_i
 - 9: **end if**
 - 10: **end for**
 - 11: **end for**
 - 12: **for** $k=1$ to $|V| - 1$ **do**
 - 13: $f(\geq k) \leftarrow \#v_i$, such that $d(v_i) \geq k, \forall v_i \in V$
 - 14: **end for**
 - 15: $K \leftarrow \text{Max}_{\forall k \in \{0, 1, \dots, |V| - 1\}: f(\geq k) \geq k+1} \binom{k}{k} + 1$
-

Lemma 3.2 For an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, Algorithm 1 provides an upper bound for the size of the largest DAN that is present in \tilde{G} with respect to the Association Density threshold δ .

Proof 3 Lemma 3.2 is proved by contradiction. For an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, let us assume that there exists an N -vertexlet of size $K + 1$. Evidently, $f(\geq k)$ monotonically decreases with increase in k . So, it is deducible that,

$$\begin{aligned}
K + 1 &= \max_{\forall k \in \{0, 1, \dots, |V| - 1\}: f(\geq k) \geq k+1} \binom{k}{k} + 2 \\
&= \min_{\forall k \in \{0, 1, \dots, |V| - 1\}: f(\geq k) < k+1} \binom{k}{k} + 1.
\end{aligned} \tag{4}$$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}
v_1	1.00	0.67	0.52	0.62	0.55	0.69	0.63	0.69	0.69	0.71	0.72
v_2	0.67	1.00	0.59	0.69	0.69	0.70	0.62	0.70	0.64	0.69	0.72
v_3	0.52	0.59	1.00	0.65	0.64	0.60	0.72	0.67	0.60	0.67	0.75
v_4	0.62	0.69	0.65	1.00	0.74	0.58	0.63	0.71	0.66	0.61	0.75
v_5	0.55	0.69	0.64	0.74	1.00	0.75	0.68	0.63	0.69	0.68	0.74
v_6	0.69	0.70	0.60	0.58	0.75	1.00	0.66	0.68	0.71	0.73	0.75
v_7	0.63	0.62	0.72	0.63	0.68	0.66	1.00	0.72	0.68	0.72	0.76
v_8	0.69	0.70	0.67	0.71	0.63	0.68	0.72	1.00	0.76	0.76	0.82
v_9	0.69	0.64	0.60	0.66	0.69	0.71	0.68	0.76	1.00	0.76	0.75
v_{10}	0.71	0.69	0.67	0.61	0.68	0.73	0.72	0.76	0.76	1.00	0.75
v_{11}	0.72	0.72	0.75	0.75	0.74	0.75	0.76	0.82	0.75	0.75	1.00

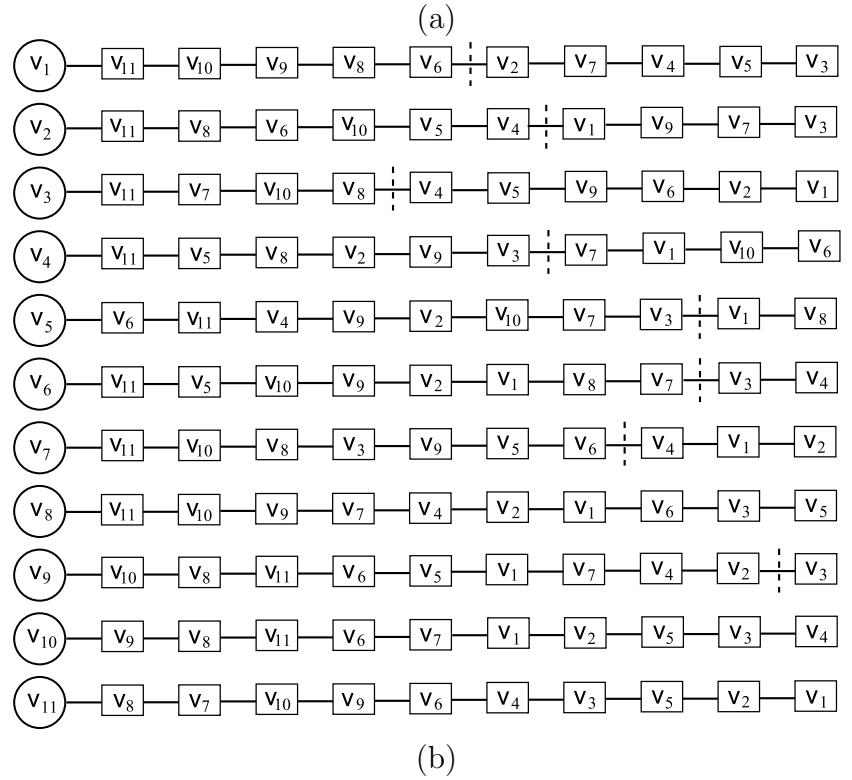


Figure 3: (a) The fuzzy weight matrix of an FCG and (b) Construction of $NList$ from the FCG represented in (a) w.r.t. $\delta = 0.7$.

This implies that, \tilde{G} may contain an N -vertexlet of size $(k + 1)$ such that $f(\geq k) < k + 1$. But, an N -vertexlet of size $(k + 1)$ should contain exactly $(k + 1)$ vertices from \tilde{G} where $d(v_i) \geq k$. Hence, $f(\geq k) < k + 1$ cannot be true. So, the assumption is not valid, which, in turn, proves the lemma.

Lemma 3.3 *Algorithm 1 runs within $O(n^2 \log n)$ time, where n denotes the number of vertices in the FCG.*

Proof 4 *The construction of the $NList$ involves, for every vertex, the ordering of the neighboring vertices that can be done within time $O(n \log n)$ (best average case time complexity of sorting). Again, for every vertex, the $NList$ can be scanned for the support of δ in time $O(n)$ for an average case. After that, the calculation of $f(\geq k)$, for all k , can be done in $O(n^2)$ time, and finally, K can be calculated in time $O(n)$. Thus, the total average case time complexity is,*

$$\begin{aligned} T(n) &= n * O(n \log n) + n * O(n) + O(n^2) + O(n) \\ &= O(n^2 \log n). \end{aligned}$$

The proposed method for solving the $MAX-DAN$ problem, in case of a scale-free FCG, is presented and discussed in detail in the following section. The complexity analysis is also provided.

4 A Solution Approach to the $MAX-DAN$ Problem

The $MAX-DAN$ problem, as the previous analysis highlights, falls into the NP complexity class resembling many other combinatorial optimization problems. Thus, modulo a basic complexity theoretic conjecture, $MAX-DAN$ cannot be solved exactly in polynomial time, unless $P=NP$ [9]. Therefore, a heuristic solution method for providing an approximate solution of the $MAX-DAN$ problem is proposed in Algorithm 2. In the beginning (Steps 1-2), the number of incident edges, to every vertex, having a weight of at least the threshold value δ is counted. The vertex set V is arranged in the ascending order of this count. It is highly expected that the largest DAN present in \tilde{G} will contain the vertices from the end of this ordered set in case of a scale-free graph. This is so, because in scale-free graphs new vertices preferentially attach with previously well connected vertices [3]. The next few steps (Steps 3-13) are similar to as it were in Algorithm 1 (Steps 1-11) where $NList$ is prepared and the maximum size of largest DAN that could be formed by every vertex is determined. The vertex with the maximum $d(v_i)$ value is used to initialize the largest DAN

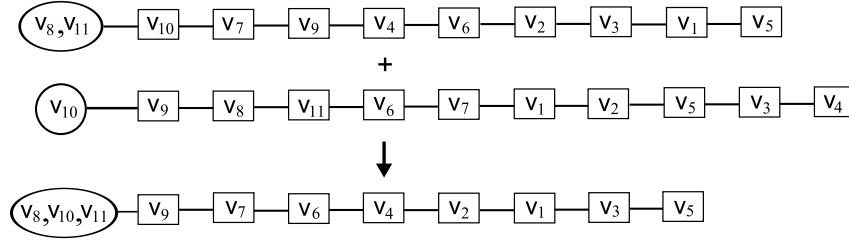


Figure 4: Updating the connector list of $V_{let}^{N_{max}}$ from \tilde{G} .

$V_{let}^{N_{max}}$ (step 14). Construction of the largest DAN then proceeds as in steps 16-25. To start with, a connector list is prepared from the list of neighboring vertices of v_{max} , taken from the $NList$. Then, a set V' , used temporarily, is computed as $V' = V_{let}^{N_{max}} \cup \{v_{first}\}$, where v_{first} is the first member of the connector list of $V_{let}^{N_{max}}$. It is then checked whether V' is densely associated. If not, then the process terminates and $V_{let}^{N_{max}}$ is returned as the largest DAN. Otherwise, $V_{let}^{N_{max}}$ is set to V' . Thereafter, the connector list of $V_{let}^{N_{max}}$ is updated by taking a weighted aggregation of the current connector list of $V_{let}^{N_{max}}$ and $NList[v_{first}]$, the row of $NList$ corresponding to v_{first} . An example is provided in Fig. 4. The iterative process continues till either all the vertices is included in $V_{let}^{N_{max}}$ or it terminates earlier in step 20. This methodology, which is referred hereafter as Maximum DAN Solver (in short, MaDSolver), is formally given in Algorithm 2.

Now, we present a detailed complexity analysis of the heuristic graph mining algorithm MaDSolver in the following subsection.

4.1 Complexity Analysis

Let, n be the number of vertices of the FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$, from which the largest DAN is required to be found out. Obviously, the number of edges in \tilde{G} will be n^2 , as it is essentially a complete graph. Since, the running time of MaDSolver is dependent on the size of the vertex set V , the time complexity will be computed in terms of the number of vertices, n , of the FCG. Initially, the ordering of vertices depending on the defined function C can be achieved in time $O(n \log n)$. In the next few steps, the $NList$ is constructed and $d(v_i)$ values are counted, for every vertex v_i , requiring (as shown in Lemma (3.3)) $O(n^2 \log n)$ running time. The seed vertex, v_{max} , is selected and the connector list is prepared in time $O(n)$. Finally, the iterative process for the construction of $V_{let}^{N_{max}}$ involves, at each iteration ($O(n)$), Association Density verification in time $O(n)$ (cleverly) and then, updating of the connector list of $V_{let}^{N_{max}}$ in time $O(n \log n)$, thereby claiming a total running time of $O(n) * (O(n) + O(n \log n))$. Therefore, MaDSolver involves

Algorithm 2 (MaDSolver) Heuristic Solution Method for MAX-DAN

Input: An FCG $\tilde{G} = (V, \tilde{E}, \Omega)$ and an Association Density threshold δ .

Output: The largest N-vertexlet $V_{\ell et}^{Nmax}$ in \tilde{G} satisfying the Association Density threshold δ .

Data Structure: A multilevel linked list $NList$ which stores the ordered list of neighbors for each vertex. $NList(v_m, n)$ denotes the n^{th} dense neighbor of vertex v_m .

Steps of the algorithm:

- 1: Define a cutoff weight function $C : \tilde{E} \rightarrow [0, 1]$, such that $C_{v_i v_j} = 1$, if $\Omega_{v_i v_j} \geq \delta$ and $C_{v_i v_j} = 0$, otherwise
 - 2: Arrange the vertices of \tilde{G} in the order $(v_1, v_2, v_3, \dots, v_{|V|})$, such that $\sum_{\forall v_k \in V, v_k \neq v_i} C_{v_i v_k} \leq \sum_{\forall v_k \in V, v_k \neq v_j} C_{v_j v_k}, \forall v_i, v_j \in V$ and $i < j$
 - 3: **for** each vertex $v_i \in V$ **do**
 - 4: Set $NList(v_i, n) \leftarrow v_t$, where $v_t \in V - \{v_i\}$, such that $\Omega_{v_i NList(v_i, j)} \geq \Omega_{v_i NList(v_i, k)}$, if $j < k$ and $NList(v_i, j) \neq NList(v_i, k)$, if $j \neq k, \forall n \in \{1, 2, \dots, |V| - 1\}$
 - 5: **end for**
 - 6: **for** each vertex $v_i \in V$ **do**
 - 7: **for** $k=1$ to $|V| - 1$ **do**
 - 8: **if** $\sum_{j=1}^k \Omega_{v_i NList(v_i, j)} < \delta k$ **then**
 - 9: $d(v_i) \leftarrow k - 1$.
 - 10: Break out of the inner for loop and proceed to next v_i
 - 11: **end if**
 - 12: **end for**
 - 13: **end for**
 - 14: $V_{\ell et}^{Nmax} \leftarrow \{v_{max}\}$, such that $d(v_{max}) \geq d(v_i), \forall v_i \in V$ /* In case of a tie select v_{max} randomly */
 - 15: $Connector(n) = NList(v_{max}, n), \forall n \in \{1, 2, \dots, |V| - 1\}$
 - 16: **while** $|V_{\ell et}^{Nmax}| < |V|$ **do**
 - 17: $V_{\ell et}^{Nmax} \leftarrow V_{\ell et}^{Nmax} \cup \{Connector(next)\}$, such that $next \leq n, \forall n \in [1, 2, \dots, |V| - 1] :$
 $Connector(n) \notin V_{\ell et}^{Nmax}$
 - 18: **if** $\mu_{V_{\ell et}^{Nmax}} < \delta$ **then**
 - 19: $V_{\ell et}^{Nmax} \leftarrow V_{\ell et}^{Nmax} - \{Connector(next)\}$
 - 20: Proceed to the end of the algorithm
 - 21: **else**
 - 22: $Order(v_i) = Index(Connector(n), v_i)(|V_{\ell et}^{Nmax}| - 1) + Index(NList(Connector(next), n), v_i), \forall v_i \in V$ /* $Index(V, v_i)$ denotes the position of the element v_i in set V */
 - 23: $Connector(n) \leftarrow v_t$, where $v_t \in V - V_{\ell et}^{Nmax}$, such that $Order(Connector(i)) \leq Order(Connector(j))$, for each $i < j$
 - 24: **end if**
 - 25: **end while**
-

a total worst case time complexity of,

$$\begin{aligned}
T(n) &= O(n \log n) + O(n^2 \log n) + O(n) + O(n) * (O(n) + O(n \log n)) \\
&= O(n \log n) + O(n^2 \log n) + O(n) + O(n^2 \log n) \\
&= O(n^2 \log n).
\end{aligned} \tag{5}$$

Now, we study the space complexity of MaDSolver which is also dependent on the size of the vertex set of the given FCG. It is clear that, the space complexity is independent of the size of the largest DAN, as we do not need to store them. Only the data structure *NList* (of size $n \times n$) is required to be kept permanently in memory during the execution of the algorithm. Additionally, $O(n)$ space is required for the solution set $V_{let}^{N_{max}}$ and $O(n)$ space is required for its connector list. Thus, the total space complexity of MaDSolver becomes $O(n^2)$.

4.2 Effectiveness of the Proposed Density Measure

In this section, we show theoretically that the proposed density measure is how much robust in the presence of noise vis-à-vis two existing density measures. Real-life networks, be it social, biological or the Internet, are constructed by quantifying the relationships between the nodes with edges. Usually, the process of quantification of these edge weights is prone to noise. As a result, false DANs are generated in such networks. We will explore here the sensitivity of the proposed density measure, following a probabilistic analysis, to the formation of false DANs in a *graph* because of noise. Since the existing density measures are normally used for crisp *graphs* [10] [29], the comparative analysis has been done on unweighted (crisp) *graphs*. Conventionally, two basic approaches are there to define the denseness of an N-vertexlet. The first one computes the ratio of the total number of edges appearing within an N-vertexlet to the maximum possible number of edges [10]. Improving this slightly, the second approach incorporates a minimum degree criterion in this computation [29]. We will first calculate the expected number of DANs that can be formed by noisy edges using the existing density measures and thereafter using the proposed measure.

Let p be the probability with which noisy edges occur independently in an unweighted *graph*, $G = (V, E)$. It is evident that the maximum possible number of edges within an N-vertexlet of G is given by $N(N - 1)/2$. According to the definition provided in [10], the density of an N-vertexlet reflects the frequency of edges within the set of vertices [10]. Thus, the probability of an N-vertexlet being dense as per the definition in [10], with

respect to a density threshold δ , is calculated as,

$$Pr(N, \delta, p) = \sum_{m=\lceil \delta N(N-1)/2 \rceil}^{N(N-1)/2} \binom{N(N-1)/2}{m} p^m (1-p)^{[N(N-1)/2]-m}. \quad (6)$$

In [29], a minimum degree criterion is attached with this density definition. Let, d be the minimum degree threshold for every vertex of an N-vertexlet. Then, the probability of an N-vertexlet being dense as per the definition in [29], with respect to density threshold δ and minimum degree threshold d , becomes,

$$Pr(N, \delta, d, p) = \left(\sum_{m=\lceil \delta N(N-1)/2 \rceil}^{N(N-1)/2} \binom{N(N-1)/2}{m} p^m (1-p)^{[N(N-1)/2]-m} \right) P(N, e, d). \quad (7)$$

In Eqn. 7, $P(N, e, d)$ represents the probability of an N-vertexlet, that has e number of edges, having minimum degree value of d . Now, let us turn our attention to the proposed density measure. In this case, every vertex is required to individually support the minimum Association Density threshold δ . So, every vertex should be connected to at least $\delta(N-1)$ vertices among the remaining $(N-1)$ vertices in the N-vertexlet. We first calculate the probability of a vertex v_i being dense in an N-vertexlet. It is calculated as,

$$P(v_i, N, \delta, p) = \sum_{m=\lceil \delta(N-1) \rceil}^{N-1} \binom{N-1}{m} p^m (1-p)^{N-m-1}. \quad (8)$$

Obviously, this probability value $P(v_i, N, \delta, p)$ (vide Eqn. 8) is the same for every vertex v_j being dense in an N-vertexlet. Therefore, the probability of the N-vertexlet, $V_{\ell et}^N$, itself being dense, with respect to δ , can be derived as,

$$Pr_{new}(N, \delta, p) = P\left(\bigcup_{v_j \in V_{\ell et}^N} v_j, N, \delta, p \right). \quad (9)$$

In Eqn. 9, $P(\bigcup_{v_j \in V_{\ell et}^N} v_j, N, \delta, p)$ represents the joint probability of all the vertices v_j being dense in $V_{\ell et}^N$. Eqn. 9, following the proposed density measure, produces a comparatively lower false positive rate of forming N-vertexlets than in Eqn. 6 (following [10]) and Eqn. 7 (following [29]) where $\delta < d$. If $\delta = d$, Eqn. 9 and Eqn. 7 yields equivalent probability values, while the first one becomes poorer. Finally, for the cases where $\delta > d$, Eqn. 7 becomes better than the remaining two measures. Another perspective in which $Pr_{new}(N, \delta, p)$ proves to be better is that it associates only one threshold parameter δ instead of two used in $Pr(N, \delta, d, p)$. The probabilistic errors occurring from multiple parameter tuning is, thus, less in the proposed approach. Most importantly, all the three aforesaid density

definitions significantly increase the chance of false DANs being generated with a slight growth of p .

Now let us consider the fuzzy *graphs*. Suppose, the set $\{\Omega_{v_i v_1}, \Omega_{v_i v_2}, \dots, \Omega_{v_i v_{N-1}}\}$ defines the edge weights from a vertex v_i to the remaining vertices $\{v_1, v_2, \dots, v_{N-1}\}$ within an N-vertexlet, $V_{\ell et}^N$, of an FCG, $\tilde{G} = (V, \tilde{E}, \Omega)$. Let the percentage of noise throughout the edges of $\tilde{G} = (V, \tilde{E}, \Omega)$ be q . Then, the increase in density of v_i due to noise effects is given by,

$$\frac{\sum_{j=1}^{N-1} \Omega_{v_i v_j} (1 + q) - \sum_{j=1}^{N-1} \Omega_{v_i v_j}}{N - 1} = \left(\mu_{v_i / V_{\ell et}^N} \right) q. \quad (10)$$

Here, $\mu_{v_i / V_{\ell et}^N}$ denotes the density of a vertex v_i with respect to an N-vertexlet $V_{\ell et}^N$. Thus, the probability of a vertex being dense in an N-vertexlet of an FCG due to noise probability q is simply q .

Now, we present the results of the empirical study in the following section.

5 Results

The efficacy of MaDSolver has been demonstrated by applying it on various datasets to mine the largest DAN. First, it has been used to mine the largest DAN from a number of synthetic datasets. Thereafter, it is used to extract the largest module of genes from two co-expression networks derived from microarray experiments. A set of C programs have been written in Linux platform for the experimental study. All the simulations have been performed on an HP xw8400 Workstation with Dual Core 3.0 GHz Intel Xeon processors, 4 MB Cache memory and having 2 GB primary memory. The study has been carried out on two types of datasets viz., Artificial datasets and Real-life (taken from biological microarray experiment) datasets. We have started from (or constructed in some cases) an FCG representing the similarity between the objects (vertices of a graph or genes) with the edge weights and mine it to locate the largest DAN. Notably, higher fuzzy edge weight denotes higher similarity between the objects in all kinds of networks considered in the study. MaDSolver not only determines the size of the largest DAN, but also produces a solution set. For artificial datasets, the size of the largest DAN found is reported. For real-life gene co-expression networks, the largest gene module mined using MaDSolver is reported and also biologically validated. The empirical details are given follows.

5.1 Artificial Datasets

We have constructed a number of artificial scale-free networks, with order (number of vertices), $O(\tilde{G})$, ranging from 100-1000, exclusively for this study. These networks, in the form of scale-free FCGs, which have been generated using a program written in MATLAB, follow power-law degree distribution with different values of γ . The statistical details of these synthetic FCGs are given in Table 1. A tolerance of 0.1 in the value of γ has been incorporated in order to bring randomness in this process while retaining the scale-free property of the network. The degree distribution of the finalized FCGs of all the networks generated is shown in Fig. 5, which reflects the scale-free property. While generating the artificial networks this has been kept in mind that the topology of a scale-free FCG depends on three important parameters, namely, δ , γ and $O(\tilde{G})$. So, various combinations of δ , γ and $O(\tilde{G})$ have been considered to produce the networks to verify the performance of MaDSolver over different patterns of scale-free networks. A wide range of Association Density threshold (δ) value within 0.5 and 0.9, which helps in varying the distribution of edge weights in the FCG, is used to construct the networks. In brief, first, a scale-free unweighted *graph* is constructed with respect to the parameters γ and $O(\tilde{G})$. Subsequently, this *graph* is transformed into an FCG by assigning edge weights to the absent edges randomly taking from $[0, \delta]$ and to the existing edges randomly taking from $[\delta, 1]$. Thus, the assignment of weights to the edges is random, but bounded by the threshold δ . The order of the artificial networks has also been taken from a large span to test the capability of MaDSolver in mining *graphs* of diverse sizes.

Dataset	$O(\tilde{G})$	% Edges satisfying $\Omega_{v_i v_j} \geq \delta$	Density	δ	γ
Scale-freeFCG_1	100	$\sim 6.1\%$	~ 0.38	0.7	1.5 ± 0.1
Scale-freeFCG_2	500	$\sim 0.4\%$	~ 0.35	0.7	2 ± 0.1
Scale-freeFCG_3	1000	$\sim 0.14\%$	~ 0.25	0.5	2.5 ± 0.1
Scale-freeFCG_4	1000	$\sim 0.12\%$	~ 0.35	0.7	2.5 ± 0.1

Table 1: Statistical details of the artificial dataset.

The largest DAN in these FCGs are mined using MaDSolver for different δ values. Their sizes are denoted by $\hat{\omega}(\tilde{G})$. We have determined the upper bound of $\tilde{\omega}(\tilde{G})$ using Algorithm 1, and then searched for the largest DAN using MaDSolver. The actual value of $\tilde{\omega}(\tilde{G})$ is found by permuting all the possible DANs and then comparing them. The results are provided in Table 2. Columns 3 and 5 show the value of actual $\tilde{\omega}(\tilde{G})$ and that produced by MaDSolver ($\hat{\omega}(\tilde{G})$) respectively. The entries under $\hat{\omega}(\tilde{G})$ represent the mean values obtained over 30 runs of the algorithm. The upper bounds of $\tilde{\omega}(\tilde{G})$ derived by

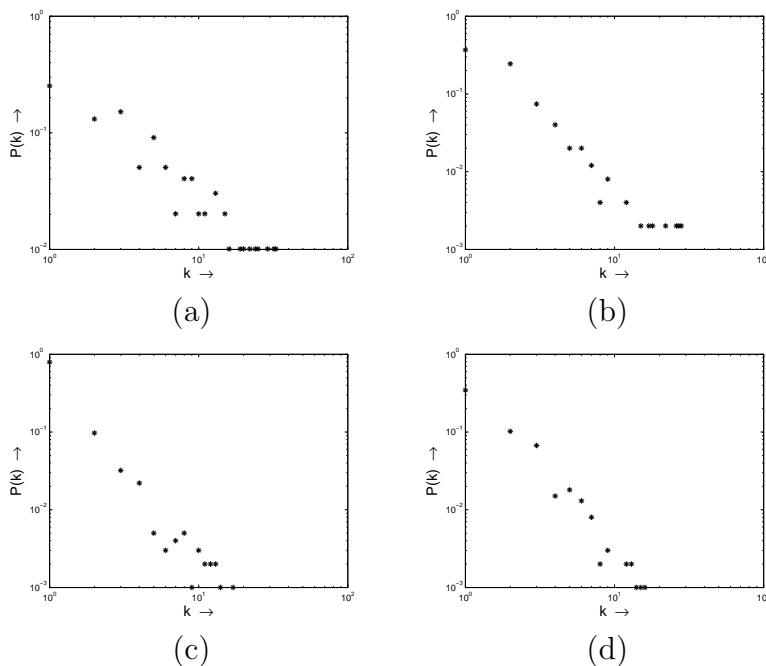


Figure 5: Degree distribution in (a) Scale-freeFCG_1 w.r.t. $\delta = 0.7$, (b) Scale-freeFCG_2 w.r.t. $\delta = 0.7$, (c) Scale-freeFCG_3 w.r.t. $\delta = 0.5$, and (d) Scale-freeFCG_4 w.r.t. $\delta = 0.7$.

Algorithm 1 is given under the fourth column. The δ values indicated within the brackets under the very first column indicate the values with respect to which the networks have been generated to be scale-free. Now, these networks have been taken to search for the largest DAN with respect to different δ value as shown in column 2. As can be seen from the fourth column, the derived upper bounds of $\tilde{\omega}(\tilde{G})$ produces an r -approximate result where the value of r increases with the expansion in the order of the graph. It is expected that MaDSolver will efficiently mine the largest DAN for the exact δ with which it has been generated to be scale-free. On the other hand, if this value is increased or decreased, disrupting the scale-free property of the network, the performance of MaDSolver is likely to degrade. The results from the synthetic scale-free networks produces supporting results. Again, for a very low δ (e.g., consider 0.2 in Table 2) the algorithm shows efficiency because the complete FCG itself becomes the largest DAN. For cases where $\tilde{\omega}(\tilde{G})$ is more than 30, the actual sizes are not computable due to super-polynomial time requirement. The results also show that the largest DAN size $\tilde{\omega}(\tilde{G})$ monotonically decreases with the increment of δ value as discussed in Lemma 3.2.

Dataset	δ	$\tilde{\omega}(G)$	Upper bound of $\tilde{\omega}(G)$ derived	$\hat{\omega}(G)$
Scale-freeFCG_1($\delta = 0.7$)	0.2	100	100	100
	0.3	100	100	100
	0.4	≥ 30	86	30
	0.5	25	64	23
	0.6	19	43	17
	0.7	16	28	16
	0.8	14	17	10
	0.9	5	11	4
Scale-freeFCG_2($\delta = 0.7$)	0.2	500	500	500
	0.3	500	500	500
	0.4	≥ 30	419	30
	0.5	19	288	17
	0.6	16	158	11
	0.7	13	43	12
	0.8	11	15	11
	0.9	5	9	3
Scale-freeFCG_3($\delta = 0.5$)	0.2	1000	1000	1000
	0.3	≥ 30	790	28
	0.4	18	416	14
	0.5	13	75	12
	0.6	11	20	11
	0.7	11	13	8
	0.8	6	9	3
	0.9	3	6	2.5 (Min=2, Max=3)
Scale-freeFCG_4($\delta = 0.7$)	0.2	1000	1000	1000
	0.3	1000	1000	1000
	0.4	≥ 38	840	38
	0.5	17	571	11
	0.6	13	301	12
	0.7	12	56	11
	0.8	10	14	7 (Min=5, Max=9)
	0.9	4	7	3

Table 2: Largest DANs mined from the artificial datasets.

5.2 Biological Datasets

Knowledge discovery from gene and protein interaction networks is a rapidly growing field. Using microarray technology, the expression of thousands of genes can be measured simultaneously [29]. Standard microarray dataset comprises of two dimensional array of expression values wherein the rows and columns correspond to genes and experiments, respectively. We have primarily used the well-known Eisen dataset (consisting of 79 Microarray experiments of 2467 yeast genes) [18] for mining the largest DAN from it. The percentage of missing values in the dataset is 1.93% (3760 missing values out of 194893) which have been estimated using the BPCA method [23] for greater accuracy in results. Although the estimation of missing values in microarray data is a debatable issue, a current study shows that the method is good enough for making the analysis more accurate [8]. Leave-One-Out Pearson (LOOP) correlation coefficient has been used to quantify the pairwise gene expression patterns in microarray data. Suppose, ρ_{ij}^n denotes the Pearson correlation coefficient between the genes i and j leaving out the n^{th} experiment (column). Then, the LOOP correlation coefficient between i and j for N experiments is given by $\rho_{ij}^{LOOP} = \min_{n=\{1,2,\dots,N\}}(\rho_{ij}^n)$. For the entire dataset, we have mapped the value of LOOP correlation coefficient ρ_{ij}^{LOOP} ($\rho_{ij}^{LOOP} \in [-1, 1]$) into a fuzzy similarity value (in the range $[0, 1]$) computed as $\frac{\rho_{ij}^{LOOP} + 1}{2}$ between each gene pair and these are used as the weights, denoting the proximities between the genes, of the FCG. With this fuzzy similarity metric the negative correlation is computed to be closer to 0 and the positive correlation closer to 1, whereas no correlation is computed as 0.5. We have verified that the gene correlation network follows the power-law degree distribution, and hence is scale-free, for higher density thresholds ($\delta \geq 0.7$). Table 3 shows the results found by applying MaDSolver on this FCG constructed from the Eisen dataset. The values of $\hat{\omega}(\tilde{G})$ derived by MaDSolver is given under column 3 by varying δ from 0.1 to 0.9. For each of the instances, 30 simulation runs were performed. It may be observed from Table 3 that, when δ is set high, each time only a single vertex attain the maximum degree value. In each simulation run, V_{let}^{Nmax} is initialized with this fixed vertex by MaDSolver producing unique results. But when δ goes low, multiple vertices can attain the maximum degree value. To break this tie, MaDSolver randomly initializes V_{let}^{Nmax} with one of these vertices, which, in turn, may result into different solutions over different simulation runs. Therefore, for lower δ values, the average value of the different results received over multiple simulations, is given. But again, for very lower values of δ the complete vertexset becomes the largest DAN and the solution becomes a unique one. Exceptionally, this variety in solutions may also be received for higher δ values for a different topology of the network where

scale-free property may distort as observed in Table 2 (setting $\delta = 0.9$ in Scale-freeFCG_3 and $\delta = 0.8$ in Scale-freeFCG_4) for some artificial networks. we may receive The upper bound of $\tilde{\omega}(\tilde{G})$ has also been derived using Algorithm 1 in each case for various δ values and is given in column 2. As expected, larger δ values result in smaller DANs. The results highlight that the upper bound derived by our algorithm is fairly good approximation on the actual upper bounds for such networks.

δ	Upper bound of $\tilde{\omega}(\tilde{G})$ derived	$\tilde{\omega}(\tilde{G})$
0.1	2467	2467
0.2	2467	2467
0.3	2467	2467
0.4	2467	2467
0.5	2134	1124.3 (Min=546, Max=1307)
0.6	1170	592
0.7	553	308
0.8	224	163
0.9	103	94

Table 3: Largest DANs found from Eisen dataset for different δ values.

The genes with same functional activity are likely to follow the same expression pattern. By observing the Expression graph of multiple genes, we can, therefore, estimate the level of similarity between them. An Expression graph plots the expression values of multiple genes simultaneously and an Eisen plot represents the expression values with different color levels in an image matrix. The Expression graph (Expression profile plot) and the Eisen plot [18] of the largest DANs found in Eisen data for different δ values obtained by MaDSolver are shown in Fig. 6. In these Expression graphs X-axis corresponds to the columns of the microarray data and the Y-axis represents the expression values obtained in the form $\log_2(R/G)$. The bold line in these Expression graphs represent the average of the expression values of all the genes over successive time points. In an Eisen plot X-axis corresponds to the expression values and Y-axis to the genes. It is evident from the graphs and plots that for higher δ values the Expression graphs and the Eisen plots reflect higher similarity pattern. This is expected since a higher δ value will result in mining vertexlets that are more dense, and therefore, the corresponding gene module will be highly similar.

We have tested the quality of the largest DAN found with respect to $\delta = 0.95$, of size 36, using Gene Ontology (GO) (The Gene Ontology Consortium, 2000). For this purpose, a web-based Gene Ontology tool FatiGO [6] has been utilized. FatiGO extracts

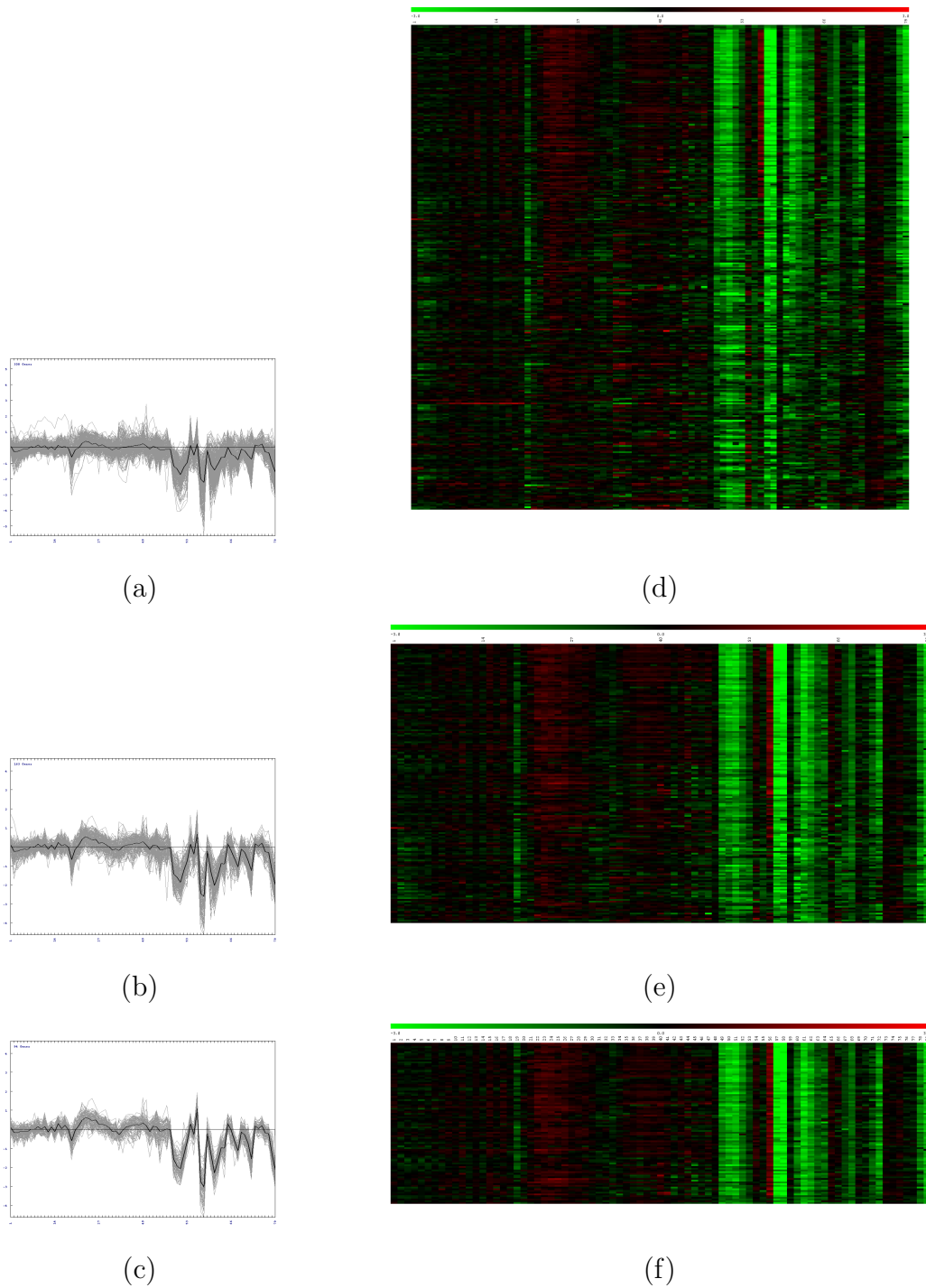


Figure 6: Expression graph of largest DAN found in Eisen dataset having (a) $\tilde{\omega}(\tilde{G}) = 308$ w.r.t. $\delta = 0.7$, (b) $\tilde{\omega}(\tilde{G}) = 163$ w.r.t. $\delta = 0.8$, and (c) $\tilde{\omega}(\tilde{G}) = 94$ w.r.t. $\delta = 0.9$. Eisen plot of largest DAN found in Eisen dataset having (d) $\tilde{\omega}(\tilde{G}) = 308$ w.r.t. $\delta = 0.7$, (e) $\tilde{\omega}(\tilde{G}) = 163$ w.r.t. $\delta = 0.8$, and (f) $\tilde{\omega}(\tilde{G}) = 94$ w.r.t. $\delta = 0.9$.

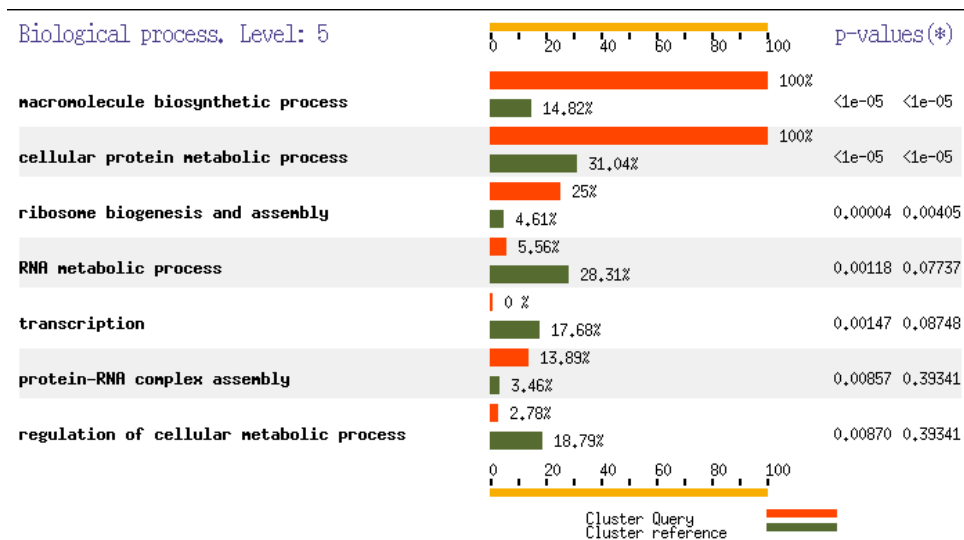


Figure 7: Gene Ontology biological process level 5 partial results.

the Gene Ontology terms for a query and a reference set of genes. In this experiment, a query is the set of genes forming the DAN whose biological relevance is to be measured. The remaining set of genes is taken as the reference set. The GO level is fixed at 5. The result obtained from the FatiGo tool for biological process is shown partially in Fig. 7. The complete list of significant subfunctions (with $p\text{-values} \leq 0.001$) in biological process, cellular component and molecular function found by FatiGo is given in Table 4. Adjusted p-values have been calculated using the false discovery rate (FDR) which is the expected number of false rejections among the rejected hypothesis. The third column indicates the percentage of genes found in the largest DAN that are functionally annotated with a particular subfunction. Under all the functions of the first hierarchy of Gene Ontology (i.e. biological process, cellular component and molecular function) the results show that there are many subfunctions wherein 100% of the genes found in the largest DAN are involved in similar functional activity. Moreover, very small p-values (as low as in the order of 10^{-27}) computed, side by side, for this annotation process indicates that this involvement of the genes in resembling functional activity is hardly expected to be random.

Since the problem of mining the largest DAN in an FCG has some similarity with that of clustering a *graph*, the quality of the largest DAN found using MaDSolver has been compared with the most dense clusters (of size ≥ 30 for a rational comparison with largest DANs) produced by three well-known clustering algorithms viz., K-Means, K-Medians and Hierarchical clustering (HCL) on the same dataset. The Expression graphs and Eisen plots of the most dense clusters derived by the said above algorithms are shown in Fig. 8. In order to compare the results with a recently proposed algorithm for clustering

Function	Subfunction	Annotation %	Adjusted p-value	Level
Biological process	Biosynthetic process	100%	2.211E-18	3
	Macromolecule metabolic process	100%	1.99E-5	3
	Cellular biosynthetic process	100%	1.468E-19	4
	Protein metabolic process	100%	5.886E-16	4
	Cellular macromolecule metabolic process	100%	6.673E-16	4
	Biopolymer metabolic process	5.56%	1.491E-5	4
	Macromolecule biosynthetic process	100%	5.029E-27	5
	Cellular protein metabolic process	100%	2.835E-16	5
Cellular component	Non-membrane-bound organelle	100%	<1E-5	3
	Membrane-bound organelle	11.11%	<1E-5	3
	Organelle part	100%	<1E-5	3
	Membrane	0%	8E-5	4
	Membrane part	0%	5.8E-4	5
	Ribonucleoprotein complex	100%	<1E-5	6
	Cytoplasm	100%	1E-5	6
	Intracellular non-membrane-bound organelle	100%	<1E-5	7
	Intracellular membrane-bound organelle	11.11%	<1E-5	7
	Intracellular organelle part	100%	<1E-5	7
Cytoplasmic part	100%	<1E-5	7	
Molecular function	rRNA binding	100%	4.518E-8	5

Table 4: Significant terms found by functional annotation of the largest DAN with respect to $\delta = 0.95$ having adjusted p-value $\leq 1E-3$.

microarray gene expression data we considered the weighted Chinese restaurant process based clustering method (CRC) [20] on the same dataset. The parameters have been set in CRC algorithm as: number of chains = 10, number of cycles = 20, inversion flag = 1, maximum shift = 2 and posterior probability threshold = 0.5, inspired by the default values [20]. The result, plotted using Expression graph and side by side Eisen plot for the most dense cluster found by CRC in the Eisen dataset [18], is also shown in Fig. 8. The details of the cluster sizes, which is received by applying these clustering methods, are given in Table 5.

Clustering Methods	# Iterations	# Clusters	Most dense cluster size
K-Means	500	50	93
K-Medians	500	50	73
HCL (weighted linkage)	-	50	162
CRC	-	44	41

Table 5: Most dense clusters determined by some well-known clustering algorithms.

As can be seen from Figs. 6 and 8, the Expression graph of the gene modules found by MaDSolver have better similarity in expression patterns as compared to those found by the other four existing clustering algorithms. On comparing the Expression graphs obtained by CRC (see Fig. 8(d)), with those obtained by the standard clustering algorithms (see Fig. 8(a), (b), and (c)) the former one is found to give superior results. The similar

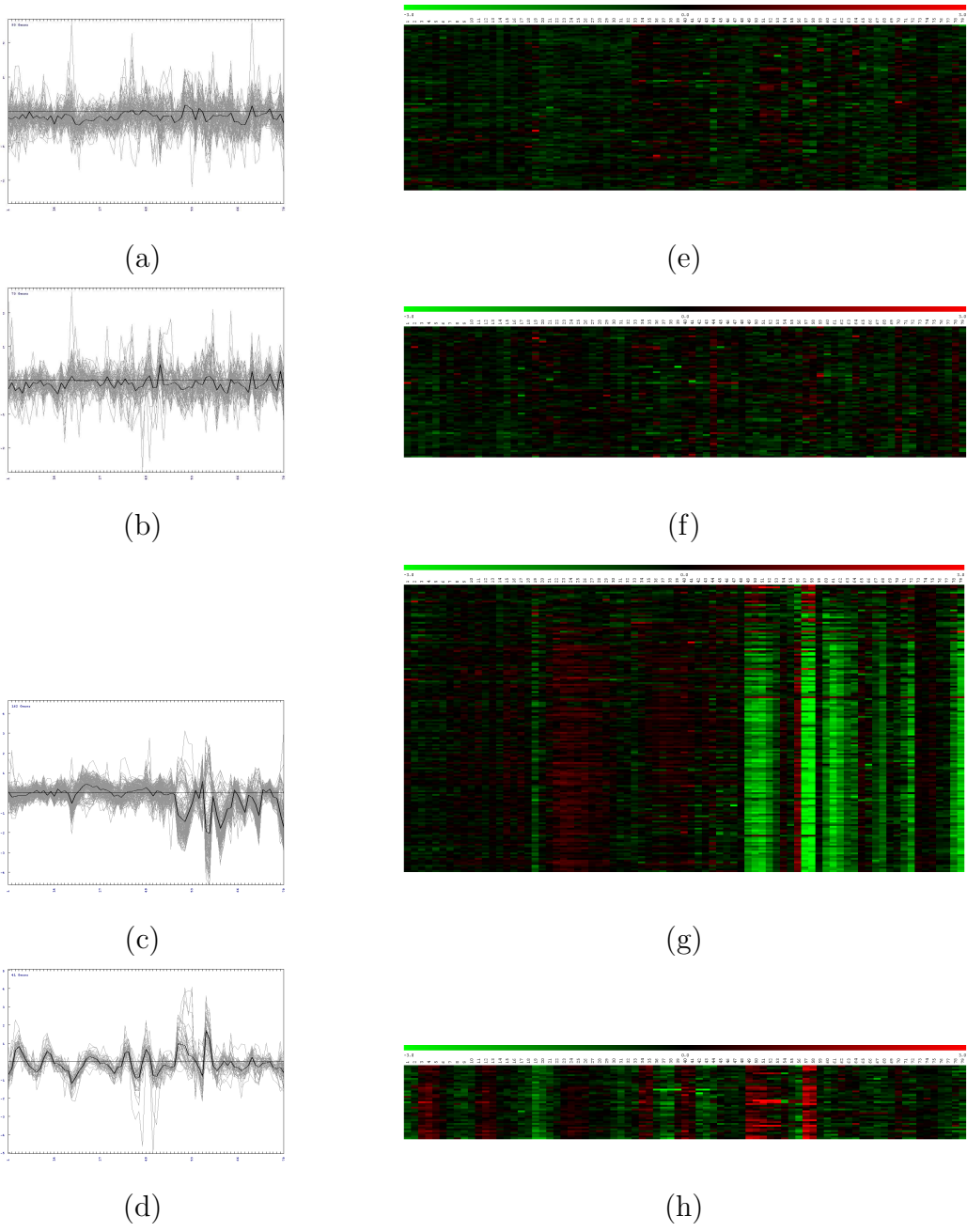


Figure 8: Expression graph of the most dense cluster found in Eisen dataset using (a) K-Means of size 93, (b) K-Medians of size 73, (c) HCL (weighted linkage) of size 162, and (d) CRC of size 41. Eisen plot of the most dense cluster found in Eisen dataset using (e) K-Means of size 93, (f) K-Medians of size 73, (g) HCL (weighted linkage) of size 162, and (h) CRC of size 41.

interpretations also come out from the results reflected through the Eisen plots where MaDSolver shows better expression level image pattern. While assessing the size of the clusters or N-vertexlets found by these methods, MaDSolver demonstrates its efficacy in maximizing the size of the dense part of a found. The overall results indicate the superiority of the proposed MaDSolver algorithm in evolving the largest dense portion of a graph with respect to certain predefined density threshold. Additionally, from the poor clustering results provided by the conventional algorithms like K-Means and K-Medians, it becomes clear that they are not at all suitable for clustering time series gene expression data. This is a supportive result to the well-known fact that the algorithms like K-Means or K-Medians are not efficient for clustering high-dimensional data points.

For quantitatively testing the significance of the DANs found by MaDSolver (for various δ) and the most dense clusters found by the aforesaid clustering methods, the Silhouette index [17], a well-known cluster validity index, has been used. Here, as the solution is not a set of clusters but a single DAN (cluster), so to compute the Silhouette index, we measure the degree of compactness within the vertices of the DAN and its separation from the set of other vertices outside the DAN (assuming it to be the second cluster). The distance metric used is the correlation coefficient. Thus, Silhouette index $SI_{C/V}$ of an N-vertexlet (cluster) C , with respect to the complete reference set V , is calculated as,

$$SI_{C/V} = \frac{1}{|C|} \sum_{v_i \in C} \frac{\sum_{v_j \in V-C} \frac{\|v_i - v_j\|}{|V-C|} - \sum_{v_j \in C - \{v_i\}} \frac{\|v_i - v_j\|}{|C-1|}}{\max\{\sum_{v_j \in C - \{v_i\}} \frac{\|v_i - v_j\|}{|C-1|}, \sum_{v_j \in V-C} \frac{\|v_i - v_j\|}{|V-C|}\}}. \quad (11)$$

The value of $SI_{C/V}$ ranges within $[-1,+1]$, with higher values indicating better clustering. We calculate this index of the most dense clusters found by some existing clustering methods and the N-vertexlets found by MaDSolver for various δ . The results are given in Table 6.

Algorithm	δ	Size of most dense cluster/ V_{let}^{Nmax}	$SI_{C/V}$
K-Means	-	93	0.1053
K-Medians	-	73	0.1647
HCL (weighted linkage)	-	162	0.5129
CRC	-	41	0.5663
MaDSolver	0.7	308	0.6095
MaDSolver	0.8	163	0.7781
MaDSolver	0.9	94	0.8827

Table 6: Silhouette index of the most dense N-vertexlets found by various algorithms from Eisen dataset.

The Silhouette index values, as earlier, indicate that the K-Means and K-Medians algorithms are not suitable for clustering this gene expression data. However, the other algorithms perform better in this regard. The performance of CRC is comparatively better than the other existing ones. The N-vertexlets found by MaDSolver for δ values ≥ 0.7 are better than those obtained using the other clustering approaches. Note that, in general, the N-vertexlets mined using MaDSolver are quite large in size, yet they still represent significantly coherent groups as revealed by the high $SI_{C/V}$ scores. Moreover, the performance of MaDSolver can be improved by tuning the parameter δ to higher values.

To demonstrate the effectiveness of MaDSolver on another real-life dataset, we considered the Human Fibroblasts Serum microarray data. The original dataset contains 13 columns (12 time points and 1 unsynchronized sample) against 8613 human genes [25]. A subset of 510 genes (a highly correlated subpart) and their corresponding expression levels through 12 synchronized time points (without any missing value) has been taken to construct the dataset for the study. As earlier, an FCG has been constructed that corresponds to the gene-gene co-expression over the microarray data. The largest DANs found in this FCG by MaDSolver for different δ values are shown in Table 7. The Expression graphs and Eisen plots corresponding to the largest DANs mined for $\delta = 0.7, 0.8,$ and 0.9 are shown in Fig. 9. Fig. 9 clearly highlights the similarity in expression pattern of the genes present in the largest DAN found by MaDSolver and it becomes more compact with an increasing δ .

δ	Upper bound of $\tilde{\omega}(\tilde{G})$ derived	$\hat{\tilde{\omega}}(\tilde{G})$
0.7	259	186
0.8	179	111
0.9	80	39

Table 7: Largest DAN found from Human Fibroblasts Serum microarray dataset for different δ .

The number of time points in the Serum dataset (12 columns) is quite low as compared to the Eisen dataset (79 columns) considered for the earlier study. The gene-gene correlation values are therefore expected to differ less in the FCG constructed from the Serum dataset. Still then, the result is very similar to the previous findings. The size of the DANs mined by MaDSolver become less and very compact with increase in the value of δ . For $\delta = 0.7$, MaDSolver identifies a gene set (N-vertexlet) of size 186 from the total 510 genes which is a good fraction of the entire network. The Expression graph and the Eisen plot (see Fig. 9(a) and Fig. 9(d)) of this N-vertexlet of size 186 still indicates a

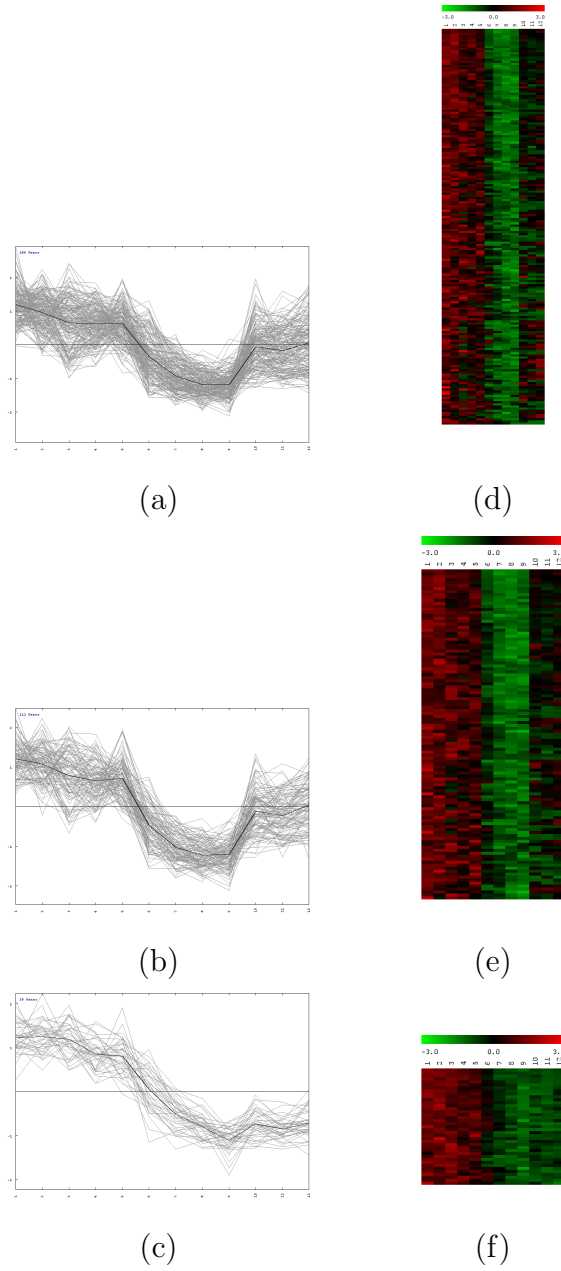


Figure 9: Expression graph of largest DAN found in Human Fibroblasts Serum dataset having (a) $\tilde{\omega}(\tilde{G}) = 186$ w.r.t. $\delta = 0.7$, (b) $\tilde{\omega}(\tilde{G}) = 111$ w.r.t. $\delta = 0.8$, and (c) $\tilde{\omega}(\tilde{G}) = 39$ w.r.t. $\delta = 0.9$. Eisen plot of largest DAN found in Serum dataset having (d) $\tilde{\omega}(\tilde{G}) = 186$ w.r.t. $\delta = 0.7$, (e) $\tilde{\omega}(\tilde{G}) = 111$ w.r.t. $\delta = 0.8$, and (f) $\tilde{\omega}(\tilde{G}) = 39$ w.r.t. $\delta = 0.9$.

high-quality similarity pattern in expression values.

As illustrated by the simulation results on the Eisen dataset, K-Means and K-Medians clustering methods are not suitable for clustering time series gene expression data. Therefore, while experimenting on the Serum data, these two algorithms have not been considered for comparison. The other two algorithms, showing good results for Eisen dataset, have been applied on the Serum dataset. The cluster size in HCL was set to be 30 and required parameters in CRC algorithm were set to be the default ones. Very short sized clusters (≤ 20) were not considered for a justified comparison with the MaDSolver algorithm which is efficient in mining very large groups. The Expression graphs and Eisen plots of the most dense clusters found by HCL and CRC in the Serum dataset is given in Fig. 10. On comparing Fig. 9 with Fig. 10, we found MaDSolver is superior to HCL and CRC in mining large groups of genes having resembling patterns which further improves for higher δ values. The compactness in the Expression graphs and coherency in the Eisen plots are clear indications of the dominance of MaDSolver over the other two.

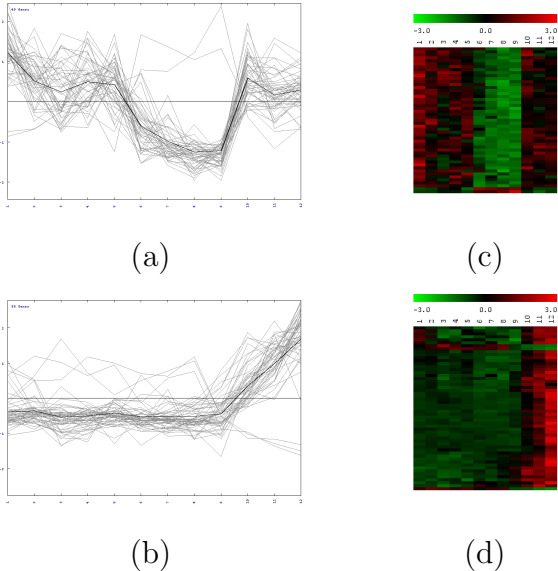


Figure 10: Expression graph of the most dense cluster found in Human Fibroblasts Serum dataset using (a) HCL (weighted linkage) of size 49 and (b) CRC of size 55. Eisen plot of the most dense cluster found in Serum dataset using (c) HCL (weighted linkage) of size 49 and (d) CRC of size 55.

The Silhouette index of these most dense clusters and the DANs found by MaDSolver for different δ values have been computed using Eqn. 11. The results are given in Table 8. On examining the Silhouette index values given under column 4, it becomes evident that MaDSolver outperforms the existing methods in locating dense parts of a graph satisfying

certain density threshold. Among the existing methods, CRC algorithm performs better than HCL. Obviously, the $SI_{C/V}$ values computed for DANs found for different δ indicate that the increase in δ value enhances the compactness in the N-vertexlet mined using MaDSolver.

Algorithm	δ	Size of Most dense cluster/ $V_{let}^{N_{max}}$	$SI_{C/V}$
HCL (weighted linkage)	-	49	0.6003
CRC	-	55	0.6798
MaDSolver	0.7	186	0.6911
MaDSolver	0.8	111	0.7543
MaDSolver	0.9	39	0.8616

Table 8: Silhouette index of the most dense N-vertexlets found by various algorithms from Serum dataset.

The experiments done on the Eisen and Serum dataset clearly indicates the efficiency of the proposed MaDSolver method in mining dense subgraphs of various sizes by proper tuning of δ .

6 Discussion and Conclusions

An efficient method for mining the largest dense N-vertexlet from an FCG, that follows scale-free property, with respect to a threshold value, is discussed in this paper. The density of an N-vertexlet is defined with respect to a minimum density threshold for each of the participating vertices. A novel approach has been used to define the denseness of a group of vertices in a *graph*. We have proved that the decision version of the original problem, *DAN*, falls in the NP-complete complexity class. While proving the results it has been established that *DAN* is at least as hard as the Clique Problem [5]. The problem addressed here is more general in nature being mapped on a fuzzy *graph* that is essentially a generalization of the crisp *graph*. However, our algorithm MaDSolver is designed for a special kind of graph, namely scale-free graph, whose vertices follow a power-law degree distribution. The time complexity of the algorithm has been established to be $O(n^2 \log n)$ and the space complexity is $O(n^2)$.

The initialization of the solution set $V_{let}^{N_{max}}$ has a significant role in the performance of MaDSolver. Note that, $V_{let}^{N_{max}}$ in \tilde{G} is initialized heuristically in this algorithm with v_{max} , since it has a higher probability of being included in the largest DAN. However, this might not always be the case. One possible solution of this may be taking every vertex to initialize $V_{let}^{N_{max}}$ in \tilde{G} in turn and computing the corresponding largest DAN.

From a probabilistic view, the chance of a vertex being in an N -vertexlet, taken randomly from a fully connected graph with M vertices, is $\frac{\binom{M-1}{N-1}}{\binom{M}{N}} = \frac{N}{M}$. Comparing all the DANs, so generated, a better solution can be achieved. Evidently, this will increase the time complexity from $O(n^2 \log n)$ to $O(n^3 \log n)$, while the space complexity will remain the same. Another significant phase of the proposed approach is the tuning of parameter δ for an arbitrary dataset. The selection of proper δ value to enhance the performance of knowledge mining is again an important task in this regard.

An in depth study of the working principle of MaDSolver shows that its performance is very dependent on the distribution of edge weights over the network. Consequently, for unweighted *graphs* the heuristic fails to keep track of the largest DAN. Therefore, when mining for the largest DAN in a scale-free unweighted *graph*, MaDSolver will not produce better results. This has been generally observed in an experimental analysis on twenty artificially generated unweighted networks [19]. The results improve only for higher δ (~ 1) values.

MaDSolver can be used to explore the transcriptional regulatory modules from the vast amount of microarray data accumulated in public repositories. Since the algorithm has limited polynomial time complexity, it can be used to locate large web clusters in gigantic networks like World Wide Web in real-time. It might also be used to extract knowledge of protein interactions from functional protein networks that could be beneficial in the advanced research areas like drug discovery.

References

- [1] Cliques, coloring, and satisfiability. In D. S. Johnson and M. A. Trick, editors, *Second DIMACS Implementation Challenge, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 26. American Mathematical Society, Providence, RI, 1996.
- [2] R. Albert A. L. Barabási and H. Jeong. Mean-field theory for scale-free random networks. *Physica A*, 272:173–187, 1999.
- [3] A. L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
- [4] P. R. J. Östegard. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120:197–207, 2002.

- [5] D. Z. Du and K. I. Ko. *Theory of Computational Complexity*. John Wiley Sons, Inc., New York, 2000.
- [6] R. Daz-Uriarte F. Al-Shahrour and J. Dopazo. Fatigo: a web tool for finding significant associations of gene ontology terms with groups of genes. *Bioinformatics*, 20:578–580, 2004.
- [7] T. Fahle. Simple and fast: Improving a branch-and-bound algorithm for maximum clique. In *Proceedings of the 10th Annual European Symposium on Algorithms*, pages 485–498, Kinsale, Ireland, 2002.
- [8] R. E. Blakesley M. J. Lotz G. N. Brock, J. R. Shaffer and G. C. Tseng. Which missing value imputation method to use in expression profiles: a comparative study and two selection schemes. *BMC Bioinformatics*, 9, 2008.
- [9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman Co., New York, 1979.
- [10] Y. Huang J. Han H. Hu, X. Yan and X. J. Zhou¹. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21:i213–i221, 2005.
- [11] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Academic Press, San Francisco, 2001.
- [12] A. T. Adai I. Lee, S. V. Date and E. M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306:1555–1558, 2004.
- [13] P. M. Pardalos I. M. Bomze, M. Budinich and M. Pelillo. The maximum clique problem. In D. Z. Du and P.M. Pardalos, editors, *Handbook of Combinatorial Optimization: Supplementary Volume A*, pages 1–74. Kluwer Academic, Dordrecht, 1999.
- [14] A. Schuster I. Sharfman and D. Keren. A geometric approach to monitoring threshold functions over distributed data streams. *ACM Transactions on Database Systems*, 32:23:1–23:29, 2007.
- [15] Z. Tang J. Wang and R. Wang. Maximum neural network with nonlinear self-feedback for maximum clique problem. *Neurocomputing*, 57:485–492, 2004.
- [16] A. Hamamoto K. Katayama and H. Narihisa. An effective local search for the maximum clique problem. *Information Processing Letters*, 95:503–511, 2005.

- [17] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, New York, 1990.
- [18] P. O. Brown M. B. Eisen, P. T. Spellman and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of National Academy of Sciences*, 95:14863–14868, 1998.
- [19] W. Sha P. Mendes and K. Ye. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19:ii122–ii129, 2003.
- [20] Z. S. Qin. Clustering microarray gene expression data using weighted chinese restaurant process. *Bioinformatics*, 22:1988–1997, 2006.
- [21] J. C. Régin. Solving the maximum clique problem with constraint programming. In *Proceedings of the CPAIOR'03*, pages 634–648, Kinsale, Ireland, 2003.
- [22] A. Rosenfeld. Fuzzy graphs. In K. Tanaka L.A. Zadeh, K.S. Fu and M. Shimura, editors, *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, pages 77–95. Academic Press, New York, 1975.
- [23] I. Takemasa M. Monden K. Matsubara S. Oba, M. Sato and S. Ishii. A bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19:2088–2096, 2003.
- [24] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.
- [25] D. T. Ross G. Schuler T. Moore J. C. F. Lee J. M. Trent L. M. Staudt J. Hudson Jr. M. S. Boguski D. Lashkari D. Shalon D. Botstein P. O. Brown V. R. Iyer, M. B. Eisen. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283:83–87, 1999.
- [26] D. Wood. An algorithm for finding maximum clique in a graph. *Operations Research Letters*, 21:211–217, 1997.
- [27] J. Xiao X. Geng, J. Xu and L. Pan. A simple simulated annealing algorithm for the maximum clique problem. *Information Sciences*, 177:5064–5071, 2007.
- [28] J. Ma X. Xu and J. Lei. An improved ant colony optimization for the maximum clique problem. In *Proceedings of the Third International Conference on Natural Computation*, volume 4, pages 766–770, Hainan, China, 2007.

- [29] Y. Huang M. S. Waterman P. S. Yu X. Yan, M. R. Mehan and X. J. Zhou. A graph-based approach to systematically reconstruct human transcriptional regulatory modules. *Bioinformatics*, 23:i577–i586, 2007.
- [30] R. Xu. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16:645–678, 2005.