Rough-Fuzzy Clustering for Grouping Functionally Similar Genes from Microarray Data

Pradipta Maji and Sushmita Paul

Abstract—Gene expression data clustering is one of the important tasks of functional genomics as it provides a powerful tool for studying functional relationships of genes in a biological process. Identifying co-expressed groups of genes represents the basic challenge in gene clustering problem. In this regard, a gene clustering algorithm, termed as robust rough-fuzzy c-means, is proposed judiciously integrating the merits of rough sets and fuzzy sets. While the concept of lower and upper approximations of rough sets deals with uncertainty, vagueness, and incompleteness in cluster definition, the integration of probabilistic and possibilistic memberships of fuzzy sets enables efficient handling of overlapping partitions in noisy environment. The concept of possibilistic lower bound and probabilistic boundary of a cluster, introduced in robust rough-fuzzy c-means, enables efficient selection of gene clusters. An efficient method is proposed to select initial prototypes of different gene clusters, which enables the proposed c-means algorithm to converge to an optimum or near optimum solutions and helps to discover co-expressed gene clusters. The effectiveness of the algorithm, along with a comparison with other algorithms, is demonstrated both qualitatively and quantitatively on fourteen yeast microarray data sets.

Index Terms—Microarray, gene clustering, overlapping clustering, rough sets, fuzzy sets

1 Introduction

Microarray technology is one of the important biotechnological means that has made it possible to simultaneously monitor the expression levels of thousands of genes during important biological processes and across collections of related samples [1], [2], [3]. An important application of microarray data is to elucidate the patterns hidden in gene expression data for an enhanced understanding of functional genomics.

A microarray gene expression data set can be represented by an expression table, where each row corresponds to one particular gene, each column to a sample or time point, and each entry of the matrix is the measured expression level of a particular gene in a sample or time point, respectively [1], [2], [3]. However, the large number of genes and the complexity of biological networks greatly increase the challenges of comprehending and interpreting the resulting mass of data, which often consists of millions of measurements. A first step toward addressing this challenge is the use of clustering techniques, which is essential in pattern recognition process to reveal natural structures and identify interesting patterns in the underlying data [4].

Cluster analysis is a technique for finding natural groups present in the gene set. It divides a given gene set into a set of clusters in such a way that two genes from the same cluster are as similar as possible and the genes from different clusters are as dissimilar as possible [2], [3]. To understand gene function, gene regulation, cellular processes, and subtypes of cells, clustering techniques have proven to be helpful. The co-expressed genes, that is, genes with similar expression patterns, can be clustered together with similar cellular functions. This approach may further understanding of the functions of many genes for which information has not been previously available [5], [6]. Furthermore, co-expressed genes in the same cluster are likely to be involved in the same cellular processes, and a strong correlation of expression patterns between those genes indicates co-regulation. Searching for common DNA sequences at the promoter regions of genes within the same cluster allows regulatory motifs specific to each gene cluster to be identified and cis-regulatory elements to be proposed [6], [7]. The inference of regulation through gene expression data clustering also gives rise to hypotheses regarding the mechanism of transcriptional regulatory network [8].

The purpose of gene clustering is to group together co-expressed genes which indicate co-function and co-regulation. Due to the special characteristics of gene expression data, and the particular requirements from the biological domain, gene clustering presents several new challenges and is still an open problem. The cluster analysis is typically the first step in data mining and knowledge discovery. The purpose of clustering gene expression data is to reveal the natural data structures and gain some initial insights regarding data distribution. Therefore, a good clustering algorithm should depend as little as possible on prior knowledge, which is usually not available before cluster analysis.

Different clustering techniques such as hierarchical clustering [9], k-means algorithm [10], self organizing map [11], graph theoretical approaches [12], [13], [14], [15], model based clustering [16], [17], [18], [19], and density based approach [20] have been widely applied to find groups of co-expressed genes from microarray data. A comprehensive survey on various gene clustering

algorithms can be found in [4], [7].

One of the main problems in gene expression data analysis is uncertainty. Some of the sources of this uncertainty include incompleteness and vagueness in cluster definitions. Also, the empirical study has demonstrated that gene expression data are often highly connected, and the clusters may be highly overlapping with each other or even embedded one in another [20]. Therefore, gene clustering algorithms should be able to effectively handle this situation. Moreover, gene expression data often contains a huge amount of noise due to the complex procedures of microarray experiments [21]. Hence, clustering algorithms for gene expression data should be capable of extracting useful information from a high level of background noise.

In this background, the possibility concept introduced by fuzzy set theory [22] and rough set theory [23] have gained popularity in modeling and propagating uncertainty. Both fuzzy sets and rough sets provide a mathematical framework to capture uncertainties associated with the data. One of the most notable prototype based partitional clustering algorithms is fuzzy c-means [24], [25]. It assigns each gene to every cluster by allowing gradual memberships. In effect, it offers the opportunity to deal with the data that belong to more than one cluster at the same time. It assigns memberships to a gene which are inversely related to the relative distance of the gene to cluster prototypes. Also, it can deal with the uncertainties arising from overlapping cluster boundaries and reveal additional information concerning gene co-expression [26], [27], [28], [29]. In particular, information regarding overlapping clusters and overlapping cellular pathways has been identified from fuzzy clustering results [29], [30]. However, the resulting membership values of fuzzy c-means do not always correspond well to the degrees of belonging of the data, and it may be inaccurate in a noisy environment [31]. To reduce this weakness and to produce memberships that have a good explanation of the degrees of belonging for the data, Krishnapuram and Keller [31] proposed possibilistic c-means algorithm. However, it sometimes generates coincident clusters [3].

Integrating the merits of rough sets and fuzzy sets, different rough-fuzzy clustering algorithms such as rough-fuzzy c-means [32], rough-possibilistic c-means [33], and rough-fuzzy-possibilistic c-means [33] have been proposed, where each cluster is represented by a cluster prototype, a crisp lower approximation and a probabilistic and/or possibilistic fuzzy boundary. The cluster prototype is computed based on the weighted average of crisp lower approximation and fuzzy boundary. All these algorithms can be used for clustering co-expressed genes from microarray gene expression data sets [3]. However, the crisp lower approximation of a gene cluster in rough-fuzzy clustering [32], [33] is usually assumed to be spherical in shape, which restricts to find arbitrary shapes of gene clusters. Recently, fuzzy-rough supervised gene clustering algorithm is proposed in [34] to find groups of co-regulated genes whose collective expression is strongly associated with sample categories.

In this paper, a rough-fuzzy clustering algorithm, termed as robust rough-fuzzy c-means (rRFCM), is proposed for clustering functionally similar genes from microarray gene expression data sets. It integrates judiciously the merits of rough sets, and probabilistic and possibilistic memberships of fuzzy sets. While the integration of both membership functions of fuzzy sets enables efficient handling of overlapping partitions in noisy environment, the concept of lower and upper approximations of rough sets deals with uncertainty, vagueness, and incompleteness in cluster definition. Each cluster is represented by a set of three parameters, namely, a cluster prototype or centroid, a possibilistic lower approximation, and a probabilistic boundary. The cluster prototype depends on the weighting average of the possibilistic lower approximation and probabilistic boundary. An efficient method is proposed to select initial prototypes of different gene clusters; thereby circumventing the initialization and local minima problems of c-means algorithm. The effectiveness of the proposed algorithm, along with a comparison with other gene clustering algorithms, is demonstrated on a set of fourteen microarray gene expression data sets using some standard validity indices.

The rest of this paper is organized as follows: Section 2 presents a new clustering algorithm, termed as robust rough-fuzzy c-means (rRFCM), based on the theory of rough sets and fuzzy sets. Implementation details and a brief description of different microarray gene expression data sets are reported in Section 3. Experimental results and a comparison among several gene clustering algorithms are presented in Section 4. Section 5 reports the biological significance of generated gene clusters. Concluding remarks are given in Section 6.

2 Robust RFCM Algorithm

This section introduces a new c-means algorithm, termed as robust rough-fuzzy c-means. The proposed c-means adds the concepts of fuzzy memberships, both probabilistic and possibilistic, of fuzzy sets, and lower and upper approximations of rough sets into c-means algorithm. While the integration of both probabilistic and possibilistic memberships of fuzzy sets enables efficient handling of overlapping clusters in noisy environment, the rough sets deal with uncertainty, vagueness, and incompleteness in cluster definition.

2.1 Objective Function

Let $X = \{x_1, \cdots, x_j, \cdots, x_n\}$ be the set of n objects and $V = \{v_1, \cdots, v_i, \cdots, v_c\}$ be the set of c centroids, where $x_j \in \Re^m$ and $v_i \in \Re^m$. Each of the clusters β_i is represented by a cluster center v_i, a lower approximation $\underline{A}(\beta_i)$ and a boundary region $B(\beta_i) = \{\overline{A}(\beta_i) \setminus \underline{A}(\beta_i)\}$, where $\overline{A}(\beta_i)$ denotes the upper approximation of cluster

β_i. The proposed c-means algorithm partitions X into c clusters by minimizing following objective function:

$$J = \begin{cases} w\mathcal{A}_1 + (1-w)\mathcal{B}_1 & \text{if } \underline{A}(\beta_i) \neq \emptyset,\ B(\beta_i) \neq \emptyset \\ \mathcal{A}_1 & \text{if } \underline{A}(\beta_i) \neq \emptyset,\ B(\beta_i) = \emptyset \\ \mathcal{B}_1 & \text{if } \underline{A}(\beta_i) = \emptyset,\ B(\beta_i) \neq \emptyset \end{cases} \quad (1)$$

$$\text{where } \mathcal{A}_1 = \sum_{i=1}^{c} \sum_{x_j \in \underline{A}(\beta_i)} (\nu_{ij})^{\acute{m}_2} ||x_j - v_i||^2 + \sum_{i=1}^{c} \eta_i \sum_{x_j \in \underline{A}(\beta_i)} (1-\nu_{ij})^{\acute{m}_2};$$

$$\text{and } \mathcal{B}_1 = \sum_{i=1}^{c} \sum_{x_j \in B(\beta_i)} (\mu_{ij})^{\acute{m}_1} ||x_j - v_i||^2.$$

The parameters w and $(1-w)$ correspond to the relative importance of lower and boundary regions, while $1 \leq \acute{m}_1 < \infty$ and $1 \leq \acute{m}_2 < \infty$ are the probabilistic and possibilistic fuzzifiers, respectively. Note that $\mu_{ij} \in [0,1]$ is the probabilistic membership function as that in fuzzy c-means and $\nu_{ij} \in [0,1]$ represents the possibilistic membership function that has the same interpretation of typicality as in possibilistic c-means.

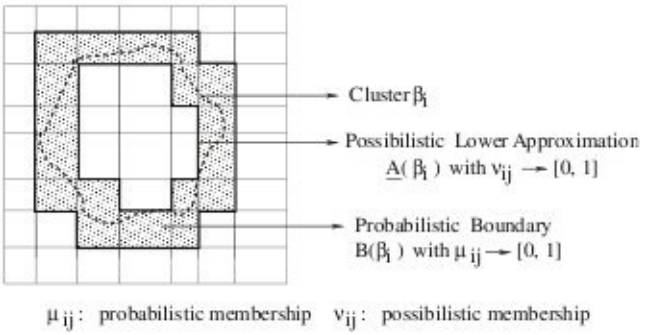


Fig. 1. Robust RFCM: cluster β_i is represented by possibilistic lower approximation and probabilistic boundary

In robust rough-fuzzy c-means, each cluster is represented by a centroid, a possibilistic lower approximation, and a probabilistic boundary (Fig. 1). The lower approximation influences the fuzziness of final partition. According to the definitions of lower approximation and boundary of rough sets [23], if an object $x_j \in \underline{A}(\beta_i)$, then $x_j \notin \underline{A}(\beta_k), \forall k \neq i$, and $x_j \notin B(\beta_i), \forall i$. That is, the object x_j is contained in β_i definitely. Hence, the memberships of the objects in lower approximation of a cluster should be independent of other centroids and clusters. Also, the objects in lower approximation should have different influence on the corresponding centroid and cluster. From the standpoint of "compatibility with the cluster prototype", the membership of an object in the lower approximation of a cluster should be determined solely by how far it is from the prototype of the cluster, and should not be coupled with its location with respect to other clusters. As the possibilistic membership ν_{ij} depends only on the distance of object x_j from cluster β_i, it allows optimal membership solutions to lie in the entire unit hypercube rather than restricting them to the hyperplane given by (3). Whereas, if $x_j \in B(\beta_i)$, then the object x_j possibly belongs to cluster β_i and potentially belongs to another cluster. Hence, the objects in boundary regions should have different influence on the centroids and clusters, and their memberships should depend on the positions of all cluster centroids. So, in robust rough-fuzzy c-means, the membership values of objects in lower approximation are identical to possibilistic c-means [(4)], while those in boundary region are the same as fuzzy c-means [(2)].

2.2 Membership Function

Solving (1) with respect to μ_{ij} and ν_{ij}, we get

$$\mu_{ij} = \left[\sum_{k=1}^{c} \left(\frac{||x_j - v_i||^2}{||x_j - v_k||^2} \right)^{\frac{1}{\acute{m}_1 - 1}} \right]^{-1}; \quad (2)$$

$$\text{subject to } \sum_{i=1}^{c} \mu_{ij} = 1, \forall j, \text{ and } 0 < \sum_{j=1}^{n} \mu_{ij} < n, \forall i, \quad (3)$$

$$\nu_{ij} = \left[1 + \left\{ \frac{||x_j - v_i||^2}{\eta_i} \right\}^{\frac{1}{(\acute{m}_2 - 1)}} \right]^{-1}; \quad (4)$$

$$\text{subject to } 0 < \sum_{j=1}^{n} \nu_{ij} \leq n, \forall i; \text{ and } \max_i \nu_{ij} > 0, \forall j; \quad (5)$$

where the scale parameter η_i is given by

$$\eta_i = \mathrm{K} \cdot \frac{\sum_{j=1}^{n} (\nu_{ij})^{\acute{m}_2} ||x_j - v_i||^2}{\sum_{j=1}^{n} (\nu_{ij})^{\acute{m}_2}}. \quad (6)$$

which represents the zone of influence or size of the cluster β_i. Typically K is chosen to be 1. Combining (2) and (4), the following relation can be established between the probabilistic and possibilistic memberships of object x_j in cluster β_i:

$$\mu_{ij} = \left[\sum_{k=1}^{c} \left\{ \frac{\eta_i}{\eta_k} \left(\frac{\nu_{kj}(1-\nu_{ij})}{\nu_{ij}(1-\nu_{kj})} \right)^{\acute{m}_2 - 1} \right\}^{\frac{1}{\acute{m}_1 - 1}} \right]^{-1}. \quad (7)$$

From the above relation, it can be seen that the probabilistic membership μ_{ij} is directly proportional to the possibilistic membership ν_{ij} of object x_j in cluster β_i.

2.3 Cluster Prototypes

The new centroid is calculated based on the weighting average of the possibilistic lower approximation and probabilistic boundary. Computation of the centroid is modified to include the effects of both fuzzy memberships, probabilistic and possibilistic, and lower and

upper bounds. The modified centroid calculation for robust rough-fuzzy c-means is obtained by solving (1) with respect to v_i:

$$v_i = \begin{cases} w\mathcal{C}_1 + (1-w)\mathcal{D}_1 & \text{if } \underline{A}(\beta_i) \neq \emptyset,\ B(\beta_i) \neq \emptyset \\ \mathcal{C}_1 & \text{if } \underline{A}(\beta_i) \neq \emptyset,\ B(\beta_i) = \emptyset \\ \mathcal{D}_1 & \text{if } \underline{A}(\beta_i) = \emptyset,\ B(\beta_i) \neq \emptyset \end{cases} \quad (8)$$

where $\mathcal{C}_1 = \dfrac{\sum_{x_j \in \underline{A}(\beta_i)} (\nu_{ij})^{\acute{m}_2} x_j}{\sum_{x_j \in \underline{A}(\beta_i)} (\nu_{ij})^{\acute{m}_2}}$; $\mathcal{D}_1 = \dfrac{\sum_{x_j \in B(\beta_i)} (\mu_{ij})^{\acute{m}_1} x_j}{\sum_{x_j \in B(\beta_i)} (\mu_{ij})^{\acute{m}_1}}$.

Hence, the cluster prototypes or centroids depend on the parameters w and $(1-w)$, and fuzzifiers $\acute{m}_1$ and $\acute{m}_2$ rule their relative influence.

2.4 Details of the Algorithm

Approximate optimization of the objective function J [(1)] by the robust rough-fuzzy c-means is based on Picard iteration through (2), (4), and (8). This type of iteration is called alternating optimization.

The process starts by choosing c objects as the initial centroids of the c clusters. The possibilistic memberships of all the objects are calculated using (4). The scale parameters η_i for c clusters are obtained using (6). Let $\nu_i = (\nu_{i1}, \cdots, \nu_{ij}, \cdots, \nu_{in})$ represents the possibilistic cluster β_i associated with the centroid v_i. After computing ν_{ij} for c clusters and n objects, the values of ν_{ij} for each object x_j are sorted and the difference of two highest memberships of x_j is compared with a threshold value δ_1. Let ν_{ij} and ν_{kj} be the highest and second highest memberships of x_j. If $(\nu_{ij} - \nu_{kj}) > \delta_1$, then $x_j \in \underline{A}(\beta_i)$, otherwise $x_j \in B(\beta_i)$ and $x_j \in B(\beta_k)$ if $\nu_{ij} > \delta_2$. After assigning each object in lower approximations or boundary regions of different clusters based on the thresholds δ_1 and δ_2, the probabilistic memberships μ_{ij} for the objects lying in the boundary regions are computed from the possibilistic memberships ν_{ij} using (7). The new centroids of different clusters are computed as per (8). The main steps of the robust rough-fuzzy c-means algorithm proceed as follows:

1) Assign initial centroids v_i, $i = 1, 2, \cdots, c$. Choose values for fuzzifiers $\acute{m}_1$ and $\acute{m}_2$, and calculate thresholds δ_1 and δ_2 . Set iteration counter $t = 1$.
2) Compute ν_{ij} by (4) for c clusters and n objects.
3) If ν_{ij} and ν_{kj} be the highest and second highest possibilistic memberships of object x_j and $(\nu_{ij} - \nu_{kj}) > \delta_1$ then $x_j \in \underline{A}(\beta_i)$. In addition, by properties of rough sets, $x_j \in \overline{A}(\beta_i)$.
4) Otherwise, $x_j \in B(\beta_i)$ and $x_j \in B(\beta_k)$ if $\nu_{ij} > \delta_2$. Furthermore, x_j is not part of any lower bound.
5) Compute μ_{ij} for the objects lying in boundary regions for c clusters using (7).
6) Compute new centroid as per (8).
7) Repeat steps 2 to 6, by incrementing t, until no more new assignments can be made.

In effect, the proposed algorithm has an overall computational complexity of $\mathcal{O}(tcnm)$, where t, c, n, and m represent the number of iteration, clusters, objects, and dimension of each object, respectively.

2.5 Selection of Parameters

The parameter w has an influence on the performance of robust rough-fuzzy c-means algorithm. Since the genes lying in lower approximation definitely belong to a cluster, they are assigned a higher weight w compared to $(1-w)$ of the genes lying in boundary regions. On the other hand, the performance of proposed c-means significantly reduces when $w \simeq 1.0$. In this case, since the clusters cannot see the genes of boundary regions, the mobility of the clusters and the centroids reduces. As a result, some centroids get stuck in local optimum. Hence, to have the clusters and the centroids a greater degree of freedom to move, $0 < (1-w) < w < 1$.

The performance of robust rough-fuzzy c-means also depends on the values of two thresholds δ_1 and δ_2, which determine the cluster labels of all the genes. In other word, the robust rough-fuzzy c-means partitions the data set into two classes, namely, lower approximation and boundary, based on the values of δ_1 and δ_2. The thresholds δ_1 and δ_2 control the size of granules of rough-fuzzy clustering. In practice, the following definitions work well:

$$\delta_1 = \frac{1}{n}\sum_{j=1}^{n}(\nu_{ij} - \nu_{kj}) \quad (9)$$

where n is the total number of genes, ν_{ij} and ν_{kj} are the highest and second highest memberships of object x_j. That is, the value of δ_1 represents the average difference of two highest possibilistic memberships of all the genes in the data set. A good clustering procedure should make the value of δ_1 as high as possible. On the other hand, the genes with $(\nu_{ij} - \nu_{kj}) \leq \delta_1$ are used to calculate the threshold δ_2:

$$\delta_2 = \frac{1}{\acute{n}}\sum_{j=1}^{\acute{n}}\nu_{ij} \quad (10)$$

where $\acute{n}$ is the number of genes those do not belong to lower approximations of any cluster and ν_{ij} is the highest membership of gene x_j. That is, the value of δ_2 represents the average of highest memberships of $\acute{n}$ genes in the data set.

2.6 Selection of Initial Cluster Prototypes

A limitation of the c-means algorithm is that it can only achieve a local optimum solution that depends on the initial choice of the cluster prototypes. Consequently, computing resources may be wasted in that some initial centers get stuck in regions of the input space with a scarcity of data points and may therefore never have the chance to move to new locations where they are needed. To overcome this limitation of the c-means algorithm, next a method is proposed to select initial cluster

prototypes, which is based on a similarity measure. It enables the algorithm to converge to an optimum or near optimum solutions or cluster centers.

Prior to describe the proposed method for selecting initial cluster centers, next a quantitative measure, called degree of similarity, is defined to evaluate the similarity between two genes.

Definition 1: The degree of similarity (DOS) between two objects x_i and x_j is defined as

$$\mathrm{DOS}(x_i, x_j) = \frac{1}{m}\sum_{k=1}^{m}\left[1 - \frac{|x_{ik} - x_{jk}|}{|k_{max} - k_{min}|}\right] \quad (11)$$

where m is the number of features of the object x_i, k_{max} and k_{min} denote the maximum and minimum values along the kth feature, respectively. The DOS is used to quantify the similarity between pairs of genes. If expression values of two genes are different, the DOS between them is small. A high value of $\mathrm{DOS}(x_i, x_j)$ between two genes x_i and x_j asserts that they may have similar expression patterns and are likely to be involved in same biological process. If two genes are same, the DOS between them is maximum, that is, $\mathrm{DOS}(x_i, x_i) = 1$. Hence, $0 \leq \mathrm{DOS}(x_i, x_j) \leq 1$. Also, $\mathrm{DOS}(x_i, x_j) = \mathrm{DOS}(x_j, x_i)$.

Based on the concept of degree of similarity, next a method is described for selecting initial prototypes. The main steps of this method proceed as follows:

1) For each gene x_i, calculate $\mathrm{DOS}(x_i, x_j)$ between itself and the gene x_j, $\forall_{j=1}^{n}$.
2) Calculate similarity score between genes x_i and x_j

$$\mathrm{S}(x_i, x_j) = \begin{cases} 1 & \text{if } \mathrm{DOS}(x_i, x_j) > \lambda \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

   where $0.5 \leq \lambda \leq 1$.
3) For each gene x_i, calculate total number of similar genes of x_i as

$$\mathrm{N}(x_i) = \sum_{j=1}^{n}\mathrm{S}(x_i, x_j). \quad (13)$$

4) Sort n genes according to their values of $\mathrm{N}(x_i)$ such that $\mathrm{N}(x_1) > \mathrm{N}(x_2) > \cdots > \mathrm{N}(x_n)$.
5) If $\mathrm{N}(x_i) > \mathrm{N}(x_j)$ and $\mathrm{DOS}(x_i, x_j) > \lambda$, then x_j cannot be considered as a initial cluster center, resulting in a reduced set of genes to be considered for initial cluster centers.

Finally, c initial centers are selected from the reduced set as potential initial centers. The main motive of introducing this initialization method lies in identifying different dense regions present in the data set. The identified dense regions ultimately lead to discovering natural groups present in the data set. The whole approach is, therefore, data dependent.

3 Experimental Setup

In the present research work, the performance of the proposed robust rough-fuzzy c-means (rRFCM) algorithm is compared with that of hard c-means (HCM) [10], fuzzy c-means (FCM) [27], rough-fuzzy c-means (RFCM) [32], cluster identification via connectivity kernels (CLICK) [14], and self organizing map (SOM) [11] on several microarray gene expression data sets. The major metrics for evaluating the performance of different algorithms are Silhouette index [35], Davies-Bouldin index [36], Dunn index [36], β index [37], Eisen plot [5], and execution time. Also, the biological significance of the generated gene clusters using different methods is analyzed using the Gene Ontology Term Finder [38], [39]. For each microarray gene expression data set, the number of gene clusters c is decided by using the CLICK [14] algorithm. The weight parameter w for rough-fuzzy clustering is set to 0.99, while the values of fuzzifiers $\acute{m}_1 = 2.0$ and $\acute{m}_2 = 2.0$. The source code of the proposed algorithm and the supplementary information are available at http://www.isical.ac.in/~pmaji/results/rrfcm.html.

3.1 Gene Expression Data Sets Used

In this paper, publicly available fourteen yeast microarray time series gene expression data sets are used to compare the performance of different gene clustering methods. Table 1 presents the accession number, number of genes, and time points of each microarray data set, which are downloaded from *Gene Expression Omnibus* (http://www.ncbi.nlm.nih.gov/geo/).

3.2 Quantitative Measures

Following quantitative indices are used to evaluate the performance of different gene clustering algorithms for grouping functionally similar genes from microarray gene expression data sets.

3.2.1 Davies-Bouldin Index

The Davies-Bouldin (DB) index [36] is a function of the ratio of sum of within-cluster distance to between-cluster separation and is given by

$$\mathrm{DB} = \frac{1}{c}\sum_{i=1}^{c}\max_{i\neq k}\left\{\frac{S(v_i) + S(v_k)}{d(v_i, v_k)}\right\} \quad (14)$$

for $1 \leq i, k \leq c$. The DB index minimizes the within-cluster distance $S(v_i)$ and maximizes the between-cluster separation $d(v_i, v_k)$. Therefore, for a given data set and c value, the higher the similarity values within the clusters and the between-cluster separation, the lower would be the DB index value. A good clustering procedure should make the value of DB index as low as possible.

3.2.2 Dunn Index

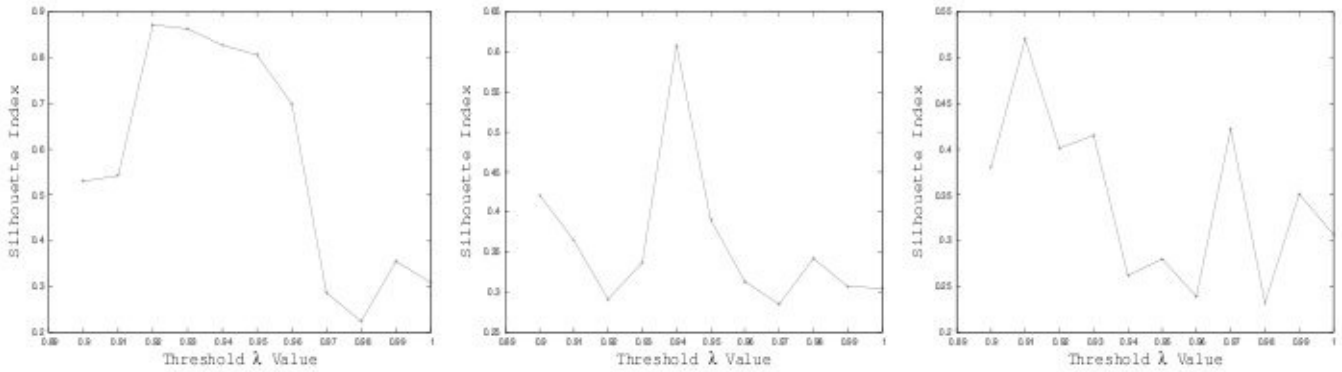
Dunn's index [36] is also designed to identify sets of clusters that are compact and well separated. Dunn's (D) index maximizes

$$\mathrm{D} = \min_{i}\left\{\min_{i\neq k}\left\{\frac{d(v_i, v_k)}{\max_l S(v_l)}\right\}\right\} \quad (15)$$

for $1 \leq i, k, l \leq c$. A good clustering procedure should make the value of Dunn index as high as possible.

TABLE 1
Brief Description of Fourteen Yeast Microarray Time Series Data Sets

Parameter/GDS	608	759	1013	1550	1611	2002	2003	2196	2267	2318	2347	2712	2713	2715
Genes	6303	6350	9275	9275	9275	5617	5617	9275	9275	6216	6228	9275	9275	9275
Time Points	10	24	24	6	96	30	30	12	36	13	13	21	21	21



(a) GDS1013 (b) GDS2267 (c) GDS2713

Fig. 2. Variation of Silhouette index for different values of threshold λ

3.2.3 β Index

The β index [37] is defined as the ratio of total variation and within-cluster variation, and is given by

$$\beta = \frac{\mathrm{N}}{\mathrm{M}}; \quad \text{where} \quad \mathrm{N} = \sum_{i=1}^{c}\sum_{j=1}^{n_i} \|x_{ij} - \overline{v}\|^2;$$

$$\mathrm{M} = \sum_{i=1}^{c}\sum_{j=1}^{n_i} ||x_{ij} - v_i||^2; \quad \text{and} \quad \sum_{i=1}^{c} n_i = n; \tag{16}$$

n_i is the number of objects in the ith cluster ($i = 1, 2, \cdots, c$), n is the total number of objects, x_{ij} is the jth object in cluster i, v_i is the mean or centroid of ith cluster, and $\overline{v}$ is the mean of n objects. For a given data set and c value, the higher the homogeneity within the clusters, the higher would be the β value. The value of β also increases with c.

4 Results and Discussion

The experimental results on fourteen microarray data sets are presented in this section. Subsequent discussions analyze the results with respect to DB index [36], Dunn index [36], Silhouette index [35], β index [37], Eisen plot [5], and execution time.

4.1 Optimum Value of Threshold λ

The threshold λ in (12) plays an important role to select initial cluster prototypes of different gene clusters. It controls the redundancy among the initial prototypes. Hence, it has a direct influence on the performance of proposed algorithm as well as other c-means algorithms.

To find out the optimum values of threshold λ for different microarray gene expression data sets, the Silhouette index [35] is used. Let a gene $x_i \in \beta_r$, $i = 1, ..., n_r$ and n_r is the cardinality of cluster β_r. For each gene x_i let a_i be the average distance between gene x_i and rest of the genes of β_r, that is,

$$a_i = d_{\text{avg}}(x_i, \beta_r - \{x_i\}) \tag{17}$$

where $d_{\text{avg}}(.,.)$ denotes the average distance measure between a gene and a set of genes. For any other cluster $\beta_p \neq \beta_r$, let $d_{\text{avg}}(x_i, \beta_p)$ denote the average distance of gene x_i to all genes of β_p. The scalar b_i is the smallest of these $d_{\text{avg}}(x_i, \beta_p), p = 1, \cdots, c, p \neq r$, that is,

$$b_i = \min_{p=1,\cdots,c,p\neq r} \{d_{\text{avg}}(x_i, \beta_p)\}. \tag{18}$$

The Silhouette width of gene x_i is then defined as

$$s(x_i) = \frac{b_i - a_i}{\max\{b_i, a_i\}} \tag{19}$$

where $-1 \leq s(x_i) \leq 1$. The value of $s(x_i)$ close to 1 implies that the distance of gene x_i from the cluster β_r where it belongs is significantly less than the distance between x_i and its nearest cluster excluding β_r, which indicates that x_i is well clustered. On the other hand, the value of $s(x_i)$ close to -1 implies that the distance between x_i and β_r is significantly higher than the distance between x_i and its nearest cluster excluding β_r, which indicates that x_i is not well clustered. Finally, the values of $s(x_i)$ close to 0 indicate that x_i lies close to the border between the two clusters. Based on the definition of $s(x_i)$, the Silhouette of the cluster β_k ($k = 1, \cdots, c$) is defined as

$$S(\beta_k) = \frac{1}{n_k} \sum_{x_i \in \beta_k} s(x_i) \tag{20}$$

where n_k is the cardinality of the cluster β_k. The global Silhouette index for threshold λ is defined as

$$\mathcal{S}_c = \frac{1}{c} \sum_{k=1}^{c} S(\beta_k) \tag{21}$$

where $\mathcal{S}_c \in [-1, 1]$. Also, the higher the value of $\mathcal{S}_c$, the better the corresponding clustering is.

TABLE 2
Performance of Proposed Algorithm on Fourteen Microarray Data Sets for Different Values of λ

Microarray Data Sets	Value of c	Cluster Validity Index	Different Values of Threshold λ										
			0.90	0.91	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.99	1
GDS608	26	Silhouette Index	0.27	0.25	0.19	0.18	0.19	0.18	0.16	0.15	0.16	0.13	0.17
		DB Index	0.92	0.95	1.07	1.20	1.18	1.48	1.53	1.69	1.50	1.80	1.30
		Dunn Index	0.55	**0.65**	0.49	0.61	0.55	0.40	0.42	0.40	0.49	0.39	0.56
GDS759	25	Silhouette Index	*	0.81	0.71	0.54	0.56	0.45	0.27	0.16	0.12	0.12	0.17
		DB Index	*	0.22	0.33	0.57	0.61	0.75	1.28	1.67	2.18	2.19	1.76
		Dunn Index	*	1.11	**1.17**	0.90	0.64	0.63	0.41	0.35	0.27	0.21	0.30
GDS1013	18	Silhouette Index	0.53	0.54	0.87	0.86	0.83	0.81	0.70	0.29	0.22	0.36	0.31
		DB Index	0.47	0.48	0.11	0.12	0.14	0.18	0.28	0.71	1.03	0.96	0.76
		Dunn Index	0.40	1.04	8.05	4.92	4.69	4.35	0.99	0.52	0.23	0.04	0.00
GDS1550	21	Silhouette Index	*	*	*	*	0.88	0.84	0.79	0.55	0.26	0.24	0.25
		DB Index	*	*	*	*	0.08	0.10	0.14	0.30	0.75	1.16	1.10
		Dunn Index	*	*	*	*	6.69	4.04	4.11	1.27	0.66	0.13	0.00
GDS1611	26	Silhouette Index	0.54	0.44	0.30	0.31	0.26	0.19	0.20	0.19	0.16	0.18	0.21
		DB Index	0.42	0.54	0.71	0.70	0.79	0.99	0.94	1.10	1.44	1.15	0.91
		Dunn Index	1.51	1.48	1.41	0.94	1.19	0.80	0.80	0.56	0.15	0.55	1.04
GDS2002	25	Silhouette Index	0.85	0.75	0.74	0.64	0.61	0.49	0.43	0.25	0.17	0.17	0.16
		DB Index	0.19	0.26	0.29	0.47	0.63	0.75	1.01	1.44	1.67	1.85	1.99
		Dunn Index	3.42	1.54	1.38	0.60	0.54	0.44	0.46	0.33	0.35	0.25	0.17
GDS2003	23	Silhouette Index	0.80	0.75	0.68	0.52	0.54	0.34	0.25	0.26	0.14	0.17	0.13
		DB Index	0.21	0.32	0.45	0.62	0.83	1.09	1.33	1.37	1.76	1.70	1.98
		Dunn Index	1.80	1.25	0.89	0.75	0.44	0.43	0.35	0.37	0.34	0.30	0.16
GDS2196	24	Silhouette Index	*	*	*	*	*	*	0.87	0.84	0.59	0.31	0.35
		DB Index	*	*	*	*	*	*	0.08	0.11	0.35	0.80	0.47
		Dunn Index	*	*	*	*	*	*	3.53	2.82	1.10	0.27	0.00
GDS2267	14	Silhouette Index	0.42	0.37	0.29	0.34	0.61	0.39	0.31	0.29	0.34	0.31	0.31
		DB Index	0.50	0.62	0.69	0.59	0.32	0.55	0.69	0.82	0.71	0.81	0.73
		Dunn Index	0.37	0.11	0.12	0.15	0.97	0.27	0.42	0.41	0.24	0.15	0.06
GDS2318	21	Silhouette Index	*	0.79	0.76	0.69	0.63	0.57	0.45	0.26	0.17	0.13	0.24
		DB Index	*	0.15	0.17	0.24	0.34	0.41	0.60	1.06	1.44	1.70	1.29
		Dunn Index	*	1.53	**1.69**	0.89	0.97	0.91	0.81	0.48	0.43	0.34	0.49
GDS2347	18	Silhouette Index	0.88	0.87	0.81	0.70	0.68	0.67	0.47	0.34	0.20	0.12	0.29
		DB Index	0.09	0.13	0.19	0.21	0.20	0.26	0.49	0.86	1.35	1.82	1.25
		Dunn Index	3.10	2.95	0.95	1.03	1.18	1.12	0.85	0.45	0.40	0.33	0.38
GDS2712	15	Silhouette Index	0.56	0.39	0.40	0.36	0.32	0.27	0.23	0.27	0.33	0.36	0.35
		DB Index	0.33	0.44	0.53	0.58	0.54	0.71	0.72	0.96	0.62	0.53	0.55
		Dunn Index	0.81	0.23	0.26	0.62	0.28	0.31	0.35	0.09	0.12	0.11	0.06
GDS2713	14	Silhouette Index	0.38	0.52	0.40	0.42	0.26	0.28	0.24	0.42	0.23	0.35	0.31
		DB Index	0.49	0.35	0.44	0.40	0.63	0.69	0.93	0.41	1.67	0.53	0.66
		Dunn Index	0.24	0.67	**0.84**	0.30	0.53	0.46	0.24	0.20	0.04	0.11	0.07
GDS2715	16	Silhouette Index	0.58	0.39	0.39	0.38	0.34	0.27	0.39	0.40	0.41	0.33	0.30
		DB Index	0.30	0.46	0.45	0.44	0.50	0.66	0.41	0.43	0.46	0.56	0.65
		Dunn Index	1.13	0.23	0.27	0.27	0.23	0.22	0.25	0.17	0.15	0.13	0.04

For fourteen microarray data sets, the value of λ is varied from 0.90 to 1.0. Fig. 2 represents the variation of Silhouette index with respect to different values of λ on GDS1013, GDS2267, and GDS2713 data sets considering $w = 0.99$, $\acute{m}_1 = 2.0$, and $\acute{m}_2 = 2.0$. From the results reported in Fig. 2, it is seen that as the threshold λ increases, the Silhouette index value increases and attains its maximum value at a particular value of $\lambda^\star$. After that the Silhouette index value decreases with the increase in the value of λ. Hence, the optimum value of λ for each data set is obtained using the following relation:

$$\lambda^\star = \arg\max_{\lambda}\{\mathcal{S}_c\}. \tag{22}$$

The optimum values of λ^* obtained using (22) are 0.90 for GDS608, GDS1611, GDS2002, GDS2003, GDS2347, GDS2712, and GDS2715, 0.91 for GDS759, GDS2318, and GDS2713, 0.94 for GDS1550 and GDS2267, 0.92 and 0.96 for GDS1013 and GDS2196, respectively. Finally, Table 2 presents the performance of the proposed clustering algorithm for different values of λ. The results and subsequent discussions are presented in this table with respect to Silhouette index, DB index, and Dunn index. From the results reported in Table 2, it is seen that the proposed clustering algorithm achieves its best performance at $\lambda = \lambda^\star$, irrespective of the cluster validity indices used. However, the Dunn index attains its maximum values at $\lambda = 0.91$ for GDS608 data and 0.92 for GDS759, GDS2318, and GDS2713 data, which are marked bold in Table 2.

4.2 Random Versus Proposed Initialization Method

Table 3 provides the comparative results of different c-means algorithms with random initialization of centroids and proposed initialization method described in Section 2.6 for fourteen yeast microarray data sets. The best results of each c-means clustering algorithm are reported for their optimal λ values. In most of the cases, the proposed initialization method is found to improve the performance in terms of Silhouette index, DB index, and Dunn index for all c-means algorithms. Out of 168 comparisons, the proposed initialization method is found to provide significantly better results in 147 cases

TABLE 3
Comparative Performance Analysis of Random and Proposed Initialization Methods

Data Sets	Initial Centers	Silhouette Index				DB Index				Dunn Index			
		HCM	FCM	RFCM	rRFCM	HCM	FCM	RFCM	rRFCM	HCM	FCM	RFCM	rRFCM
GDS608	Random	0.078	0.005	0.110	0.238	1.931	2.082	1.608	0.974	0.256	0.000	0.272	0.848
	Proposed	0.082	0.013	0.147	**0.269**	1.922	2.070	1.396	**0.921**	0.255	0.000	0.348	**0.551**
GDS759	Random	0.082	0.017	0.121	0.278	2.392	2.898	1.779	1.231	0.035	0.000	0.081	0.453
	Proposed	0.300	0.037	0.373	**0.814**	1.718	1.990	0.878	**0.221**	0.073	0.000	0.268	**1.112**
GDS1013	Random	0.220	0.249	0.213	0.508	0.937	1.439	1.142	0.522	0.001	0.002	0.001	0.083
	Proposed	0.475	0.253	0.486	**0.872**	0.475	1.515	0.736	**0.109**	0.036	0.003	0.074	**8.050**
GDS1550	Random	0.245	0.259	0.243	0.419	0.871	1.439	1.042	0.565	0.000	0.001	0.000	0.078
	Proposed	0.451	0.280	0.466	**0.878**	0.531	1.330	0.491	**0.080**	0.011	0.003	0.141	**6.692**
GDS1611	Random	0.158	0.088	0.177	0.232	1.384	1.725	1.193	0.793	0.417	0.016	0.493	1.144
	Proposed	0.243	0.089	0.368	**0.539**	1.179	1.890	0.690	**0.420**	0.502	0.071	0.419	**1.506**
GDS2002	Random	0.079	0.001	0.109	0.413	2.129	2.876	1.679	0.979	0.021	0.000	0.037	0.455
	Proposed	0.175	0.527	0.372	**0.849**	1.729	2.030	0.989	**0.188**	0.025	0.000	0.407	**3.421**
GDS2003	Random	0.082	0.014	0.128	0.364	2.033	2.975	1.672	0.989	0.045	0.000	0.056	0.430
	Proposed	0.191	0.168	0.308	**0.802**	1.635	1.780	1.900	**0.211**	0.074	0.000	0.000	**1.801**
GDS2196	Random	0.309	0.300	0.317	0.446	0.553	1.182	0.691	0.463	0.000	0.001	0.000	0.025
	Proposed	0.493	0.383	0.481	**0.865**	0.375	1.090	0.836	**0.083**	0.020	0.003	0.118	**3.529**
GDS2267	Random	0.230	0.197	0.233	0.495	0.888	2.402	1.006	0.951	0.011	0.008	0.016	0.143
	Proposed	0.317	0.197	0.313	**0.608**	0.793	1.110	0.864	**0.325**	0.022	0.008	0.015	**0.965**
GDS2318	Random	0.153	0.065	0.220	0.430	1.684	1.986	1.674	0.767	0.015	0.000	0.031	0.649
	Proposed	0.353	0.086	0.438	**0.791**	0.813	0.940	0.789	**0.151**	0.051	0.000	0.179	**1.532**
GDS2347	Random	0.134	0.031	0.160	0.474	2.147	2.794	2.594	0.692	0.005	0.000	0.007	0.519
	Proposed	0.513	0.031	0.627	**0.875**	0.955	1.980	0.491	**0.087**	0.029	0.000	0.188	**3.101**
GDS2712	Random	0.250	0.208	0.251	0.618	0.806	1.760	0.790	0.209	0.031	0.012	0.038	0.724
	Proposed	0.310	0.211	0.318	**0.558**	0.711	1.684	0.624	**0.332**	0.047	0.013	0.062	**0.814**
GDS2713	Random	0.223	0.201	0.221	0.598	0.893	1.673	0.864	0.248	0.017	0.016	0.025	0.497
	Proposed	0.298	0.199	0.280	**0.520**	0.777	1.643	1.961	**0.350**	0.046	0.015	0.033	**0.669**
GDS2715	Random	0.222	0.177	0.224	0.581	0.866	4.032	0.840	0.251	0.027	0.016	0.031	0.492
	Proposed	0.277	0.185	0.260	**0.580**	0.875	1.990	0.872	**0.296**	0.035	0.015	0.022	**1.130**

compare to the random initialization method. From the results marked bold in Table 3, it is seen that the rRFCM algorithm with proposed initialization method performs better than any other c-means clustering algorithms in all cases irrespective of the initialization methods.

However, it can also be seen that the HCM algorithm with the proposed initialization method outperforms the rRFCM algorithm with random initialization method in five and four cases in terms of Silhouette index and DB index, respectively. On the other hand, the RFCM algorithm with the proposed initialization method performs better in six, five, and two cases compare to the rRFCM algorithm with random initialization method with respect to Silhouette index, DB index, and Dunn index, respectively. Also, the FCM algorithm with proposed initialization method performs better than the rRFCM algorithm with the random initialization method in only one case in terms of Silhouette index. The better performance of the proposed initialization method is achieved due to the fact that it enables the algorithm to converge to an optimum or near optimum solutions.

4.3 Quantitative Performance Analysis

This section presents the comparative performance analysis of different gene clustering algorithms with respect to Silhouette index, DB index, Dunn index, β index, and execution time.

4.3.1 Performance of Different C-Means Algorithms

In order to establish the importance of possibilistic lower approximation of proposed robust rough-fuzzy c-means (rRFCM) over crisp lower approximation of existing rough-fuzzy c-means (RFCM) [32], extensive experimentation is carried out on fourteen yeast microarray data sets. Results and subsequent discussions are presented in Table 4, along with the performance of both HCM [10] and FCM [27], for optimum values of λ. The bold value in Table 4 signifies the best value.

All the results reported in Table 4 establish the fact that the proposed gene clustering algorithm is superior to other c-means clustering algorithms. It is also seen that the proposed rRFCM algorithm achieves better results with comparable time than that obtained using existing RFCM algorithm, irrespective of the data sets and quantitative indices used. The possibilistic lower approximation of the rRFCM helps to extract gene groups of any shape, while crisp lower approximation of the RFCM is forced to extract circular shaped gene clusters. In effect, the chance of inclusion of noisy genes becomes more in the RFCM as compare to the proposed rRFCM algorithm. Hence, the possibilistic lower approximation of the proposed rRFCM helps in discovering clusters of genes that are highly similar to each other.

The best performance of the rRFCM, in terms of Silhouette, DB, Dunn, and β indices, is achieved due to the fact that the probabilistic membership function of the rRFCM handles efficiently overlapping gene clusters; and the concept of possibilistic lower approximation and probabilistic boundary of the rRFCM algorithm deals with uncertainty, vagueness, and incompleteness in cluster definition.

TABLE 4
Performance of HCM, FCM, RFCM, and rRFCM Algorithms

Different Indices	Method/ Algorithm	Microarray Data Sets/GDS													
		608	759	1013	1550	1611	2002	2003	2196	2267	2318	2347	2712	2713	2715
Silhouette Index	HCM	0.08	0.30	0.48	0.45	0.24	0.18	0.19	0.49	0.32	0.35	0.51	0.31	0.30	0.28
	FCM	0.01	0.04	0.25	0.28	0.09	0.53	0.17	0.38	0.20	0.09	0.03	0.21	0.20	0.19
	RFCM	0.15	0.37	0.49	0.47	0.37	0.37	0.31	0.48	0.31	0.44	0.63	0.32	0.28	0.26
	rRFCM	**0.27**	**0.81**	**0.87**	**0.88**	**0.54**	**0.85**	**0.80**	**0.87**	**0.61**	**0.79**	**0.88**	**0.56**	**0.52**	**0.58**
DB Index	HCM	1.92	1.72	0.47	0.53	1.18	1.73	1.63	0.37	0.79	0.81	0.95	0.71	0.78	0.88
	FCM	2.07	1.99	1.51	1.33	1.89	2.03	1.78	1.09	1.11	0.94	1.98	1.68	1.64	1.99
	RFCM	1.40	0.88	0.74	0.49	0.69	0.99	1.90	0.84	0.86	0.79	0.49	0.62	1.96	0.87
	rRFCM	**0.92**	**0.22**	**0.11**	**0.08**	**0.42**	**0.19**	**0.21**	**0.08**	**0.32**	**0.15**	**0.09**	**0.33**	**0.35**	**0.30**
Dunn Index	HCM	0.26	0.07	0.04	0.01	0.50	0.03	0.07	0.02	0.02	0.05	0.03	0.05	0.05	0.03
	FCM	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.01	0.00	0.00	0.01	0.02	0.02
	RFCM	0.35	0.27	0.07	0.14	0.42	0.41	0.00	0.12	0.02	0.18	0.19	0.06	0.03	0.02
	rRFCM	**0.55**	**1.11**	**8.05**	**6.69**	**1.51**	**3.42**	**1.80**	**3.53**	**0.97**	**1.53**	**3.10**	**0.81**	**0.67**	**1.13**
β Index	HCM	3.03	3.64	19.54	43.95	19.12	2.74	2.50	65.27	10.75	3.16	1.93	10.74	9.65	10.29
	FCM	2.01	3.16	22.57	47.44	17.60	1.00	1.28	65.75	9.42	2.46	1.64	10.02	9.16	9.26
	RFCM	2.62	2.70	14.34	23.74	15.93	1.74	1.62	32.18	10.57	1.90	1.55	10.08	4.72	8.01
	rRFCM	**4.68**	**4.03**	**41.4**	**56.34**	**23.01**	**3.43**	**3.25**	**102.91**	**39.51**	**3.42**	**3.31**	**34.78**	**34.26**	**32.14**
Time (sec)	HCM	**23**	**52**	28	32	617	81	43	49	40	42	25	**15**	**15**	42
	FCM	65	111	103	44	738	97	117	95	104	53	18	67	60	75
	RFCM	50	56	39	**10**	**327**	**13**	117	**30**	80	**23**	**8**	27	30	66
	rRFCM	40	85	**11**	**10**	623	51	**26**	82	**38**	28	11	32	36	**35**

TABLE 5
Performance of CLICK, SOM, and rRFCM on Fourteen Yeast Microarray Data Sets

Microarray Data Sets	Silhouette Index			DB Index			Dunn Index			β Index		
	CLICK	SOM	rRFCM	CLICK	SOM	rRFCM	CLICK	SOM	rRFCM	CLICK	SOM	rRFCM
GDS608	-0.04	-0.03	**0.27**	11.52	18.03	**0.92**	0.06	0.02	**0.55**	0.79	2.23	**4.68**
GDS759	-0.08	-0.02	**0.81**	27.91	19.03	**0.22**	0.02	0.01	**1.11**	0.82	2.31	**4.03**
GDS1013	-0.52	0.06	**0.87**	9713.81	3.40	**0.11**	0.00	0.00	**8.05**	0.97	3.64	**41.40**
GDS1550	-0.49	0.15	**0.88**	525.51	2.09	**0.08**	0.00	0.00	**6.69**	1.07	6.81	**56.34**
GDS1611	-0.27	0.05	**0.54**	69.72	8.04	**0.42**	0.01	0.05	**1.51**	1.02	15.49	**23.01**
GDS2002	-0.12	-0.05	**0.85**	26.70	13.41	**0.19**	0.03	0.00	**3.42**	0.82	1.67	**3.43**
GDS2003	-0.09	-0.06	**0.80**	17.61	15.22	**0.21**	0.05	0.01	**1.80**	0.82	1.70	**3.25**
GDS2196	-0.53	0.17	**0.87**	3728.40	2.58	**0.08**	0.00	0.00	**3.53**	0.97	4.33	**102.91**
GDS2267	-0.42	0.02	**0.61**	759.38	5.76	**0.32**	0.00	0.00	**0.97**	0.98	3.88	**39.51**
GDS2318	-0.13	-0.11	**0.79**	17.87	78.46	**0.15**	0.05	0.00	**1.53**	0.86	1.65	**3.42**
GDS2347	-0.11	-0.13	**0.88**	16.91	35.16	**0.09**	0.03	0.00	**3.10**	0.86	1.30	**3.31**
GDS2712	-0.42	0.07	**0.56**	293.57	2.07	**0.33**	0.00	0.00	**0.81**	0.90	3.28	**34.78**
GDS2713	-0.39	0.07	**0.52**	6493.09	1.97	**0.35**	0.00	0.00	**0.67**	0.92	3.87	**34.26**
GDS2715	-0.41	0.08	**0.58**	5473.46	1.98	**0.30**	0.00	0.00	**1.13**	0.89	3.48	**32.14**

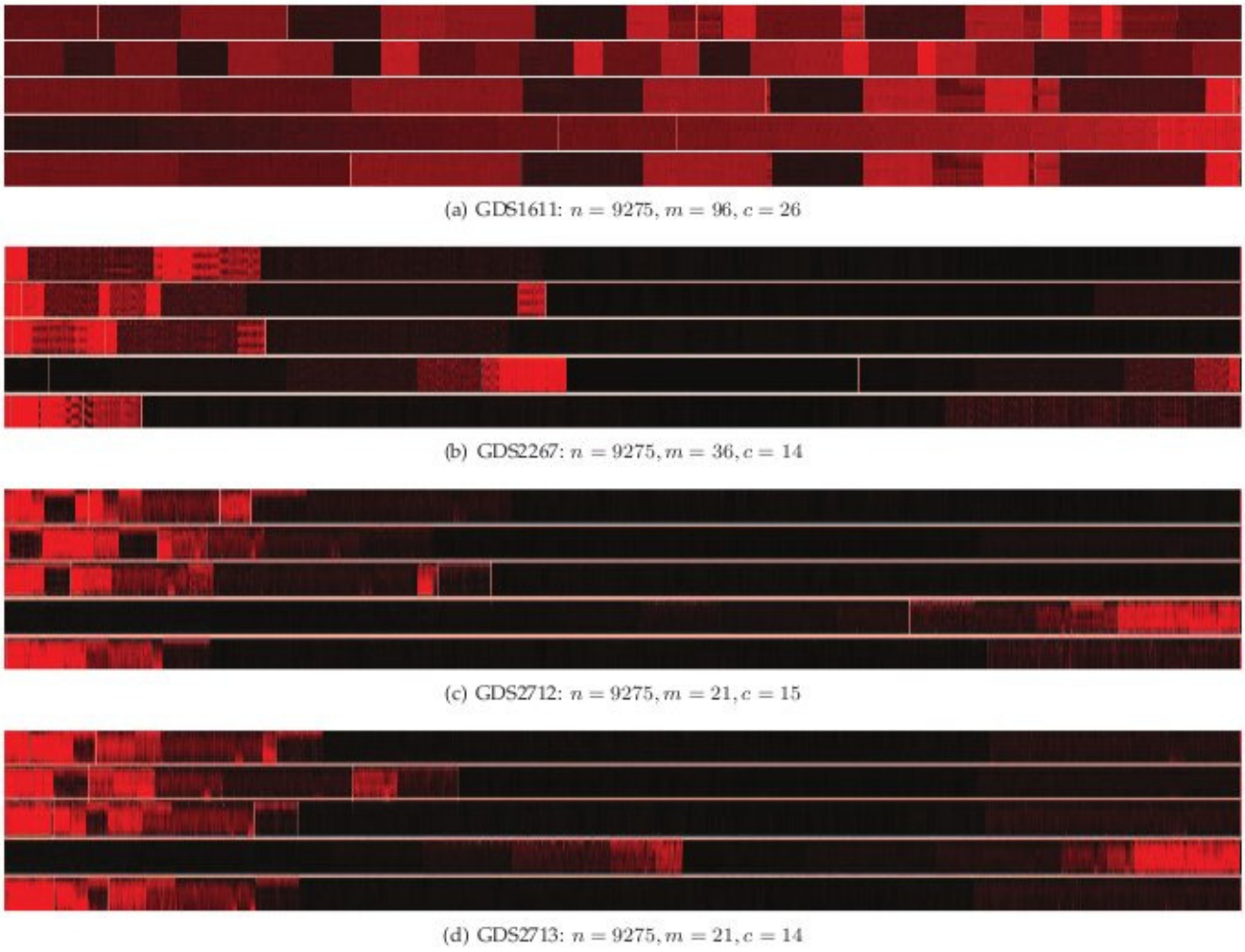
4.3.2 *Performance of CLICK, SOM, and rRFCM*

In order to establish the superiority of the proposed rRFCM algorithm over two existing gene clustering algorithms, namely, CLICK [14] and SOM [11], extensive experimentation is performed on fourteen yeast microarray data sets. Table 5 presents the comparative assessment of these three clustering algorithms, in terms of Silhouette index, DB index, Dunn index, and β index, where bold value represents the best value. From the results reported in this table, it can be seen that the proposed rRFCM algorithm performs significantly better than both CLICK and SOM, irrespective of microarray data sets and quantitative indices used. Hence, the proposed algorithm can identify compact groups of co-expressed genes.

4.4 Qualitative Performance Analysis

The Eisen plot gives a visual representation of the clustering result. In Eisen plot [5], the expression value of a gene at a specific time point is represented by coloring the corresponding cell of the data matrix with a color similar to the original color of its spot on the microarray. The shades of red color represent higher expression level, the shades of green color represent low expression level and the colors towards black represent absence of differential expression values. In the present representation, the genes are ordered before plotting so that the genes that belong to the same cluster are placed one after another. The cluster boundaries are identified by white colored blank rows.

The gene clusters produced by the HCM, FCM, RFCM, SOM, and rRFCM algorithms on fourteen yeast data sets are visualized by TreeView software, which is available at http://rana.lbl.gov/EisenSoftware and the plots for four data sets are reported in Fig. 3 as examples. From the Eisen plots presented in Fig. 3, it is evident that the expression profiles of the genes in a cluster are similar to each other and they produce similar color pattern, whereas the genes from different clusters differ in color patterns. Also, the results obtained by both RFCM and rRFCM algorithms are more promising than that by both HCM and FCM algorithms. For the purpose of illustration, the Eisen plots for gene clusters generated by the SOM are also presented in Fig. 3. From the plots



(a) GDS1611: $n = 9275, m = 96, c = 26$

(b) GDS2267: $n = 9275, m = 36, c = 14$

(c) GDS2712: $n = 9275, m = 21, c = 15$

(d) GDS2713: $n = 9275, m = 21, c = 14$

Fig. 3. Eisen plots of different clusters for four yeast data sets generated by HCM, FCM, RFCM, SOM, and rRFCM

presented in Fig. 3, it is clearly evident that the proposed rRFCM generates the Eisen plots having similar color pattern within the cluster as compare to the SOM.

5 Biological Significance Analysis

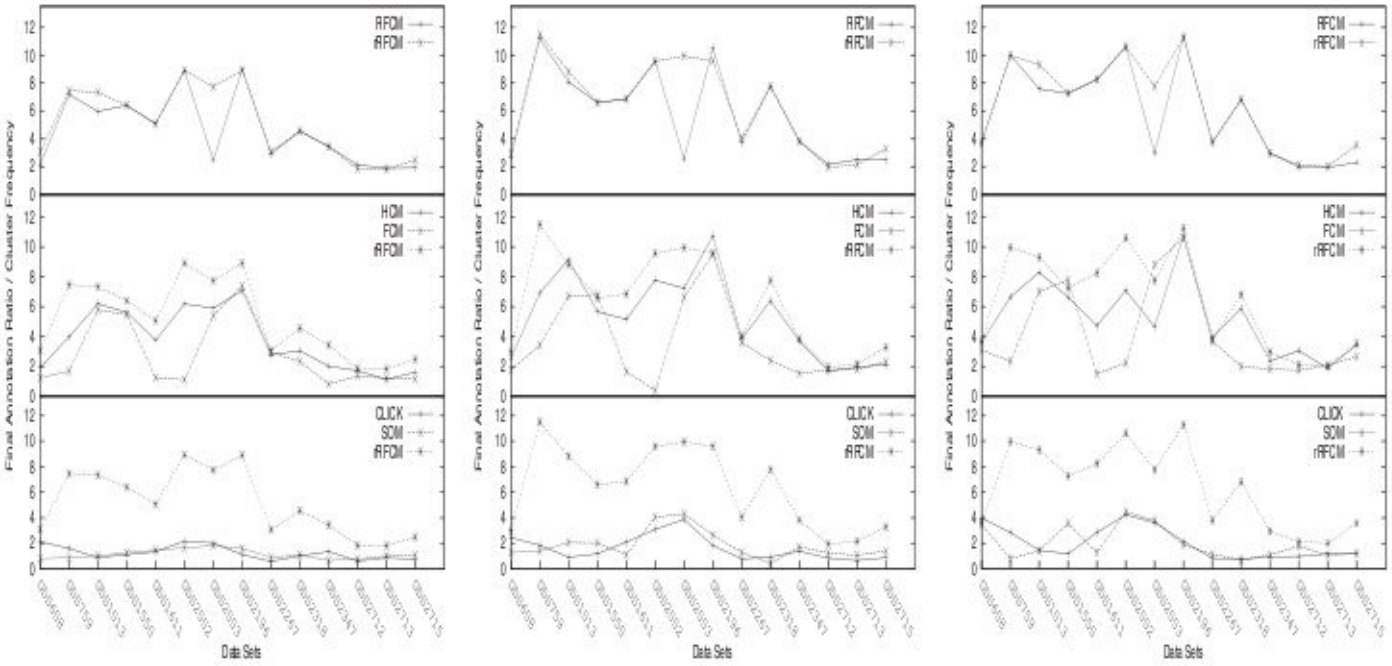
To interpret the biological significance of the generated gene clusters, the Gene Ontology (GO) Term Finder is used [38], [39]. It finds the most significantly enriched GO terms associated with the genes belonging to a cluster. The GO project aims to build tree structures, controlled vocabularies, also called ontologies, that describe gene products in terms of their associated biological processes (BP), molecular functions (MF) and cellular components (CC). The GO Term Finder determines whether any GO term annotates a specified list of genes at a frequency greater than that would be expected by chance, calculating the associated p-value by using the hypergeometric distribution and the Bonferroni multiple-hypothesis correction [38], [39]:

$$\mathrm{p} = 1 - \sum_{i=0}^{k-1} \frac{\binom{\mathcal{M}}{i}\binom{\mathcal{N}-\mathcal{M}}{n-i}}{\binom{\mathcal{N}}{n}} \tag{23}$$

where $\mathcal{N}$ is the total number of genes in the background distribution, $\mathcal{M}$ is the number of genes within that distribution that are annotated, either directly or indirectly, to the node of interest, n is the size of the list of genes of interest and k is the number of genes within that list which are annotated to the node. The closer the p-value is to zero, the more significant the particular GO term associated with the group of genes is, that is, the less likely the observed annotation of the particular GO term to a group of genes occurs by chance. On the other hand, the false discovery rate (FDR) is a multiple-hypothesis testing error measure indicating the expected proportion of false positives among the set of significant results. The FDR is particularly useful in the analysis of high-throughput data such as microarray gene expression.

5.1 Functional Consistency of Clustering Result

In order to evaluate the functional consistency of the gene clusters produced by different algorithms, the biological annotations of the gene clusters are considered in terms of the GO. The annotation ratios of each gene cluster in three GO ontologies are calculated using the GO Term Finder [38]. The GO term is searched in which most of the genes of a particular cluster are enriched.



(a) Molecular Functions (b) Biological Processes (c) Cellular Components

Fig. 4. Biological annotation ratios of different algorithms on fourteen gene expression data sets

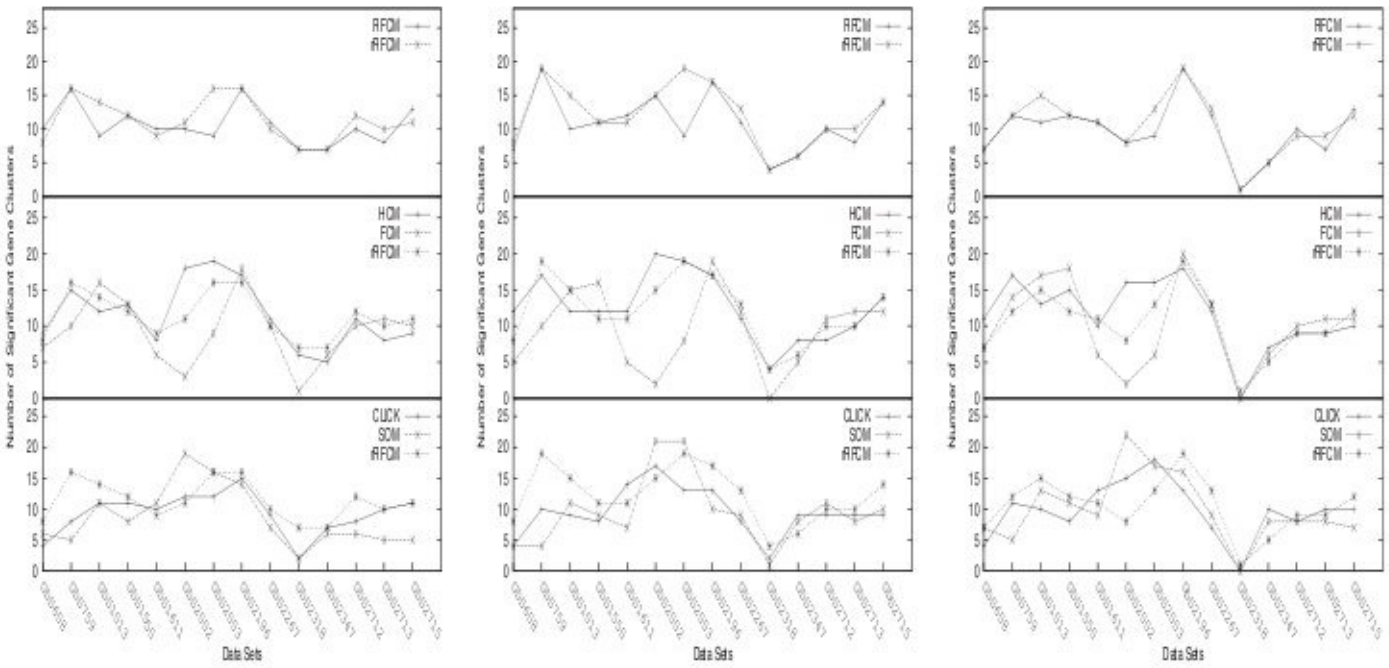
The annotation ratio, also termed as cluster frequency, of a gene cluster is defined as the number of genes in both the assigned GO term and the cluster divided by the number of genes in that cluster. A higher value of annotation ratio indicates that the majority of genes in the cluster are functionally more closer to each other, while a lower value signifies that the cluster contains much more noises or irrelevant genes. After computing the annotation ratios of all gene clusters for a particular ontology, the sum of all annotation ratios is treated as the final annotation ratio. A higher value of final annotation ratio represents that the corresponding clustering result is better than other, that is, the genes are better clustered by function, indicating a more functionally consistent clustering result [40].

The upper portion of Fig. 4 presents the comparative results of the RFCM and rRFCM algorithms, in terms of final annotation ratio or cluster frequency, for the MF, BP, and CC ontologies on fourteen yeast microarray data sets. All the results reported here confirm that the rRFCM provides higher or comparable final annotation ratios than that obtained using the RFCM algorithm in most of the cases. Out of 14 cases, the RFCM provides higher final annotation ratios in only 2, 3, and 1 cases for the MF, BP, and CC ontologies, respectively. The middle portion of Fig. 4 reports the comparative final annotation ratio or cluster frequency of the HCM, FCM, and rRFCM algorithms on fourteen data sets. From the results reported in this portion, it is seen that the proposed rRFCM algorithm attains higher final annotation ratio than that obtained using other c-means algorithms in 14, 11, and 9 cases for the MF, BP, and CC ontologies, respectively. On the other hand, the HCM achieves higher values, for the BP and CC ontologies, respectively, in 2 and 2 cases, while the FCM provides in 1 and 3 cases.

Finally, the lower portion of Fig. 4 compares the final annotation ratios obtained using the CLICK, SOM, and rRFCM algorithms. From the results reported in this portion, it can be seen that the final annotation ratio obtained using the proposed rRFCM algorithm is higher than that obtained using both CLICK and SOM, irrespective of the ontologies and data sets used. Hence, all the results reported in Fig. 4 establish the fact that the majority of genes in a cluster produced by the rRFCM algorithm are functionally more closer to each other than those by other algorithms, while the clusters obtained using existing algorithms include much more noises or irrelevant genes.

5.2 Biologically Significant Gene Clusters

This section presents the comparative performance analysis of different gene clustering algorithms, in terms of number of significant gene clusters generated. Fig. 5 presents the results for the MF, BP, and CC ontologies on fourteen yeast microarray data sets. The GO Term Finder [38] is used to determine the statistically significant gene clusters produced by different algorithms for all the GO terms from the MF, BP, and CC ontologies. If any cluster of genes generates a p-value smaller than 0.05, then that cluster is considered as a significant cluster. The upper portion of Fig. 5 presents the comparative results of the RFCM and rRFCM algorithms for the MF, BP, and CC ontologies. From the results, it is seen that the proposed rRFCM generates more or comparable number of significant gene clusters in ten, thirteen, and twelve cases, while the RFCM generates more number



(a) Molecular Functions (b) Biological Processes (c) Cellular Components

Fig. 5. Biologically significant gene clusters of different algorithms on fourteen gene expression data sets

of significant gene clusters in four, one, and two cases for the MF, BP, and CC ontologies, respectively.

The middle portion of Fig. 5 reports the number of significant gene clusters generated by the HCM, FCM, and rRFCM algorithms for the MF, BP, and CC ontologies for all microarray data sets. All the results reported in this portion establish the fact that the rRFCM algorithm generates more or comparable number of significant gene clusters than that of other c-means algorithms in most of the cases. For the MF ontology, out of fourteen cases, the rRFCM generates more significant gene clusters in eight cases, while the HCM generates in six cases. On the other hand, the rRFCM produces more or comparable number of significant gene clusters in nine cases and the HCM generates more number of significant gene clusters in only five cases for the BP ontology. In case of the CC ontology, the rRFCM algorithm generates more or comparable number of significant gene clusters in eight cases and the HCM generates more number of significant gene clusters in six cases. On the other hand, the rRFCM algorithm generates more or comparable number of significant gene clusters in ten cases and the FCM generates more number of significant gene clusters in four cases for both MF and BP ontologies. While, in the CC ontology, the rRFCM generates more number of significant gene clusters in seven cases and the FCM generates more number of significant gene clusters in seven cases. In other words, all the results reported in Fig. 5 establish the fact that the proposed rough and fuzzy set based rRFCM can discover more functionally similar groups of co-expressed genes than that of other c-means algorithms.

Finally, the performance of CLICK, SOM, and rRFCM algorithms is compared in lower portion of Fig. 5 with respect to the number of significant gene clusters generated for MF, BP, and CC ontologies. From the results reported in this portion, it is seen that the proposed rRFCM algorithm generates more or comparable number of significant gene clusters compare to CLICK and SOM algorithms in most of the cases. Out of fourteen cases, the rRFCM algorithm generates more or comparable number of significant gene clusters than both CLICK and SOM in twelve, nine, and nine cases for the MF, BP, and CC ontologies, respectively. Hence, the proposed rRFCM can generate more biologically significant gene clusters than both CLICK and SOM.

5.3 Biological Interpretation of Gene Clusters

This section presents the biological interpretation of some gene clusters those are generated only by the proposed rRFCM algorithm, but not generated by any other clustering algorithms. Table 6 presents the unique GO terms obtained using the rRFCM algorithm for GDS2003 data set as an example, along with the corresponding cluster index and frequency, p-value, and FDR.

In GDS2003 data set, the gene expression of JM43 and isogenic msn2/4 mutant KKY8 cells were recorded in aerobic to anaerobic shift condition [41]. Hence, this data set should reflect the processes those are involved in aerobic and anaerobic respiration of yeast cell. In anaerobic condition, yeast cell ferments and produces alcohol [42]. The GO term *alcohol biosynthetic process* corresponding to cluster 21 of the rRFCM reflects this activity of yeast cell. The yeast cells of this data set were cultured in galactose medium that acts as a derepressor, that is, in absence of glucose and in presence of galactose,

TABLE 6
Unique GO Terms Obtained Using Proposed Algorithm for GDS2003

Ontology	Cluster	GO Term / Gene Cluster	Frequency	P-Value	FDR (%)
Molecular Function	1	transcription regulator activity	0.072	3.22E-013	0.00
	6	transferase activity	0.400	2.28E-003	0.00
	8	succinate dehydrogenase activity	0.250	8.46E-004	0.00
	10	carbon-carbon lyase activity	0.250	5.40E-003	12.00
	12	fructose transmembrane transporter activity	0.154	2.06E-005	0.00
	13	copper ion binding	0.250	1.05E-002	6.00
	23	heme binding	0.429	7.27E-005	0.00
Biological Process	1	cellular component organization	0.338	8.47E-046	0.00
	3	interphase of mitotic cell cycle	0.333	2.16E-003	0.00
	8	tricarboxylic acid cycle	0.250	2.61E-002	12.00
	9	de novo′ IMP biosynthetic process	0.385	1.48E-010	0.00
	11	electron transport chain	0.304	1.65E-015	0.00
	12	hexose transport	0.231	4.76E-008	0.00
	21	alcohol biosynthetic process	1.000	9.56E-004	0.00
Cellular Component	11	mitochondrial respiratory chain	0.283	2.80E-020	0.00
	12	plasma membrane enriched fraction	0.154	1.06E-002	4.00

the GAL gene converts galactose into glucose that can be further utilized for generating energy [43]. The transcription of GAL gene is thus regulated by the presence of galactose. This phenomena can be reflected by the term *transcription regulator activity* of cluster 1.

The terms *electron transport chain* (ETC) and *mitochondrial respiratory chain* annotating the genes of cluster 11, and *tricarboxylic acid cycle* (TCA) of cluster 8 reflect the processes involved in aerobic respiration of yeast cells [42]. The enzyme succinate dehydrogenase is involved in between TCA and ETC cycles, which is the only enzyme of the TCA cycle that is an integral membrane protein [42]. The genes of cluster 8 obtained by the rRFCM are also annotated by the term *succinate dehydrogenase activity*. Moreover, the ETC cycle involves copper bound protein [44] and cytochromes [45]. These two processes are reflected by the terms *copper ion binding* and *heme binding* of cluster 13 and cluster 23, respectively, of the rRFCM algorithm. In order for galactose to be metabolized by yeast, it must first be transported into the cell. Yeast cells have hexose transporters in their membranes to tackle this task [43]. The rRFCM algorithm is also able to group genes in cluster 12 those are annotated to the term *hexose transport*.

The biological interpretation of some unique clusters identified by the proposed rRFCM algorithm reported above establish the fact that the algorithm generates significant gene clusters those are biologically relevant with respect to the given microarray data sets. The unique GO terms obtained using the rRFCM for all fourteen microarray data sets are available at http://www.isical.ac.in/~pmaji/results/rrfcm.html.

6 Conclusion

The contribution of the paper lies in developing a new gene clustering algorithm, which integrates judiciously c-means algorithm, rough sets, and probabilistic and possibilistic memberships of fuzzy sets. This formulation is geared towards maximizing the utility of both rough sets and fuzzy sets with respect to knowledge discovery tasks. The effectiveness of the proposed algorithm is demonstrated, along with a comparison with other related algorithms, on fourteen yeast microarray gene expression data sets using some standard cluster validity indices and gene ontology.

The proposed initialization method is found to provide better performance in 87.50% cases than random initialization; thereby successful in effectively circumventing the initialization and local minima problems of iterative refinement clustering algorithms like c-means. The proposed method also attains its best results in 82.14% cases for optimum parameter values. Moreover, the proposed algorithm performs significantly better than other methods, irrespective of the microarray data sets and quantitative indices used, and provides biologically significant and relevant gene clusters.

References

[1] H. Causton, J. Quackenbush, and A. Brazma, *Microarray Gene Expression Data Analysis: A Beginner's Guide*. Wiley-Blackwell, 2003.

[2] E. Domany, "Cluster Analysis of Gene Expression Data," *Journal of Statistical Physics*, vol. 110, no. 3-6, pp. 1117–1139, 2003.

[3] P. Maji and S. K. Pal, *Rough-Fuzzy Pattern Recognition: Applications in Bioinformatics and Medical Imaging*. New Jersey, Hoboken: John Wiley & Sons, Inc., 2012.

[4] D. Jiang, C. Tang, and A. Zhang, "Cluster Analysis for Gene Expression Data: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1370–1386, 2004.

[5] M. B. Eisen, P. T. Spellman, O. Patrick, and D. Botstein, "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proceedings of the National Academy of Sciences, USA*, vol. 95, no. 25, pp. 14 863–14 868, 1998.

[6] S. Tavazoie, D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic Determination of Genetic Network Architecture," *Nature Genetics*, vol. 22, no. 3, pp. 281–285, 1999.

[7] A. Brazma and J. Vilo, "Minireview: Gene Expression Data Analysis," *Federation of European Biochemical Societies Letters*, vol. 480, no. 1, pp. 17–24, 2000.

[8] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Mining the Gene Expression Matrix: Inferring Gene Relationships from Large Scale Gene Expression Data," *Information Processing in Cells and Tissues*, pp. 203–212, 1998.

[9] J. Herrero, A. Valencia, and J. Dopazo, "A Hierarchical Unsupervised Growing Neural Network for Clustering Gene Expression Patterns," *Bioinformatics*, vol. 17, no. 2, pp. 126–136, 2001.

[10] L. J. Heyer, S. Kruglyak, and S. Yooseph, "Exploring Expression Data: Identification and Analysis of Coexpressed Genes," *Genome Research*, vol. 9, no. 11, pp. 1106–1115, 1999.

[11] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, "Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation," *Proceedings of the National Academy of Sciences, USA*, vol. 96, no. 6, pp. 2907–2912, 1999.

[12] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering Gene Expression Patterns," *Journal of Computational Biology*, vol. 6, no. 3-4, pp. 281–297, 1999.

[13] E. Hartuv and R. Shamir, "A Clustering Algorithm Based on Graph Connectivity," *Information Processing Letters*, vol. 76, no. 4-6, pp. 175–181, 2000.

[14] R. Shamir and R. Sharan, "CLICK: A Clustering Algorithm for Gene Expression Analysis," in *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, 2000.

[15] E. P. Xing and R. M. Karp, "CLIFF: Clustering of High-Dimensional Microarray Data via Iterative Feature Filtering Using Normalized Cuts," *Bioinformatics*, vol. 17, no. 1, pp. 306–315, 2001.

[16] C. Fraley and A. E. Raftery, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis," *The Computer Journal*, vol. 41, no. 8, pp. 578–588, 1998.

[17] D. Ghosh and A. M. Chinnaiyan, "Mixture Modelling of Gene Expression Data from Microarray Experiments," *Bioinformatics*, vol. 18, no. 2, pp. 275–286, 2002.

[18] G. J. McLachlan, R. W. Bean, and D. Peel, "A Mixture Model-Based Approach to the Clustering of Microarray Expression Data," *Bioinformatics*, vol. 18, no. 3, pp. 413–422, 2002.

[19] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzz, "Model-Based Clustering and Data Transformations for Gene Expression Data," *Bioinformatics*, vol. 17, no. 10, pp. 977–987, 2001.

[20] D. Jiang, J. Pei, and A. Zhang, "DHC: A Density-Based Hierarchical Clustering Method for Time-Series Gene Expression Data," in *Proceedings of the 3rd IEEE International Symposium on Bioinformatics and Bioengineering*, 2003, pp. 393–400.

[21] L. Klebanov and A. Yakovlev, "How High is the Level of Technical Noise in Microarray Data?" *Biology Direct*, vol. 2, no. 9, 2007.

[22] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[23] Z. Pawlak, *Rough Sets: Theoretical Aspects of Resoning About Data*. Dordrecht, The Netherlands: Kluwer, 1991.

[24] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact, Well-Separated Clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1974.

[25] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithm*. New York: Plenum, 1981.

[26] P. J. Woolf and Y. Wang, "A Fuzzy Logic Approach to Analyzing Gene Expression Data," *Physiological Genomics*, vol. 3, pp. 9–15, 2000.

[27] D. Dembele and P. Kastner, "Fuzzy C-Means Method for Clustering Microarray Data," *Bioinformatics*, vol. 19, no. 8, pp. 973–980, 2003.

[28] E. R. Dougherty, J. Barrera, M. Brun, S. Kim, R. M. Cesar, Y. Chen, M. Bittner, and J. M. Trent, "Inference from Clustering with Application to Gene-Expression Microarrays," *Journal of Computational Biology*, vol. 9, no. 1, pp. 105–126, 2002.

[29] A. P. Gasch and M. B. Eisen, "Exploring the Conditional Coregulation of Yeast Gene Expression Through Fuzzy K-Means Clustering," *Genome Biology*, vol. 3, no. 11, pp. 1–22, 2002.

[30] N. Belacel, M. Cuperlovic-Culf, M. Laflamme, and R. Ouellette, "Fuzzy J-Means and VNS Methods for Clustering Genes from Microarray Data," *Bioinformatics*, vol. 20, no. 11, pp. 1690–1701, 2004.

[31] R. Krishnapuram and J. M. Keller, "A Possibilistic Approach to Clustering," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 98–110, 1993.

[32] P. Maji and S. K. Pal, "RFCM: A Hybrid Clustering Algorithm Using Rough and Fuzzy Sets," *Fundamenta Informaticae*, vol. 80, no. 4, pp. 475–496, 2007.

[33] P. Maji and S. K. Pal, "Rough Set Based Generalized Fuzzy C-Means Algorithm and Quantitative Indices," *IEEE Transactions on System, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 6, pp. 1529–1540, 2007.

[34] P. Maji, "Fuzzy-Rough Supervised Attribute Clustering Algorithm and Classification of Microarray Data," *IEEE Transactions on System, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 1, pp. 222–233, 2011.

[35] J. P. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpration and Validation of Cluster Analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 1, pp. 53–65, 1987.

[36] J. C. Bezdek and N. R. Pal, "Some New Indexes for Cluster Validity," *IEEE Transactions on System, Man, and Cybernetics, Part B: Cybernetics*, vol. 28, no. 3, pp. 301–315, 1988.

[37] S. K. Pal, A. Ghosh, and B. U. Shankar, "Segmentation of Remotely Sensed Images with Fuzzy Thresholding and Quantitative Evaluation," *International Journal of Remote Sensing*, vol. 21, no. 11, pp. 2269–2300, 2000.

[38] E. I. Boyle, S. Weng, J. Gollub, H. Jin, D. Botstein, J. M. Cherry, and G. Sherlock, "GO::Term Finder Open Source Software for Accessing Gene Ontology Information and Finding Significantly Enriched Gene Ontology Terms Associated with a List of Genes," *Bioinformatics*, vol. 20, no. 18, pp. 3710–3715, 2004.

[39] J. L. Sevilla, V. Segura, A. Podhorski, E. Guruceaga, J. M. Mato, L. A. Martinez-Cruz, F. J. Corrales, and A. Rubio, "Correlation Between Gene Expression and GO Semantic Similarity," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 2, no. 4, pp. 330–338, 2005.

[40] H. Wang, Z. Wang, X. Li, B. Gong, L. Feng, and Y. Zhou, "A Robust Approach Based on Weibull Distribution for Clustering Gene Expression Data," *Algorithms for Molecular Biology*, vol. 6, no. 1, p. 14, 2011.

[41] L. C. Lai, A. L. Kosorukoff, P. V. Burke, and K. E. Kwast, "Dynamical Remodeling of the Transcriptome during Short-Term Anaerobiosis in *Saccharomyces cerevisiae*: Differential Response and Role of Msn2 and/or Msn4 and Other Factors in Galactose and Glucose Media," *Molecular and Cellular Biology*, vol. 25, no. 10, pp. 4075–4091, 2005.

[42] G. M. Walker, *Yeast Physiology and Biotechnology*. England, West Sussex: John Wiley & Sons, Inc., 1998.

[43] D. J. Timson, "Galactose Metabolism in *Saccharomyces cerevisiae*," *Dynamic Biochemistry, Process Biotechnology and Molecular Biology*, vol. 1, no. 1, pp. 63–73, 2007.

[44] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*. New York: Garland Science, 2007.

[45] H. Pelletier and J. Kraut, "Crystal Structure of a Complex Between Electron Transfer Partners, Cytochrome c Peroxidase and Cytochrome c," *Science*, vol. 258, no. 5089, pp. 1748–1755, 1992.

Pradipta Maji received the BSc degree in physics, the MSc degree in electronics science, and the PhD degree in the area of computer science from Jadavpur University, India, in 1998, 2000, and 2005, respectively. Currently, he is an assistant professor in the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. His research interests include pattern recognition, machine learning, computational biology and bioinformatics, medical image processing, and so forth. He has published more than 70 papers in international journals and conferences. He is an author of a book published by Wiley-IEEE Computer Society Press, and also a reviewer of many international journals. He has received the 2006 Best Paper Award of the International Conference on Visual Information Engineering from The Institution of Engineering and Technology, U.K., the 2008 Microsoft Young Faculty Award from Microsoft Research Laboratory India Pvt., the 2009 Young Scientist Award from the National Academy of Sciences, India, and the 2011 Young Scientist Award from the Indian National Science Academy, India, and has been selected as the 2009 Young Associate of the Indian Academy of Sciences, India.

Sushmita Paul received the BSc degree in biotechnology from Rajasthan University, India in 2005 and the MSc degree in bioinformatics from Banasthali Vidyapith, Rajasthan, India in 2007. Currently, she is a CSIR senior research fellow in the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India. Her research interests include computational biology and bioinformatics, pattern recognition, soft computing, and so forth. She has published around 10 papers in international journals and conferences. She has received the 2009 Best Paper Award of the International Conference on Information Technology from the Orissa Information Technology Society, India.