Comparative Analysis of Clustering and Biclustering Algorithms for Grouping of Genes: Co-Function and Co-Regulation

Anindya Bhattacharya[1], Nirmalya Chowdhury[2] and Rajat K. De[3,*]

[1]*Center for Integrative and Translational Genomics, University of Tennessee Health Science Center, 858 Madison Avenue, Memphis, TN 38163, USA*

[2]*Department of Computer Science and Engineering, Jadavpur University, Kolkata, India*

[3]*Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India*

Abstract: In this article, we discuss the basic challenges of clustering on gene expression data. In particular, we divide the methods of clustering into eight different categories. Then, we present specific characteristics pertinent to each clustering category. We compare the results of 27 clustering/biclustering algorithms on various gene expression datasets using different cluster validation indices. Comparison is made in terms of P -value on the best and three best clusters obtained by each algorithm along with overall results using z-score. Biclustering algorithms are also compared in terms of their capacity in handling overlapping biclusters. Finally, we provide some guidelines for the development of new clustering algorithms for gene expression data analysis.

Availability of the software: The software for most of the existing clustering algorithms has been developed using C and Visual Basic languages, and can be executed on the Microsoft Windows platforms. The software may be downloaded as a zip file from http: //www.isical.ac.in/rajat. Then it needs to be installed. Two word files (included in the zip file) need to be consulted before installation and execution of the software.

Keywords: Density-based clustering, functional enrichment, grid-based clustering, hierarchical clustering, partitional clustering, p-value, transcription factors, z-score.

1. INTRODUCTION

In this article, we compare the performance of various clustering algorithms in grouping co-expressed and co-regulated genes. It may be mentioned here that co-expressed genes are normally co-functional. These clustering algorithms may be grouped into various categories. Categorization of clustering algorithms is neither straightforward, nor canonical. Clustering algorithms [1, 2] can be grouped into eight categories, namely, partitioning, hierarchical, density-based, grid-based, graph-based, model-based, soft computing methods and biclustering.

Cluster analysis is used, in general, to identify potentially meaningful relationships among genes and/or experiments [3-6]. In particular, it is often used to infer biological functions by asso- ciating unknown genes with other genes that have similar expression patterns and known functions [7, 8]. One such example is the large-scale analysis of gene expression pattern as a function of cell cycle in yeast [9], in which a large volume of gene-expression data has been analyzed to identify genes that are co- expressed. A previous attempt [10] demonstrates how gene expression data and functional annotations can be used together to estimate the probability that genes share a common regulatory mechanism. Combining mRNA expression data with functional annotation results in a better predictive model than using either data source alone.

*Address correspondence to this author at the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India; Tel: +91-33-25753105; Fax: +91-33-25783357; E-mails: rajatkde@yahoo.co.in, rajat@isical.ac.in

However, genes may be regulated by different regulators over a time course. Co-regulation in part of the time course does not guarantee a global similarity in gene expression profiles. Therefore, new clustering algorithms emerge to address this issue [11]. Several biclustering algorithms have been proposed to discover sets of genes that are co-regulated only in a part of the experimental conditions under study. However, these algorithms may not be suitable for clustering gene expression time-series data because they ignore the internal relationship among various patterns at different time points. While analyzing the time-course gene expression data, it is necessary to consider the internal connection between these patterns and preserve the time locality in time-course gene expression data.

Some of the previous interesting surveys on clustering [12] and biclustering algorithms [12, 13] are biased towards more theoretical description of the methods. Prelic *et al.* [14] have compared performances of different biclustering algorithms on gene expression data, and proposed their divide-and-conquer biclustering algorithm (Bimax).

Here, we consider twenty one (21) clustering and six (6) biclustering algorithms of various categories, and compare their performances using five gene expression datasets. The ability to identify groups of co-functional and co-regulated genes by these algorithms will be analyzed. These 21 clustering algorithms are K-means [1, 2, 15], EM clustering [16], PAM [2], CLARA [2], CLARANS [2], Agglomerative algorithm [2, 17] with single linkage, complete linkage and average linkage, DIANA [2], [18], BIRCH [19], CURE [20], CHAMELEON [21], DBSCAN [22], DENCLUE [23],

2212-392X/12 $58.00+.00

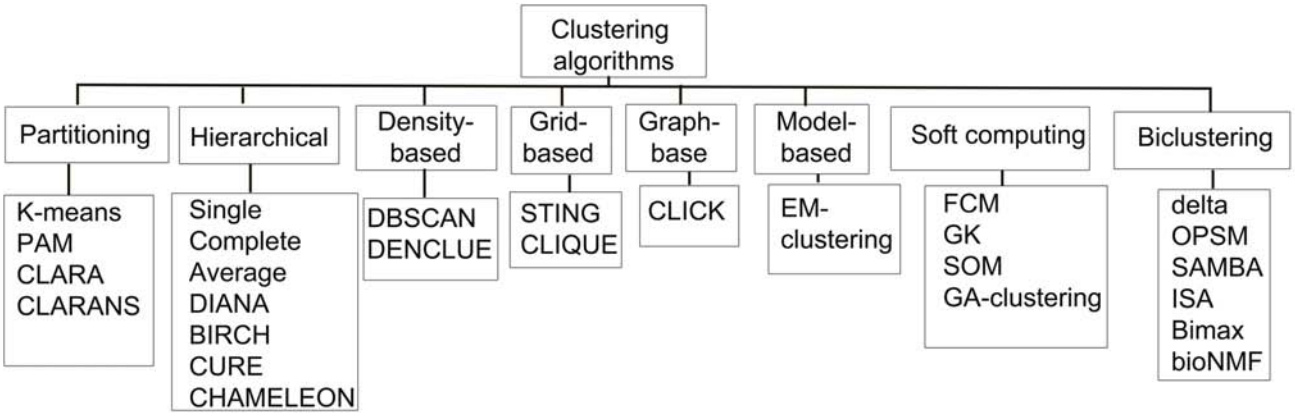

Fig. (1). Categorization of clustering algorithms.

STING [24], CLIQUE [25], CLICK [26], SOM [27], GA-clustering [28], FCM [29-31], and GK [32, 33]; six biclustering algorithms like deltabicluster [11], bioNMF [34], OPSM [35], SAMBA [36], ISA [37, 38] and Bimax [14]. It is to be mentioned here that we are considering normalized data that do not have any missing value. If there is any missing value, it may be estimated using standard software like LSimpute [39] and Expander [36].

Issues that are common to different clustering methods while dealing with gene expression data are data preprocessing (discussed in Appendix B), determination of appropriate number of clusters to be formed, proximity measures, the ability to find clusters of irregular shapes, handling outliers and assessment of clustering results. Ability to find clusters of irregular shape, handling outliers and determination of appropriate number of clusters are mentioned in Section 2. In Section 3, a brief theoretical background of gene expression and gene expression data clustering are presented. Section 4 provides the analysis of the results on gene expression datasets for all the aforesaid algorithms. From these results, a concluding remark on gene expression data clustering is made in Section 5. Data preprocessing techniques and proximity measures used with different clustering algorithms are discussed in Appendix B.

2. CATEGORIES OF VARIOUS CLUSTERING ALGORITHMS AND THEIR CHARACTERISTICS

In this section, characteristics of various algorithms like ability to find clusters of irregular shapes and handling outliers are discussed. A technique for determination of appropriate number of clusters has also been described.

2.1. Categories

Clustering algorithms can be categorized into eight classes, namely, partitioning (K-means, PAM, CLARA and CLARANS), hierarchical (single linkage, complete linkage, average linkage, DIANA, BIRCH, CURE and CHAMELEON), density-based (DBSCAN and DENCLUE), grid-based (STING and CLIQUE), model-based (EM clustering), graph-based (CLICK), soft computing (SOM, GA-clustering, FCM and GK), and biclustering algorithms (deltabicluster, bioNMF, OPSM, SAMBA, ISA and Bimax). Fig. (**1**) shows this classification of algorithms. Properties of different clustering and biclustering algorithms are summarized in Table **1**.

Partitioning algorithms try to discover clusters by iteratively relocating points among subsets. As checking of all possible subsets is computationally very expensive, certain heuristics are used in the form of iterative optimization. Using these heuristics, different relocation schemes reassign points in several clusters iteratively. With appropriate data, this results in high quality clusters. The most popular algorithm of this kind is K-means [1, 2, 40]. Other variation of partitioning algorithms include Partitioning Around Medoid (PAM) [2], CLARA [2] and CLARANS [2]. Such methods concentrate on how well points fit into their clusters. The major problem associated with the algorithms in this category is the need for a number of clusters, which these algorithms take as an input. Although, DB index, Dunn index [1, 2], [12] or any other cluster validity indices are used for the determination of an optimal number of clusters for a given dataset, this number may be found to be different for different cluster validity indices. Partitioning algorithms normally form spherically shaped clusters. K- means algorithm is sensitive to outliers. PAM, CLARA and CLARANS are less affected by outliers since they represent clusters using medoids instead of means, which leads to embedded resistance against outliers.

While partitioning algorithms form clusters directly, hierarchical algorithms build them gradually. Hierarchical clustering methods form a cluster hierarchy, or in other words, a tree of clusters, also known as a dendrogram. Every cluster node contains child clusters; sibling clusters share the points covered by their common parent. Such an approach allows exploring data at different levels of granularity. Traditional hierarchical clustering algorithms include agglomerative algorithm [2, 17] and DIANA [2, 18]. The agglomerative algorithm uses either single linkage, average linkage or complete linkage strategies. Hierarchical clustering algorithms, namely, BIRCH [19], CURE [20] and CHAMELEON [21] are already in mentioned in the literature for handling large datasets. The advantage of hierarchical clustering algorithms is that they can produce K clusters from an input dataset without taking K as an input. Here K is determined based on the threshold value of some parameters. Different problems associated with this group of algorithms are: (i) unlike partitioning algorithms, hierarchical algorithms may produce one large cluster and several singleton clusters, and (ii) hierarchical algorithms

cannot repair defects already occurred in a clustering step to produce proper clustering solution.

Hierarchical clustering based on linkage metrics suffers from the problem of outliers, as selection of threshold similarity for agglomeration or division step is trivial. For large threshold values, outliers may become parts of a cluster. The outliers form single- ton clusters and so as several non-outliers, for small threshold values. The algorithms result in clusters of convex shapes, rather than that of irregular shapes.

In order to discover clusters with arbitrary shapes, Density-based algorithms exist in literature. An open set in the Euclidean space can be divided into a set of its connected components. The implementation of this idea for partitioning a finite set of points requires concepts of density, connectivity and boundary. They are closely related to a point's nearest neighbors. A cluster, defined as a connected dense component, grows in a direction along which density attains its maximum. Therefore, density-based algorithms are capable of discovering clusters of arbitrary shapes. This also provides a natural protection against outliers. Density-based algorithms include DBSCAN [22] and DENCLUE [23]. Algorithm DENCLUE is, in fact, a blend of a density-based clustering and a grid- based preprocessing. For DBSCAN, the points that are not connected are declared to be outliers and they are not covered by any cluster. Thus DBSCAN is not susceptible to outliers. DBSCAN and DENCLUE are able to produce clusters with arbitrary (irregular) shapes.

In the case of density based partitioning methods, crucial concepts of density, connectivity and boundary are used, which require elaborate definitions. Another way of dealing with them is to shift attention from data to space partitioning. Algorithms involved in partitioning space are called grid-based methods. The grid-based clustering approach uses a multi-resolution grid data structure to quantize the space into a finite number of cells that form a grid structure on which all the operations for clustering are performed. The main advantage of this approach is its fast processing time. They frequently use hierarchical agglomerative algorithms as a phase of processing. The algorithms STING [24] and CLIQUE [26] are examples of this category. Grid-based methods can handle outliers well. STING is unable to construct clusters of different shapes as it ignores spatial relationship between children cells and their neighboring cells. The algorithm CLIQUE (Clustering In QUEst) combines density-based clustering and grid-based clustering to produce clusters with arbitrary shapes.

Graph-based clustering techniques convert the problem of clustering a dataset into graph theoretical problems of finding minimal cut or maximal cliques in the proximity graph G. In the proximity graph G, objects are represented by nodes and the proximity between two objects is indicated by the weight of the edge between the corresponding nodes. The algorithm CLICK (CLuster Identification *via* Connectivity Kernels) [26] falls in this category. CLICK seeks to identify highly connected components in the proximity graph as clusters. CLICK and other graph- based clustering algorithms have been able to produce clusters with arbitrary shapes. However, CLICK does not guarantee that generated partitions are able to remove outliers from the remaining data objects.

Model-based clustering techniques attempt to maximize the extent of fitting the given data to some mathematical model. Such methods are often based on the assumption that the data are generated by a mixture of underlying probability distribution. EM clustering [16] is an example of model-based clustering that uses a probabilistic model. It takes a conceptual point of view for identifying the clusters with a certain model whose unknown parameters have to be determined. EM clustering is actually a general probabilistic frame- work of K-means algorithm [41]. Similar to K-means, EM clustering is sensitive to outliers as the algorithm also considers these outliers for calculation of mixture model parameters. It is unable to obtain clusters of irregular shapes. The shape of clusters obtained by EM algorithm is mainly spherical and depends on the mixture model used.

Many clustering techniques have been developed under soft computing paradigm for gene expression data analysis. Soft computing refers to a collection of computational techniques (like theory of fuzzy sets, rough sets, artificial neural networks and genetic algorithms) for studying, modeling and analyzing complex phenomena for which conventional methods do not yield low cost solutions. Soft Computing methodology uses soft techniques contrasting with classical hard computing techniques. For hard computing, there is a direct connection between the size of a problem and the amount of resources needed to solve the problem. Soft computing aims at surmounting problems by using inexact methods to give useful but inexact answers to large problems. Hence soft computing can be a suitable computing technique for large scale gene expression data clustering. There exist clustering algorithms under soft computing paradigm. For example, SOM [27] is a single layer neural network based model, FCM [29-31] and GK [32] are fuzzy clustering algorithms, GA clustering is a GA based clustering algorithm under the notion of evolutionary computation. Properties of the algorithms FCM [29-31], GK [32, 33] and GA-clustering [28] are similar to those of partitioning clustering algorithms.

Common disadvantage of all the above clustering algorithms is that they try to find groups of genes that are co-expressed through all the experimental conditions (measurements). But in reality genes tend to be co-expressed under only a few experimental conditions. They may start behaving differently under different conditions. If an input dataset has many conditions and an algorithm tries to find groups of genes expressed similarly under all the conditions then chance of finding such a group with success is very low [42]. In order to overcome this problem with the clustering algorithms, the concept of biclusterin has emerged. Biclustering algorithms perform simultaneous grouping on genes and conditions of a dataset to determine subgroups of genes that exhibit similar behavior over a subset of experimental conditions.

Several biclustering algorithms [11, 37, 42-45,] have been proposed till date. They include, among others, Block Clustering by Hartigan [43], deltabiclusters by Cheng and Church [11], Coupled Two-Way Clustering (CTWC) by Getz *et al.* [44], Spectral biclustering by Kluger *et al.* [45], Iterative Signature Algorithm (ISA) of Ihmels *et al.* [37, 38], Plaid model by Lazzeroni and Owen [46], Order Preserving Sub Matrix (OPSM) algorithm of Ben-Dor *et al.* [35],

Table 1. Some of the Characteristics of the Clustering and Biclustering Algorithms. Here, n is the Number of Genes, m is the Number of Samples, d is the Upper Bound on the Degree of Each Vertex, l is the Number of Biclusters, and c is a Positive Constant

Algorithms	Shape	Handling Outliers	Time	References
K-means	regular	No	O(n)	[2]
EM clustering	regular	No	O(n)	[2, 16]
PAM	regular	Yes	O(n)	[2]
CLARA	regular	Yes	O(n)	[2]
CLARANS	regular	Yes	O(n2)	[2]
single linkage	regular	No	O(n2 logn)	[2]
complete linkage	regular	No	O(n2 logn)	[2]
average linkage	regular	No	O(n2 logn)	[2]
DIANA	regular	No	O(n2 logn)	[2, 18]
BIRCH	regular	Yes	O(n)	[2, 19]
CURE	irregular	Yes	O(n2 logn)	[2, 20]
CHAMELEON	irregular	Yes	O(n2)	[2, 21]
DBSCAN	irregular	Yes	O(n2)	[2, 22]
DENCLUE	irregular	Yes	O(n2)	[2, 23]
STING	regular	Yes	O(n)	[2, 24]
CLIQUE	irregular	Yes	O(cm + n × m)	[25]
CLICK	irregular	No	O(n)	[26]
SOM	regular	Yes	O(n)	[2, 27]
GA-clustering	regular	Yes	O(n2)	[28]
FCM	regular	No	O(n)	[29-31]
GK	regular	No	O(n)	[32, 33]
delta-bicluster		Yes	O(nm)	[11]
ISA		Yes	linear	[37, 38]
SAMBA		No	O(n2d)	[36]
OPSM		No	O(nm3 l)	[35]
Bimax		No	O(n × m2)	[14]
bioNMF		No	O(n2)	[34]

SAMBA of Tanay *et al.* [36] and xMOTIF of Murali and Kasif [47]. Prelic *et al.* [14] have compared performance of different biclustering algorithms and proposed a fast divide-and-conquer biclustering algorithm (Bimax).

Apart from different methods of biclustering, Pascual-Montano *et al.* [34] have applied the no- tion of non-negative matrix factorization (NMF) [48] to the analysis of gene-array experiments and de- signed a software tool called bioNMF. It is capable of recognizing similarity between sub portions of the data corresponding to localized features in expression space, and is able to produce biclusters as subsets of genes behaving similarly over a subset of ex- perimental measurements. Another competitive tool with different biclustering techniques regarding the analysis of gene expression data is Mining Attribute Profile (MAP) [49]. The algorithm can be characterized as a depth-first search and divide-and-conquer algorithms. Application of MAP to gene expression data allows identification of genes whose expression values follow similar pattern in response to different biological conditions.

The algorithms deltabiclusters, ISA, OPSM, SAMBA, Bimax and bioNMF may be able to detect groups with irregular shapes as they perform grouping without representative genes. ISA and deltabiclusters are able to handle outliers well. OPSM, SAMBA, Bimax and bioNMF have shown vulnerability in the presence of outliers. Most of the biclustering algorithms are approximation scheme of some NP hard problems, and the main difference lies in the approximation strategy that is used. OPSM, delta-biclusters and Bimax are greedy algorithms, ISA is a signature algorithm and SAMBA is an exhaustive enumeration type algorithm. All these algorithms try to avoid generating two or more biclusters with nearly the same set of samples and/or genes. A common approach is to remove a bicluster from the output if it shares a large fraction of genes and/or samples (based on a user-specified threshold value) with an already computed bicluster. Another approach replaces the expression values in a bicluster with random values in order to prevent the bicluster from being formed again. In spite of these measures, biclustering algorithms may compute tens,

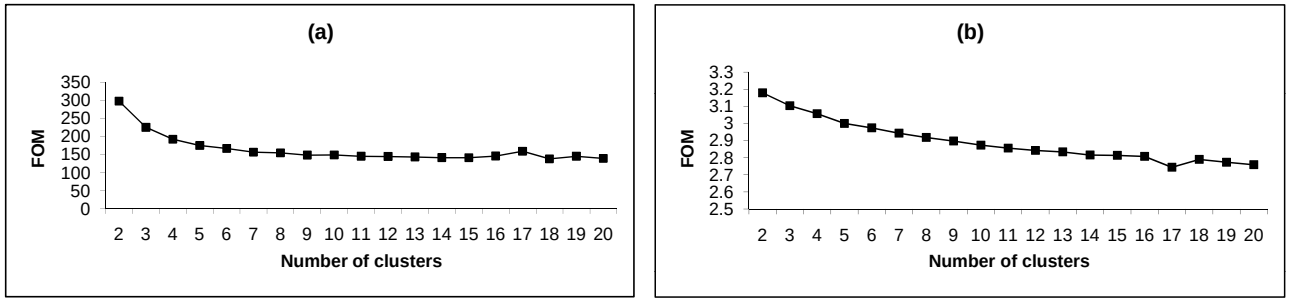

Fig. (2). FOM plot of K-means (**a**) YCCD (**b**) SPTD.

hundreds, or even thousands of biclusters with varying degrees of overlap.

Some of the characteristics of the clustering and biclustering algorithms considered here, are given in Table **1**. It shows the shape of the clusters (irregular/regular), the algorithms, in the first column, generate, and the ability of these algorithms to handle outliers in the dataset. CURE, CHAMELEON, DB- SCAN, DENCLUE, CLIQUE and CLICK are able to form clusters of irregular shapes. Regarding outliers, agglomerative algorithms with single linkage, complete linkage and average linkage, DIANA, K-means, CLICK, EM clustering, FCM, and GK are unable to handle outliers properly. Among the biclustering algorithms, deltabicluster and ISA are able to show better performances in the presence of outliers.

2.2. Determining Number of Clusters

For hierarchical methods like grid, density and graph-based clustering algorithms, number of clusters to be created does not need to be supplied as an input. In partitioning and soft computing based methods, number (K) of clusters to be created is a user specified parameter. Several cluster validity indices including Bezdek's partition coefficient, Dunn's separation in- dex, Xie-Beni's separation index, Davies-Bouldin's index and Gath-Geva's index, [1, 2] can be used for determination of an optimal number of clusters for a given dataset.

The similarity in expression values of the genes within a cluster increases the possibility of finding biologically significant clusters. Based on this idea, Yeung *et al.* [50] proposed a specific figure of merit (FOM) to estimate the predictive power of a clustering algorithm (described in Appendix A). FOM measures the mean deviation of the expression levels of genes in a sample relative to their corresponding cluster means. Thus, a small value of FOM indicates a strong prediction strength, and therefore a high level reliability of the resulting clusters.

In this article, we have used FOM to determine the K -values. A plot of FOM for different K -values is used for the selection of the best K -value corresponding to the lowest FOM-value. Fig. (**2**) shows plots of FOM with respect to K for K-means clustering algorithm on the datasets YCCD and SPTD. Here, the lowest FOM- value is associated with K = 18 for YCCD, while that for SPTD is K = 17. This procedure is followed for almost all the algorithms applied to all the datasets. In order to restrict the size of the article, we have not included all the other figures.

3. CLUSTERING ON GENE EXPRESSION DATA

Currently gene expression data is the main source of experimental data available for the study of functional genomics using different computational modeling in bioinformatics field. Amount of mRNA produced from a gene is a measure of gene expression. Higher the amount of mRNA, higher is the expression, and vice versa. Gene expression [51, 52] is activity level of a gene and it is measured by different experimental techniques. Northern blot [51, 52] and reverse transcription polymerase chain reaction (RTPCR) [51], [52] are techniques to measure gene expression of a single gene. A new and improved gene expression measurement technique, that is known as microar ray technology [51-53], allows simultaneous measurement of gene expression for hundreds or thousands of genes.

Clustering algorithms have been widely used for grouping genes based on their microarray gene expression profiles [26, 45-47, 52, 54-56]. Basic idea of all these clustering algorithms is as follows. If we consider a set of n genes $X = \{x_1, x_2, \ldots, x_n\}$, for each of which m expression values are given, they will have to be grouped into K disjoint clusters $C_1, C_2, \ldots, C_p, \ldots, C_K$.

Gene expression data clustering is performed to identify underlying meaningful relationships among genes and/or different experiments [3, 4, 5, 33, 52]. Genes with similar expression values are known as co-expressed genes and genes with similar expression pattern; over all the samples are known as co- related genes. Co-expressed and/or co-related genes are expected to share common functionality. Clustering algorithms assign co-expressed and/or co-related genes in the same cluster. Thus, gene expression data clustering is often used to predict functionality of un- known genes from other genes in the same cluster [7, 8]. Analyzing co-expressed [9] gene to infer biological functions is a common practise in bioinformatics.

Group of co-related genes usually share common control (regulatory) mechanism and often used to infer regulatory modules. In a previous attempt [10], it is demonstrated that, gene expression data and functional annotations can be analyzed together to estimate the probability that a co-related group of genes share a common regulatory mechanism. Group of genes with common regulatory mechanisms are known as co-regulated genes. Co-regulated genes usually show similarity in their expression profiles i.e., they are usually co-related. Clusters of co-related genes are performed to understanding regulatory relationships between

Table 2. A Short Description of the Datasets Used in Clustering and Biclustering Algorithms

Name (Organism)	Number of Genes	Number of Samples
Yeast Cheng and Church dataset (YCCD) (Yeast)	2879	17
Spellman *et al.* dataset (SPTD) (Yeast)	6178	77
GDS958 (Mouse)	22690	12
GDS2547 (Homo sapiens)	12646	164
GDS2938 (Homo sapiens)	22283	12

genes [57]. Understanding regulatory relationships between genes eventually leads towards understanding of causes for different diseases and abnormalities.

4. RESULTS

The effectiveness along with comparative analysis of all the clustering algorithms is demonstrated on five gene expression profiles. These profiles deal with two yeasts (Yeast Cheng and Church (YCCD) and Spellman *et al.* (SPTD) datasets) (http: //yfgdb.princeton.edu/) and three mammals (GDS958, GDS2547 and GDS2938) (http: //www.ncbi.-nlm.gov/). The rows/columns with all zeros or null values are deleted from the datasets before applying these algorithms. For example, five such rows are deleted from the original YCCD. A short description of these datasets is given in Table **2**, while they are detailed in Section 4.2. The performance of the algorithms is also compared using the indices, namely, z-score for homogeneity and P -value (on functional annotation and on transcription factors) for cluster reliability. Comparisons in terms of capability of handling over- lapping biclusters and time complexity have also been included.

For determination of K -value, we have used FOM- plots and selected the K -value corresponding to the lowest FOM-value. We have adjusted other parameters of different clustering algorithms based on minimization of FOM-values. Parameter settings for different algorithms are given in Table **3**.

Table 3. Parameter Settings Used for Different Clustering Algorithms. (n is the Number of Genes and K is the Number of Clusters)

Algorithms	Parameter Values
K-means	K = 2 : 20
EM clustering	K = 2 : 20
PAM	K = 2 : 20
CLARA	K = 2 : 20, sample size=500,1000, number of samples=2
CLARANS	K= 2 : 20, number of neighbors=20, number of local minimum=2
single linkage	distance threshold T = 1 : 50
complete linkage	distance threshold T = 1 : 50
average linkage	distance threshold T = 1 : 50
DIANA	distance threshold T = 50 : 1
BIRCH	branching factor B = 5, 10, distance threshold T = 1 : 50
CURE	number of representative c = 3 : 9, shrinking factor α = 0.3 : 0.7
CHAMELEON	distance threshold T = 1 : 50
DBSCAN	ε= 1 : 10, MinPts = 4 : 10
DENCLUE	ξ = 10 : 50
STING	distance threshold d = 5, density threshold= 4
CLIQUE	density threshold ξ = 5
CLICK	default settings in Expander [58]
SOM	default settings in Expander [58]
GA-clustering	K = 2 : 20, population size P = 20 : 50
FCM	K = 2: 20
GK	K = 2: 20
delta-bicluster	δ = 0.5, α = 1.2
ISA	tg = 0.5, 1.0, 1.5, 2.0, tc = 1.0, 1.5, 2.0, seed = 100
SAMBA	D = 40, N1 = 4, N2 = 6, k = 20, L = 30
OPSM	l = 100
Bimax	Minimum number of genes = 2, Minimum number of chips = 2
bioNMF	Sparseness = 0, Number of iterations = 1000, Stopping threshold = 1e-4

4.1. Validation of Clustering Results

For gene expression data, clustering algorithms result in groups of co-expressed genes, co-regulated genes, groups of samples with a common phenotype, or "blocks" of genes and samples involved in specific biological processes. However, different clustering algorithms, or even a single clustering algorithm using different parameter values, generally result in different sets of clusters. Therefore, it is important to compare various clustering results and select the one that best fits the "true" data distribution. Cluster validation is the process of assessing the quality and reliability of the clusters derived from various clustering methods. Here we describe two indices, namely, z-score and P -value for assessment and comparison of clustering results.

4.1.1. z-Score

There are various definitions for the homogeneity of clusters, which measure the similarity of data objects in clusters. Cluster separation is analogously defined from various perspectives measuring the dissimilarity between two clusters. Since these definitions of homogeneity and separation are based on the similarity between objects, the quality of clusters increases with higher homogeneity and separation values. Cluster validity indices are usually adopted to measure the homogeneity and separation of a clustering result. Comparing values of validity indices for different clustering results obtained by various clustering algorithms, we can identify the best clustering solutions. But

problem is that none of these indices measures functional similarity between genes inside a cluster and dissimilarity between genes in different clusters. These indices indicate homogeneity and separation based on gene expression values. For this reason, z- score computed from clustering results can be used as a measure of cluster homogeneity.

z-score [59] is calculated by observing the relation between a cluster and the functional annotation of the genes in the cluster. Here, an attribute database is used to create an $n \times n_a$ gene-attribute table for n genes and n_a attributes $\{A_1, A_2, \ldots, A_{n_a}\}$ in which a '1' in position (i, j) indicates that the gene i is known to possess attribute A_j , and a '0' indicates our lack of knowledge on the fact whether gene i possesses attribute A_j or not. With this gene-attribute table, we construct a contingency table for each cluster- attribute pair, whose rows are labeled by clusters and columns are labeled by attributes. That is, the order of the contingency table is $K \times n_a$, where $C = \{C_1, C_2, \ldots, C_K\}$ is the set of K clusters obtained by a clustering algorithm. The entries are nonnegative integers giving the number of observed attributes for each cluster. Thus the entry in position (k, j) of the contingency table indicates the number of genes in cluster C_k , which is associated with an attribute A_j . From the contingency table, we compute the entropy $H_{A_j C}$ for each attribute A_j and a set of clusters, C , H_C for C , and H_{A_j} for each of the n_a attributes in the table independent of clusters [59, 60].

Using mutual information (M I) between two variables X and Y, i.e., M I (X, Y) = H (X) + H (Y) − H (X, Y), and assuming both absolute and conditional independence of attributes, we expand the total mutual information as a sum of mutual information between the set of clusters and each individual attribute.

Thus, the total mutual information [59] between a set of clusters, C , and the set of attributes, A, is given by

$$MI(C, A_1, A_2, \ldots, A_{n_a}) = \sum_{t=1}^{n_a} MI(C, A_t) = n_a \times H_C + \sum_{t=1}^{n_a} H_{A_t} - \sum_{t=1}^{n_a} H_{A_t C}. \quad (1)$$

Now z-score is defined as [59].

$$z = \frac{MI_{real} - MI_{random}}{s_{random}} \quad (2)$$

The term MI_{real} is the computed M I for the clustered data using the attribute database. MI_{random} is computed in the following way. For a set of clusters, obtained by randomly assigning genes to clusters of uniform size, we can compute a M I -value. This is repeated many times until a distribution of M I -values is obtained. Mean of these M I -values, computed for randomly obtained clusters, is MI_{random} , and stan- dard deviation of these M I -values is s_{random} . Higher the value of z-score, better is the clustering solution.

4.1.2. P -VALUE for Cluster Reliability

While z-score is used to compare homogeneity of the clusters, this comparison may not reveal the reliability of the resulting clusters. For gene expression data analysis, P-value is used to check reliability of clustering solution. P-value indicates whether an observed level of annotations for a group of genes is significant within the context of annotation for all the genes within a reference set of genes. Let us assume that we have a population of N genes, in which M genes have a particular annotation. If we observe x genes with that annotation in a sample of n genes, then we can calculate the probability of that observation using the hyper geometric distribution, and is given by

$$P = \frac{\binom{M}{x}\binom{N-M}{n-x}}{\binom{N}{n}} \quad (3)$$

In order to obtain a P -value, we raise the question: What is the probability of having x or more out of n genes with a given annotation, given that M out of N have that particular annotation? In other words, P - value is the chance of seeing the observation, or better, given the background distribution. This is calculated by summing P for x out of n, x + 1 out of n, x + 2 out of n, and so on. Thus, the chance of seeing x or more genes with an annotation, out of n genes, given that M in the population of N have that annotation, is given by

$$P - value = \sum_{j=x}^{n} \frac{\binom{M}{j}\binom{N-M}{n-j}}{\binom{N}{n}} \quad (4)$$

A specific GO functional category is said to be "enriched" if the corresponding P -value is less than a predefined threshold value. A low P -value indicates that genes belonging to the enriched functional categories are biologically significant in the corresponding clusters. In the present article, only functional categories with P -value < 1.0×10^{-7} are reported as enriched functional categories.

Calculation of P -values for transcription factors in gene clusters is another measure of cluster reliability. Jakt *et al.* [61] have developed a method to estimate P -value for finding a certain number of matches to a transcription factor in all the gene clusters. Here we assume that we have a total population of N genes, in which M have a particular transcription factor. If we observe x genes with that transcription factor in a sample of n genes, then we can calculate P -values for that transcription factor using Equation **4**. A smaller P -value indicates a higher significance of the clustering results and vice versa. Number of enriched transcription factors for each cluster/bicluster of datasets is found based on P -values. In the present article, only transcription factors with P -value < 1.0×10^{-4} are reported as significant.

4.2. Gene Expression Datasets

Yeast Cheng and Church Dataset (YCCD): The yeast gene expression dataset, named as Yeast Cheng and Church dataset (YCCD), consists of the expression levels of 2884 yeast genes for 17 experiments. YCCD was used by Cheng and Church [11] for biclustering. The dataset was originally developed by Tavazoie *et al.* [15].

Spellman *et al.* Dataset (SPTD): Spellman *et al.* [9] dataset (SPTD) consists of the expression levels of 6178 yeast genes for 77 experiments. SPTD is related to cell cycle of the budding yeast (Saccharomyces cerevisiae).

GDS958: The oligonucleotide microarray gene expression data GDS958 generated by Wills-Karp *et al.* [62] consists of the expression levels of 22690 mouse genes of lung cells for 12 experiments on lung cells. Wills-Karp *et al.*

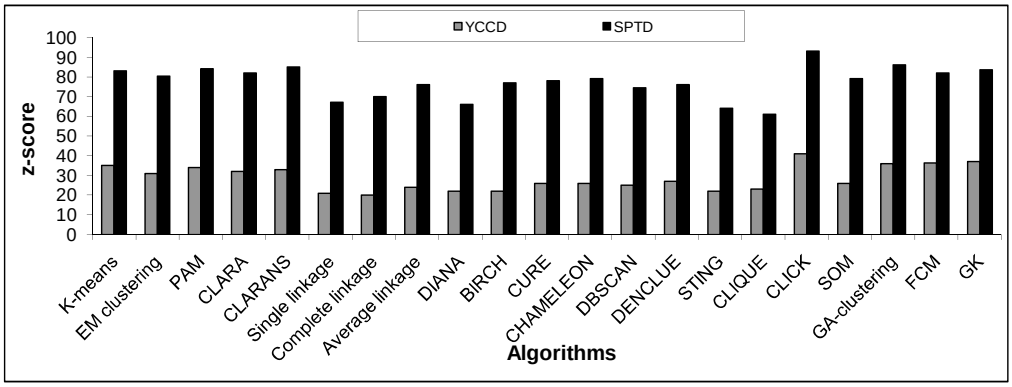

Fig. (3). Comparison of z-scores for clustering algorithms.

used GDS958 to find out responsible genes in asthma mediation [62].

Further information on this data is available at http: //www.ncbi.nlm.gov/. It is to be mentioned here that the dataset contains expression profiles of cytokines including IL-13, IL-4, IL-5, and their receptors being known as key mediators for allergen induced immediate development of asthma. This is due to hyper reactivity of the airway and mucus overproduction in the lung as immediate response to house dust mite allergen.

GDS2547: The metastatic prostate cancer gene expression data GDS2547 [63, 64] was generated from metastatic prostate tumors, primary prostate tumors, normal tissue adjacent to the tumor and normal donor tissue of Homo sapiens. GDS2547 consists of the expression levels of 12646 genes in 164 experiments.

GDS2938: The dataset GDS2938 [65] was generated for studying the effect of IFN-γ and IL-1β effect on thyroid epithelial cells of Homo sapiens. GDS2938 consists of the expression levels of 22283 genes in 12 experiments.

4.3. Homogeneity Comparison Using z-Score

For homogeneity comparison, we have used z-score. Fig. (**3**) shows the z-scores computed on the clusters obtained by the aforesaid clustering algorithms using the datasets. To calculate z-score for yeast datasets, Gibbons ClusterJudge [59] tool has been used. Saccharomyces Genome Database (SGD) annotation of the yeast genes, along with the gene ontology developed by the Gene Ontology Consortium [66, 67] has been used by ClusterJudge for the calculation of z-scores. ClusterJudge only supports yeast datasets. Calculation of z-score combines the mutual information calculated from a clustering solution produced by a clustering algorithm with the mutual information of random clustering. Biclusters produced by a biclustering algorithm may be overlapping which is not the case in random clustering used for the calculation of z-score. Thus z-score has not been calculated for biclusters in this article.

A higher value of z indicates that genes would be better clustered by function, indicating a more biologically relevant clustering result. Thus, higher z-score indicates the resulting clusters being more homogenous. Fig. (**3**) shows that z-scores corresponding to CLICK is larger than that corresponding to other algorithms, for YCCD and SPTD. This shows that the results obtained by CLICK are more biologically relevant than that generated by the others.

4.4. Functional Enrichment in Terms of P-Value

One of the main drawbacks of z-score is that it evaluates the score of a set of clusters obtained by an algorithm by comparing the score of a random clustering with uniformly sized clusters, even when true clusters are not uniformly sized. The performance of the algorithms has also been compared with respect to biological significance using P - value. To compute P - value, we have employed the software Func Associate [68]. A clustering solution can be considered more re- liable if the number of functional categories obtained from a high cluster. It signifies that most of the genes with the same functional categories have been assigned in the same cluster, i.e., co-functional genes are included in the same cluster. Only functional categories with P -value $< 5.0 \times 10^{-7}$ are reported as enriched functional categories. Higher the number of functionally enriched attributes found per cluster, better is the clustering. Here, we have considered the best cluster and the three best clusters obtained by each clustering and biclustering algorithm.

Figs. (**4a-4e**) show numbers of functionally enriched attributes in three most enriched clusters for each of the clustering and biclustering algorithms on all the five datasets. P -values corresponding to all the enriched functional categories in the most enriched cluster for all the considered clustering and biclustering algorithms on YCCD are shown in Fig. (**5**). From Figs. (**4a-4e**) and Fig. (**5**), it is clear that the reliability of the results corresponding to CLICK for all the five datasets is higher than that corresponding to the other algorithms.

Among the biclustering algorithms, SAMBA is able to obtain better results compared to the other biclustering algorithms. Figs. (**44a-4e**) and Fig. (**5**) depict that SAMBA outperforms all the clustering and biclustering algorithms in terms of obtaining functionally enriched groups for all the three best clusters.

4.5. Significant Transcription Factors in Terms of P-Values

We have considered PRIMA available in EXPANDER [36] for analysis of transcription factor binding sites corresponding to the clusters and biclusters. Similar to the analysis on functional enrichment, here we have compared performance of different algorithms by number of enriched transcription factors in clusters or biclusters. Higher

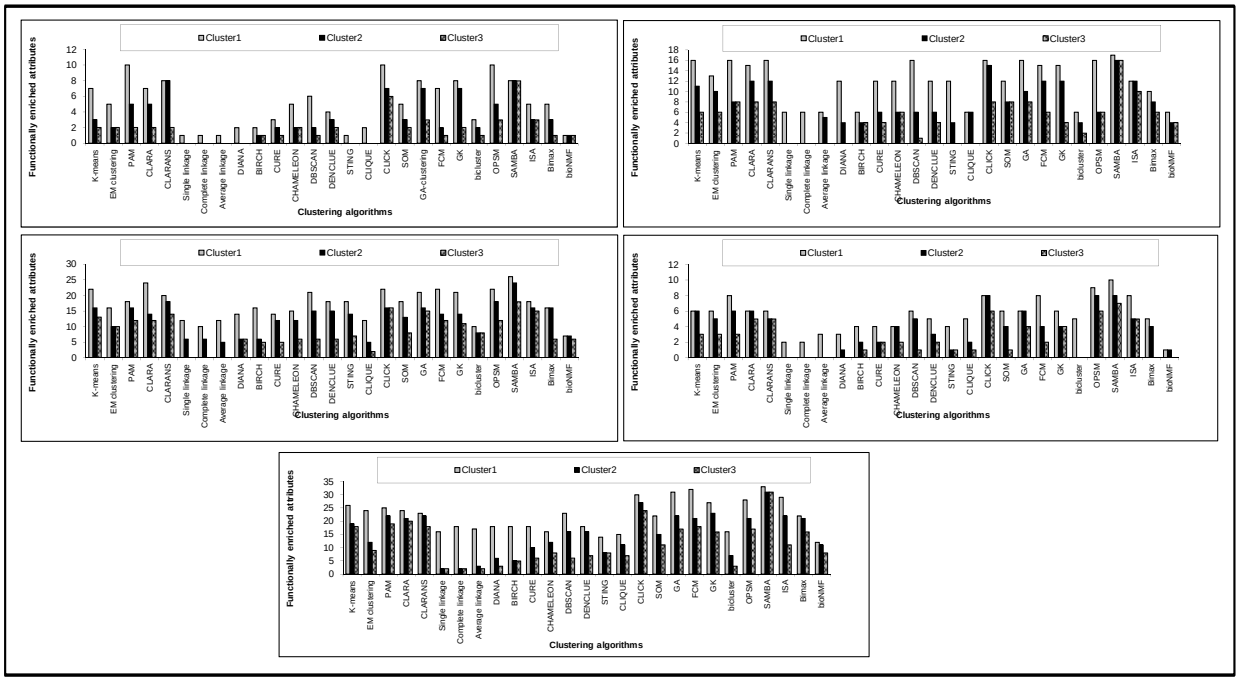

Fig. (4). Comparative results on functionally enriched attributes from the three most enriched clusters for clustering algorithms. (**a**) YCCD, (**b**) SPTD, (**c**) GDS958, (**d**) GDS2547, (**e**) GDS2938.

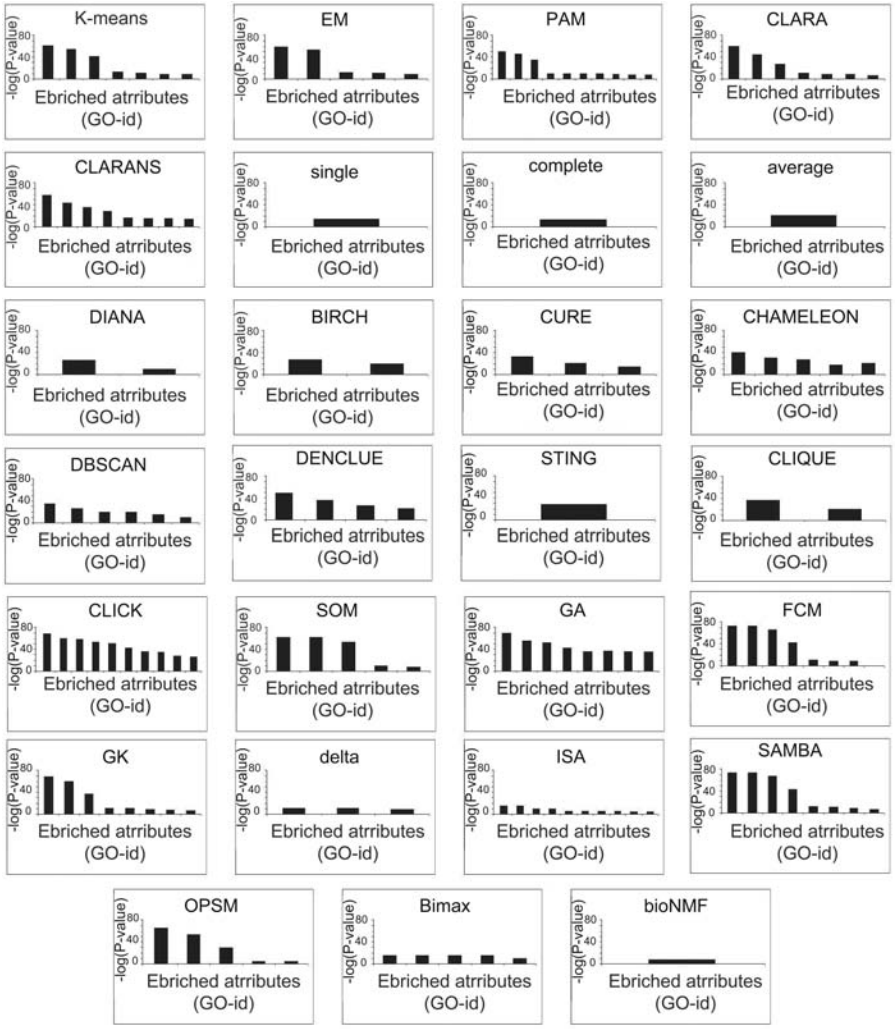

Fig. (5). Comparison of log P-values corresponding to functionally enriched attributes from the most enriched clusters for clustering algorithms on dataset YCCD.

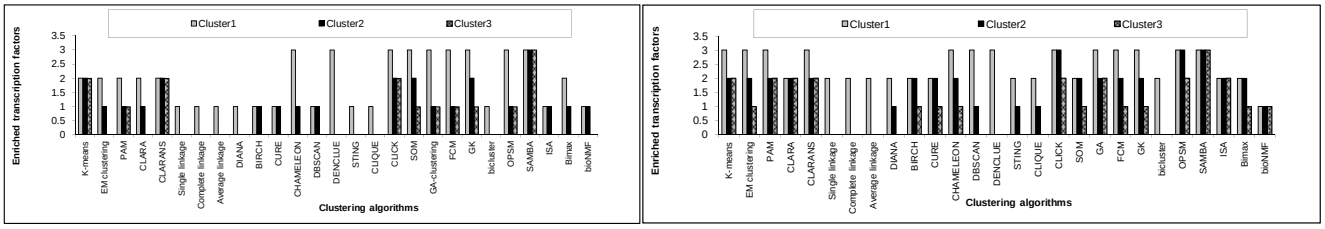

Fig. (6). Comparative results on transcription factors on Yeast datasets from three most enriched clusters for clustering algorithms. (**a**) YCCD, (**b**) SPTD.

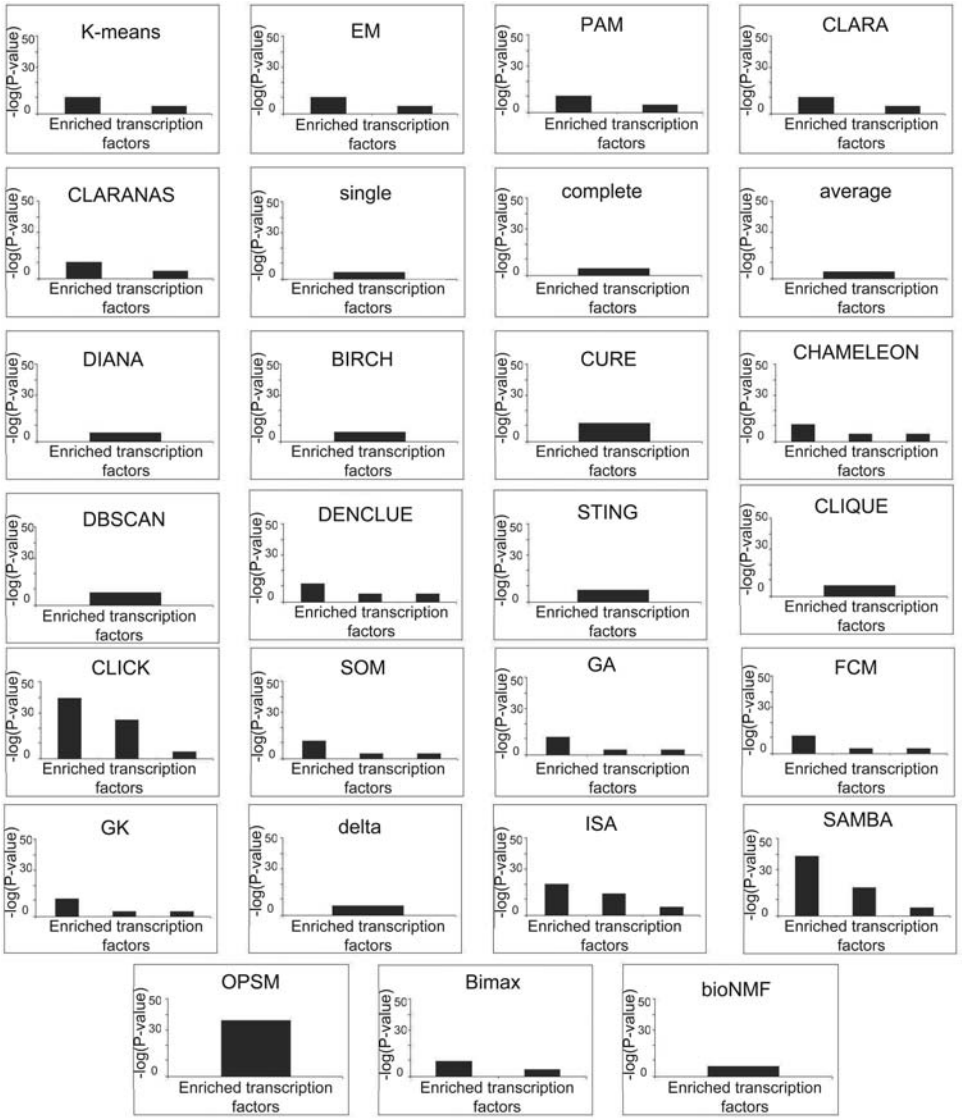

Fig. (7). Comparison of log P -values corresponding to enriched transcription factors from the most enriched clusters for clustering algorithms on dataset YCCD.

the number of enriched transcription factors in clusters or biclusters, better is the chance of finding co-regulated groups of genes. Only transcription factors with P-value $< 1.0 \times 10^{-4}$ are reported as significant.

Figs. (**6a**) and (**6b**) show numbers of enriched transcription factors in three most enriched clusters for each of the clustering and biclustering algorithms on two yeast datasets. Fig. (**7**) shows the P-values corresponding to all the

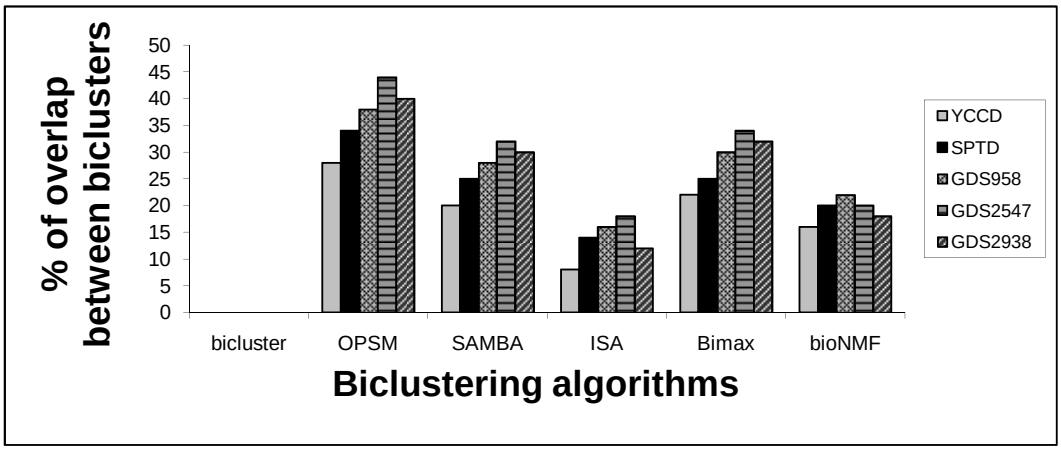

Fig. (8). Percentage of overlap between biclusters ob- tained by the biclustering algorithms.

enriched transcription factors in the most enriched cluster for all the considered clustering and biclustering algorithms on YCCD. All these figures show that CLICK, DBSCAN, DENCLUE, SOM, GA-clustering, FCM, GK and all the partition- ing clustering algorithms are able to obtain more enriched transcription factors compared to the other clustering algorithms. Regarding the comparison among biclustering algorithms, SAMBA again outperforms other algorithms in finding biclusters with enriched transcription factors for all the three best clusters. Comparing the performances of all the clustering and biclustering algorithms in Figs. (**6a-6b**) and Fig. (**7**), it is evident that SAMBA produces results with more enriched transcription factors compared to all the considered clustering and biclustering algorithms.

4.6. Handling Overlapping Biclusters

Number of functionally enriched attributes in a bi- clustering result depends on the number of biclusters and overlap between the biclusters. In the present analysis, three most enriched clusters obtained by clustering/biclustering algorithms have been compared. A comparison on overlap between biclusters has been done to understand the significance of the biclustering results. Here, we count the number of common genes between two biclusters. This number is averaged over all pairs of biclusters obtained by an algorithm. This average, in terms of percentage, gives a measure to reflect the extent of overlap between a pair of biclusters obtained by an algorithm.

Fig. (**8**) shows the percentage of overlap between all pairs of biclusters for each of the five gene expression dataset and for each biclustering algorithm. Five separate colors are used for representing the over- lapping percentage for five gene expression datasets. From Fig. (**8**), it is clear that the percentage of over- lap between the biclusters produced by SAMBA is comparable to Bimax and bioNMF. The biclustering algorithm deltabicluster produces biclusters without any overlap, and OPSM produces biclusters with more overlap compared to the other algorithms.

4.7. Time Complexity

Upper bound of the execution time of the agglomerative algorithms with single linkage, complete linkage, and average linkage, DIANA and CURE is $O(n^2 \log n)$, while that for PAM, CLARANS, CHAMELEON, DB- SCAN, DENCLUE, GA clustering and CLICK is $O(n^2)$. For BIRCH, K-means, EM Clustering, CLARA, STING, SOM, Fuzzy c-means (FCM) and GK, upper bound is O(n) [2, 17, 18, 20, 27, 28, 31, 32, 58]. Upper bound of the execution time of CLIQUE is $O(c^m + n \times m)$ [25] (Table **1**).

For the algorithm of Cheng and Church, upper bound of the execution time for a single iteration is O(nm) [11]; while that for OPSM is $O(nm^3 l)$, where l is the number of biclusters [35]. SAMBA has the time complexity of $O(n^2 d)$, where d is the upper bound on the degree of each vertex [36]. The running time complexity of Bimax is $O(nm^2)$ [14] (Table **1**). Although, nominally ISA runtime scales linearly with number of genes and samples [37, 38], it scales linearly with the number of seeds, which, to get a good clustering, is required to be larger for a bigger set of data.

5. CONCLUSION

Here we have performed five different comparative analyses of clustering and biclustering algorithms. Fig. (**3**) shows comparison of z-score for the clustering algorithms, while Fig. (**4**) shows comparative results on number of functionally enriched attribute for three most enriched clusters and Fig. (**5**) presents a comparison of log P -values corresponding to functionally enriched attributes for the most enriched clusters obtained from the clustering algorithms. Similar to Figs. (**4** and **5**), Figs. (**6** and **7**) represent comparative results on number of enriched transcription factors for three most enriched clusters and a comparison of log P -values corresponding to enriched transcription factors obtained from the most enriched clusters re- spectively. In the analysis of results, we are looking for the algorithm which shows better performance consistently in all the five comparisons, and select it as a winning clustering algorithm and then look into its features which potentially make the algorithm performing better than the others.

Analysis of the results on different clustering algorithms suggests that CLICK able to obtain more bio- logically significant clusters compared to the others. The two main reasons behind the success of CLICK are the following. The algorithm performs clustering without making any prior assumptions on the structure or the number of the clusters,

but emphasizes on relationships between the relationships between all the genes in a dataset by presenting such relations as a weighted proximity graph. CLICK performs a recursive graph partitioning to retain all the relations in subgraphs obtained from the initial proximity graph using minimum cut computations. The edge weights and the stopping criterion of the recursion are assigned with specific significance. This makes the algorithm providing re- sults with high accuracy.

Common problems with hierarchical clustering methods are that they may form single large cluster and several singleton clusters. Moreover, hierarchical algorithms cannot repair defects occurred in a clustering step to produce proper clustering solution. But the advantages of hierarchical clustering algorithms are that unlike partitioning clustering algorithms, they are able to generate clusters without taking the number of clusters as an input. In our analysis, hierarchical clustering algorithm like CURE is able to show better performance compared to linkage based algorithms of that category. CURE uses multiple cluster representatives that allow it to assign elements to clusters with higher accuracy than that by the other hierarchical algorithms (i.e., single linkage, complete linkage, DIANA) that use single cluster representative. CHAMELEON, a hierarchical clustering algorithm, also performs better than other linkage based hierarchical clustering algorithms (i.e., single linkage, complete linkage, DIANA) as they consider aggregate inter connectivity of the objects in two different clusters during their assignment to the clusters.

Different biclustering algorithms were also com- pared here. Results show that SAMBA has outperformed the other algorithms. This is because of its ability to detect clusters of genes with similar changes in gene expression values more precisely compared to the others. Biclustering algorithms that are able to identify groups of genes with similar expression values, namely, deltabicluster, produce results with less biological significance.

From the analysis, it may be concluded that a good clustering algorithm performs clustering without making any prior assumptions on the structure or the number of the clusters. It should consider relationships between the objects (genes) in the input dataset and form clusters by assigning all the genes with similar changes in gene expression values to the same cluster.

APPENDIX A

Figure of Merit (Fom)

Consider a clustering result of K disjoint clusters C_1, C_2,. . . , C_K of n genes based on samples 1, . . . , (e−1), (e + 1), . . . , m where an arbitrary sample e is left out. The FOM with respect to e and the number of clusters K is defined as

$$FOM(e,K) = \sqrt{\frac{1}{n} \times \sum_{i=1}^{K} \sum_{g \in C_i} (R(g,e) - \mu_{C_i}(e))^2}, \quad (5)$$

where R(g, e) is the expression value of a gene g for a sample e and $\mu_{C_i}(e)$ is the average expression value of the genes in cluster C_i for e. The FOM with respect to all the samples is computed as

$$FOM(K) = \sum_{e=1}^{m} FOM(e,K) \quad (6)$$

and it is used as a cluster validity index. Lower the value of FOM, better is the clustering results.

APPENDIX B

Preprocessing and Proximity Measurement

Here we briefly describe preprocessing [53, 69] techniques of gene expression data and different proximity measurement for computing similarity between gene expression patterns.

B.1. Preprocessing of Gene Expression Data

Preprocessing [53, 69] involves a group of procedures that need to be applied to the gene expression data prior to the analysis. Missing value prediction in gene expression datasets, representation of gene expression data in a suitable form (e.g., intensity ratio, log ratio or fold change), handling replication, handling outliers and filtering data are the main preprocessing tasks [53, 69].

After preprocessing of gene expression data, a nor- malization step [53, 70-72] is carried out to remove the non-biological influences (e.g., differences in labeling efficiency, scanner malfunction and uneven hybridization) on microarray experiments, which affect the measurement of expression levels. Here normalization means standardization and centralization. Standardization is the process of expanding or contracting the distribution of a statistic so that the experimental values can be compared with those generated from another experiment [53]. The process centralization is meant for moving a distribution so that it is centered around the expected mean [53].

B.2. Proximity Measurement for Gene Expression Patterns

Proximity measurement determines the similarity (or distance) between two data objects. In the case of grouping of genes, each gene is considered as a data object. The dimension of such an object is equal to the number of microarray experiments through which an expression for a gene is obtained. That is, expression values of a gene in various microarray experiments correspond to its feature values. Euclidean distance is one of the most commonly-used measure that determines the distance between two data objects. Euclidean distance between two objects x_i and x_j in m-dimensional space is defined as

$$Euclidean(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{\sum_{l=1}^{m} (\mathbf{x_{il}} - \mathbf{x_{jl}})^2} \quad (7)$$

Mahalanobis distance can also be used as a distance measure between two data objects x_i and x_j in m- dimensional space and it is defined as

$$M(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{(\mathbf{x_i} - \mathbf{x_j})^T S^{-1} (\mathbf{x_i} - \mathbf{x_j})}, \quad (8)$$

where S is the covariance matrix of the data. If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance. The Euclidean

norm-based methods find mainly spherical shaped clusters [33], while methods based on Mahalanobis distance detect mainly ellipsoidal ones [33], even if these shapes of clusters may not be present in a dataset.

An alternate measure is Pearson correlation co- efficient that determines the similarity/dissimilarity between expression patterns of two genes x_i and x_j. Pearson correlation coefficient is defined as

$$Corr(x_i, x_j) = \frac{\sum_{l=1}^{m} (x_{il} - \bar{X}_i)(x_{jl} - \bar{X}_j)}{\sqrt{\sum_{l=1}^{m} (x_{il} - \bar{X}_i)^2 \sum_{l=1}^{m} (x_{jl} - \bar{X}_j)^2}} \quad (9)$$

where x_{il} and x_{jl} are l_{th} sample values of i_{th} and j_{th} genes respectively. The terms $\bar{x}_i$ and $\bar{x}_j$ are mean values obtained from m samples of i_{th} and j_{th} genes respectively. Pearson correlation coefficient uses m sample values of a pair of genes x_i and x_j , and returns a value lying between +1 and −1. C orr(x_i , x_j) > 0 represents that x_i and x_j are positively correlated with its magnitude as the degree of correlation. On the other hand, C orr(x_i , x_j) < 0 signifies that x_i and x_j are negatively correlated with the degree of correlation as |C orr(x_i , x_j)|. C orr(x_i , x_j) = 0 indicates that these two genes are independent. Positive value of Pearson correlation coefficient indicates that the two genes are co-expressed and negative value indicates that opposite expression pattern exists between them.

With this measure, genes with low and high expression values may be in the same cluster provided that the pattern of changes in expression values over the samples of two genes is similar.

CONFLICT OF INTEREST

Declared none.

ACKNOWLEDGEMENT

Deaclared none.

REFERENCES

[1] Jain AK, Dubes RC. Algorithms for Clustering Data. New Jersey: Prentice Hall 1988.

[2] Han J, Kamber M. Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann 2001.

[3] Eisen MB, Spellman PT, Brown PO, *et al.* Cluster analysis and display of genome-wide expression patterns. PNAS 1998; 95: 14 863-14 868.

[4] Hughes TR, Marton MJ, Jones AR, *et al*. Functional discovery via a compendium of expression profiles. Cell 2000; 102: 109-126.

[5] Wu LF, Hughes TR, Davierwala AP, *et al.* Large-scale predic- tion of Saccharomyces cerevisiae gene function using overlap- ping transcriptional clusters. Nat Genet 2002; 31(3): 255-265.

[6] Madeira SC, Oliveira AL. Biclustering algorithms for bio- logical data analysis: A survey. IEEE Trans Comput Biol Bioinform 2004; 1(1): 24-45.

[7] Yeung KY, Medvedovic M, Bumgarner RE. From co-expression to co-regulation: How many microarray experiments do we need?. Genome Biol 2004; 5: R48.

[8] Wyrick JJ Young RA. Deciphering gene expression regulatory networks. Curr Opin Genet Dev 2002; 12: 130-136.

[9] Spellman PT, Zhang GS, Iyer VR, *et al.* Comprehensive identi- fication of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol Biol Cell 1998; 9: 3273-3297.

[10] Allocco DJ, Kohane IS, Butte AJ. Quantifying the relationship between co-expression, co-regulation and gene function. BMC Bioinformatics 2004; 5: 18.

[11] Cheng Y, Church GM. Biclustering of expression data. Pro- ceedings of the 8th International Conference on Intelligent Systems for Molecular Biology; 2000 August 19-23; San Diego, USA; pp. 93-103.

[12] Jiang D, Tang C, Zhang A. Cluster analysis for gene expression data: A survey. IEEE Trans Knowl Data Eng 2004; 16: 1370-1386.

[13] Tanay A, Sharan R, Shamir R. Handbook of Computational Molecular Biology, ser. Computer and Information Science. Chapman and Hall/CRC, 2005, ch. Biclustering Algorithms: A Survey.

[14] Prelic A, Bleuler S, Zimmermann P, *et al.* A systematic com- parison and evaluation of biclustering methods for gene ex- pression data. Bioinformatics 2006; 22: 1122-1129.

[15] Tavazoie S, Hughes JD, Campbell MJ, *et al.* Systematic de- termination of genetic network architecture. Nat Genet 1999; 22(3): 281-285.

[16] Fraley C, Raftery AE. Model-based clustering, discriminant analysis and density estimation. J Amer Statistical Assoc 2002; 97: 611-631.

[17] Zhang Y, Luxon BA, Casola A, *et al.* Expression of respiratory syncytial virus-induced chemokine gene networks in lower airway epithelial cells revealed by cDNA microarrays. J Virol 2001; 75: 9044-9058.

[18] Datta S, Datta S. Evaluation of clustering algorithms for gene expression data. BMC Bioinformatics 2006; 7: s17.

[19] Zhang T, Ramakrishnan R, Livny M. BIRCH: A new data clustering algorithm and its applications. Data Min Knowl Disc 1997; 1: 141-182.

[20] Guha S, Rastogi S, Shim K. CURE: An efficient clustering algo- rithm for large databases. Information Systems 2001; 26: 35-58.

[21] Karypis G, Han EH, Kumar V. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. IEEE Computer 1999; 32(8): 68-75.

[22] Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the 2nd International Conference on Knowl- edge Discovery and Data Mining; 1996 August 2-4; Portland, USA; pp. 226-231.

[23] Hinneburg A, Keim DA. An efficient approach to clustering in multimedia databases with noise. Proceedings of the Interna- tional Conference on Knowledge Discovery and Data Mining; 1998 August 27-31; New York, USA; pp. 58-65.

[24] Wang W, Yang J, Muntz R. STING: A statistical information grid approach to spatialdata mining. Proceedings of the 23rd International Conference on Very Large Data Bases; 1997 August 25-29; Athens, Greece; pp. 186-195.

[25] Agrawal R, Gehrke J, Gunopulos D. Automatic subspace clus- tering of high dimensional data for data mining applications. Proceedings of the ACM SIGMOD International Conference on Management of Data; 1998 June 2-4; Seattle, USA; pp. 94-105.

[26] Shamir R, Sharan R. Click: A clustering algorithm for gene expression analysis. Proceedings of the 8th International Con- ference on Intelligent Systems for Molecular Biology; 2000 August 19-23; San Diego, USA; pp. 307-316.

[27] Reich M, Ohm K, Angelo M, Tamayo P. GeneCluster 2.0: An advanced toolset for bioarray analysis. Bioinformatics 2004; 20: 1797-1798.

[28] Maulik U, Bandyopadhyay S. Genetic algorithm-based clus- tering technique. Pattern Recog 2000; 33: 1455-1465.

[29] Dunn JC. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J Cybernet 1973; 3: 32-57.

[30] Bezdek JC. Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum Press, 1981.

[31] Bezdek JC, Ehrlich R, Full W. FCM: The fuzzy c-means clus- tering algorithm. Comput Geosci 1984; 10: 191-203.

[32] Gustafson E, Kessel WC. Fuzzy clustering with a fuzzy covari- ence matrix. Proceedings of the IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Pro- cesses; 1978 January 10-12; San Diego, USA; pp. 761-766.

[33] Kim DW, Lee KH, Lee D. Detecting clusters of different geometrical shapes in microarray gene expression data. Bioin- formatics 2005; 21: 1927-1934.

[34] Pascual-Montano A, Carmona-Saez P, Chagoyen M, *et al.* bioNMF: A versatile tool for non-negative matrix factorization in biology. BMC Bioinformatics 2006; 7: 366.

[35] Ben-Dor A, Chor B, Karp R, *et al.* Discovering local structure in gene expression data: The order-preserving submatrix prob- lem. J Comput Biol 2003; 10: 373-384.
[36] Tanay A, Sharan R, Shamir R. Discovering statistically signif- icant biclusters in gene expression data. Bioinformatics 2002; 18: S136-S144.
[37] Ihmels J, Friedlander G, Bergmann S, *et al.* Revealing modular organization in the yeast transcriptional network. Nat Genet 2002; 31(4): 370-377.
[38] Ihmels J, Bergmann S, Barkai N. Defining transcription mod- ules using large-scale gene expression data. Bioinformatics 2004; 20: 1993-2003.
[39] Bø TH, Dysvik B, Jonassen I, Lsimpute: Accurate estimation of missing values in microarray data with least squares methods. Nucleic Acids Res 2004; 32: E34.
[40] Loganantharaj R, Cheepala S, Clifford J. Metric for measuring the effectiveness of clustering of DNA microarray expression. BMC Bioinformatics 2006; 7: s5.
[41] Mitchell T. Machine Learning. New York: McGraw-Hill, 1997.
[42] Mukhopadhyay A, Maulik U, Bandyopadhyay S. On bicluster- ing of gene expression data. Curr Bioinform 2010; 5(3): 204-216.
[43] Hartigan JA. Direct clustering of a data matrix. J Am Stat Assoc 1972; 67: 123-129.
[44] Getz G, Levine E, Domany E. Coupled two-way clustering analysis of gene microarray data. PNAS 2000; 97: 12 079-12 084.
[45] Kluger Y, Basri R, Chang JT, *et al.* Spectral biclustering of microarray cancer data: Co-clustering genes and conditions. Genome Res 2003; 13: 703-716.
[46] Lazzeroni L, Owen A. Plaid models for gene expression data. Stat Sinica 2002; 12: 61-86.
[47] Murali TM, Kasif S. Extracting conserved gene expression motifs from gene expression data. Proceedings of the Pacific Symposium on Biocomputing; 2003 January 3-7; Hawaii, USA; pp. 77-88.
[48] Kim PM, Tidor B. Subsystem identification through di- mensionality reduction of large-scale gene expression data. Genome Res 2003; 13: 1706-1718.
[49] Gyenesei A, Wagner U, Barkow-Oesterreicher S, *et al.* Mining co-regulated gene profiles for the detection of functional asso- ciations in gene expression data. Bioinformatics 2007; 23: 1927-1935.
[50] Yeung KY, Haynor DR, Ruzzo WL, *et al.* Validating clustering for gene expression data. Bioinformatics 2001; 17(4): 309-318.
[51] Alberts B, Johnson A, Lewis J, *et al.* Molecular Biology of the Cell. 5th ed. New York: Garland Science 2008.
[52] Pham TD, Wells C, Crane DT. Analysis of microarray gene expression data. Curr Bioinform 2006; 1(1): 37-53.
[53] Pasanen T, Saarela J, Saarikko I, *et al.* DNA Microarray Data Analysis. Helsinki: Scientific Computing Ltd. 2003.
[54] Bergmann S, Ihmels J, Barkai N. Iterative signature algorithm for the analysis of large-scale gene expression data. Physical Review E 2003; 67: 031902.
[55] Teng L, Chan LW. Biclustering gene expression profiles by alternately sorting with eighted correlated coefficient. Proceedings of the 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing; 2006 September 6-8; Arlington, USA; pp. 289-294.
[56] Dembele D, Kastner P. Fuzzy c-means method for clustering microarray data. Bioinformatics 2003; 19: 973-980.
[57] D'haeseleer P, Liang S, Somogyi R. Genetic network inference: From co expression clustering to reverse engineering. Bioinformatics 2000; 16: 707-726.
[58] Sharan R, Maron-Katz A, Shamir R. Click and Expander: A system for clustering and visualizing gene expression data. Bioinformatics 2003; 19: 1787-1799.
[59] Gibbons F, Roth F. Judging the quality of gene expression- based clustering methods using gene annotation. Genome Res 2002; 12: 1574-1581.
[60] Press W, Flannery B, Teukolsky S, *et al.* Numerical Recipies in C - The Art of Scientific Computing. 2nd ed. New Delhi: Cambridge University Press India Pvt. Ltd. 2009.
[61] Jakt LM, Cao L, Cheah KS, *et al.* Assessing clusters and motifs from gene expression data. Genome Res 2001; 11: 112-123.
[62] Wills-Karp M, Ewart SL. Time to draw breath: Asthma-susceptibility genes are identified. Nature Rev Genet 2004; 5(5)376-87.
[63] Chandran UR, Ma C, Dhir R, *et al.* Gene expression profiles of prostate cancer reveal involvement of multiple molecular pathways in the metastatic process. BMC Cancer 2007; 7: 64.
[64] Yu Y, Landsittel D, Jing L, *et al.* Gene expression alterations in prostate cancer predicting tumor aggression and preceding development of malignancy. Am J Clin Oncol 2004; 22: 2790-2799.
[65] Wang S, Antwerp MV, Kuick R, *et al.* Microarray analysis of cytokine activation of apoptosis pathways in the thyroid. Endocrinology 2007; 10: 4844-4852.
[66] Ashburner M, Ball CA, Blake JA, *et al.* Tool for the unification of biology: The gene ontology consortium. Nat Genet 2000; 25(1): 25-29.
[67] Issel-Tarver L, Christie K, Dolinski K, *et al.* Saccharomyces genome database. Method Enzymol 2002; 350: 329-346.
[68] Berriz FG, King OD, Bryant B, *et al.* Characterizing gene sets with FuncAssociate. Bioinformatics 2003; 19: 2502-2504.
[69] Troyanskaya O,Cantor M, Sherlock G, *et al.* Missing value es- timation methods for DNA microarrays. Bioinformatics 2001; 17: 520-525.
[70] Bolstad BM, Irizarry RA, Astrand M, *et al.* A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. Bioinformatics 2003; 19: 185-193.
[71] Kim SY, Lee JW, Bae JS. Effect of data normalization on fuzzy clustering of DNA microarray data. BMC Bioinformatics 2006; 7: 134.
[72] Hill A, Brown E, Whitley M, *et al.* Evaluation of normalization procedures for oligonucleotide array data based on spiked cRNA controls. Genome Biol 2001; 2: research0055.

Received: April 11, 2011 Revised: July 15, 2011 Accepted: November 17, 2011