Journal of Biomedical Informatics 43 (2010) 560–568

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Average correlation clustering algorithm (ACCA) for grouping of co-regulated genes with similar pattern of variation in their expression values

Anindya Bhattacharya [a], Rajat K. De [b,*]

[a] Department of Computer Science and Engineering, Netaji Subhash Engineering College, Kolkata 700 152, India
[b] Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India

ARTICLE INFO

Article history:
Received 10 June 2009
Available online 6 February 2010

Keywords:
Transcription factors
Correlation clustering
P-Value
z-Score
Functional enrichment

ABSTRACT

Distance based clustering algorithms can group genes that show similar expression values under multiple experimental conditions. They are unable to identify a group of genes that have similar pattern of variation in their expression values. Previously we developed an algorithm called divisive correlation clustering algorithm (DCCA) to tackle this situation, which is based on the concept of correlation clustering. But this algorithm may also fail for certain cases. In order to overcome these situations, we propose a new clustering algorithm, called average correlation clustering algorithm (ACCA), which is able to produce better clustering solution than that produced by some others. ACCA is able to find groups of genes having more common transcription factors and similar pattern of variation in their expression values. Moreover, ACCA is more efficient than DCCA with respect to the time of execution. Like DCCA, we use the concept of correlation clustering concept introduced by Bansal et al. ACCA uses the correlation matrix in such a way that all genes in a cluster have the highest average correlation values with the genes in that cluster. We have applied ACCA and some well-known conventional methods including DCCA to two artificial and nine gene expression datasets, and compared the performance of the algorithms. The clustering results of ACCA are found to be more significantly relevant to the biological annotations than those of the other methods. Analysis of the results show the superiority of ACCA over some others in determining a group of genes having more common transcription factors and with similar pattern of variation in their expression profiles.

Availability of the software: The software 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.

1. Introduction

Clustering is a process of organizing objects into groups where members in a group are similar and those in different groups are dissimilar. Possible similarity measures include correlation, Euclidean distance, Mahalanobis distance and the angle between vectors of observations. Most of the conventional clustering techniques use Euclidean and Mahalanobis distances for determining *similarity*/*dissimilarity* between a pair of objects and decide whether they belong to the same or different clusters. Some of the problems with these methods are as follows: (i) They find clusters of co-expressed genes, but unable to determine a group of genes having similar pattern of variations in the expression values. According to Heyer et al. [27], measuring co-expression can be done directly without any assumptions concerning gene function or regulation. Most measures scored curves with similar expression patterns well, but often gave high scores to dissimilar curves. The correlation coefficient performed better than the other measures. (ii) For large datasets, these algorithms may result in large miss clustering. Moreover, clustering algorithms like AGNES [25,50] or DIANA [25,13] may result in one single large cluster and several singletons.

Several clustering algorithms have been developed. They include development of a new framework for representing a set of multidimensional gene expression data as a minimum spanning tree (MST) [47]; a clustering method for microarray gene expression data for detecting clusters of different shapes in a dataset [32,24]; a new clustering algorithm, called CLICK, based on graph-theoretic and statistical techniques to identify tight groups (kernels) of highly similar genes that are likely to belong to the same true cluster [42]. Several heuristic procedures, viz., kernel hierarchical clustering algorithm [39]; algorithm for clustering of gene expression data based on the notion of simulated annealing [36]; a method for selecting parameters for Fuzzy c-means algorithm in relation to gene expression data clustering [16], have been developed.

* Corresponding author. Fax: +91 33 25783357.
E-mail addresses: anindyamail@rediffmail.com (A. Bhattacharya), rajat@isical.ac.in (R.K. De).

1532-0464/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2010.02.001

There also exist several biclustering algorithms [26,11,20,33,48,49,46,28]. Biclustering is a technique that performs simultaneous grouping on genes and conditions (measurements) of a dataset to determine subgroups of genes that exhibit similar behavior over a subset of experimental conditions (measurements). The technique was originally introduced by Hartigan [26] and first applied on gene expression data by Cheng and Church [11].

Bansal et al. [4] have introduced the concept of correlation clustering that is based on the notion of graph partitioning. The methodology involves construction of a graph from input data by considering genes as nodes and correlation between the genes as edges. There are two types of edges, viz., *positive* and *negative*. If the correlation coefficient between two genes is positive, there is a *positive* edge between the nodes. On the other hand, a *negative* edge between these two nodes indicates that the corresponding genes are negatively correlated. Number of *agreements* is simply the number of data points (genes) that are put in correct clusters, and is measured by the number of *positive* edges in the same clusters plus that of *negative* edges between genes in different clusters. The *positive* edges between genes indicate that they are in the same cluster. On the other hand, the number of *disagreements* is the number of genes wrongly clustered, and is measured by the number of *negative* edges in the same clusters plus number of *positive* edges between nodes in different clusters.

In the area of correlation clustering, several attempts [1,9,10,15,14] have already been made, which deal with variations of this method. If there exists a perfect clustering, i.e., if one gets all the genes correctly clustered, then the optimal clustering solution can be obtained by simply deleting all *negative* edges and output the connected components of the remaining graph [12]. It has been proved that if no perfect clustering exists, no algorithm, based on correlation coefficient can find an optimal clustering results in polynomial time [4]. There are two equivalent approaches [4] for correlation clustering. One approach is based on minimization of *disagreement* while the other is on maximization of *agreement*.

Bansal et al. have proved that the problem of minimizing *disagreement* or equivalently maximizing *agreement* is NP-complete [4]. They have provided a constant factor approximation algorithm to the problem of minimizing *disagreements*, and a polynomial-time approximation scheme (PTAS) for maximizing *agreements* [4]. Both these algorithms are based on graph partitioning. Main problems of these two algorithms are that they can only work on a given unweighted complete graph with *positive*/*negative* labels on the edges and they have considered only sign of the correlation coefficient but not the magnitude. Genes with large and positive correlation values among them are likely to be associated with the same biological functions. As both the pairs (i.e., genes with large and positive correlation values, and genes with small and positive correlation values) are represented by unweighted positive edges and treated as the same by algorithms, this may deteriorate the quality of clusters in terms of biological relevance. Another major problem with them is that they are able to obtain clustering solution if and only if there exists at least one *negative* edge. If the input dataset contains a set of data points such that all the pairs of points are only positively correlated, i.e., have only *positive* edges between them, all the previous correlation clustering algorithms [4,1,9,10,15,14] including divisive correlation clustering algorithm (DCCA) [8], a recently developed correlation clustering algorithm by the authors, fail to obtain clustering.

In order to tackle these problems with the aforesaid correlation clustering algorithms, we have considered both sign and magnitude of the correlation coefficient. Based on this notion, we have developed, in this paper, a new clustering algorithm, called *average correlation clustering algorithm* (ACCA). This is a partitional clustering method. ACCA uses Pearson correlation coefficient [23] as the similarity measure. The algorithm is based on concepts of correlation clustering but it differs from that in [4].

Among different definitions of inter-cluster distances, more common are single, complete and average linkage. Complete and single-linkage are extreme procedures with completely different properties. Complete-linkage uses the similarity between the furthest pair of objects from two clusters. In contrast to these requirements, single-linkage only uses the nearest pair of objects from each cluster. Both methods have an extreme conception of homogeneity of a cluster. Single-linkage leads to grouping and may result in a few large and heterogeneous clusters [18]. Complete-linkage results in dilatation and may produce many clusters, being more suitable for isolating poorly separated clusters [22]. Average-linkage tries to avoid these effects by computing the average. There exist investigations to show average linkage is better than single, complete and centroid linkages in discovering clusters with less number of miss clustering [41,2,34,45], although Jain et al. have argued in [31] to show complete linkage is the best in not discovering false clusters. Following [41,2,34,45], we have considered average similarity in ACCA, as in the case of average linkage in hierarchical clustering; in determining the initial members of clusters during the cluster update process.

ACCA is able to detect clusters of genes having more common transcription factors and with similar variation in their expression values. Regarding time complexity, ACCA is more efficient than DCCA. Note that unlike other correlation clustering algorithms including DCCA, ACCA is able to handle the situation where there is no negative edge. ACCA initially creates K random clusters. With an initial set of clusters, ACCA iterates until it is able to create a set of clusters where each gene belonging to each clusters has the highest similarity in terms of expression pattern with other genes inside the cluster. Algorithm terminates with such K clusters.

The superior capability of clustering by ACCA, in terms of biological significance, over a number of algorithms, viz., Bansal's minimizing disagreement (MIND) in [4], K-means [30,25,44] with euclidian distance measure (K-means (Euclidian)), K-means [30,25,35] with Pearson's correlation coefficient as distance measure (K-means (Pearson)), PAM [25], DIANA [25,13], Fuzzy c-means (FCM) [17,6,7], GK [24], EM clustering [19], SOM [40], GA clustering [37], CLICK [42], δ-biclusters [11], and DCCA [8] is demonstrated through experiments with two artificial datasets and nine gene expression datasets. Some characteristics of ACCA are also discussed.

2. Average correlation clustering algorithm (ACCA)

In this section, we describe the proposed average correlation clustering algorithm (ACCA). Let us consider a set of n genes $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, for each of which m expression values are given. These n genes will have to be grouped into K disjoint clusters $C_1, C_2, \ldots, C_p, \ldots, C_K$. ACCA uses Pearson's correlation coefficient [23] for measuring similarity/dissimilarity between expression patterns of two genes $\mathbf{x}_i$ and $\mathbf{x}_j$, which is defined as

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

where x_{il} and x_{jl} are lth sample values of the ith and jth genes, respectively. $\bar{x}_i$ and $\bar{x}_j$ are mean values obtained from m samples of the ith jth genes, respectively. Pearson correlation coefficient uses m sample values of a pair of genes $\mathbf{x}_i$ and $\mathbf{x}_j$, and returns a value lying between +1 and −1. $Corr(\mathbf{x}_i, \mathbf{x}_j) > 0$ (< 0) represents that $\mathbf{x}_i$ and $\mathbf{x}_j$ are positively (negatively) correlated with the degree of correlation as its magnitude.

Positive (negative) value of Pearson's correlation coefficient indicates that the two genes have similar (opposite) pattern of variation in their expression values.

Before describing the algorithm in details, we define the *Average correlation value* that is used in this regard.

Average correlation value: Average correlation value for a gene $\mathbf{x}_i$ with respect to cluster C_p is defined as

$$AVGC_{pi} = \frac{1}{n_p} \sum_{\substack{\mathbf{x} \in C_p \\ \mathbf{x} \neq \mathbf{x}_i}} Corr(\mathbf{x}_i, \mathbf{x}), \tag{2}$$

where n_p is the number of data points in $C_p - \{\mathbf{x}_i\}$. Thus $AVGC_{pi}$ indicates that the average correlation value for a gene $\mathbf{x}_i$ with other genes inside the cluster C_p. This value reflects the degree of inclusion of $\mathbf{x}_i$ to cluster C_p.

ACCA initially creates K clusters by random assignment of genes into these clusters. Then for a cluster C_p, we select a gene $\mathbf{x}_i$ for which $AVGC_{pi}$ is maximum over all these values corresponding to the other genes in C_p. This is done for all the randomly created clusters. Each selected gene is copied from its cluster and placed in a new cluster to create K new clusters each containing single gene. For each of the remaining genes $\mathbf{x}_k$ that are yet to be included in new clusters, we calculate AVGC_{pk}, $p = 1, 2, \ldots, K$. We place the gene $\mathbf{x}_k$ to the new cluster for which average correlation value is maximum. Thus one iteration is completed, and we get a modified and better set of K clusters. This step of creating new set of clusters from the old set iterates until for two successive iteration no changes among clusters are found, i.e., stable clustering solution is reached. In this situation, the genes in a cluster C_p have the highest $AVGC_p$-values compared to that of the other clusters. Thus the algorithm terminates with K clusters. The terms used in the algorithm are explained in Table 1.

Algorithm

Input: (i) A set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ of n genes, for each of which m expression values are given. (ii) Number of clusters K to be created.

Output: K disjoint clusters $C_1, C_2, \ldots, C_K$, so that $X = \bigcup_{p=1}^{K} C_p$.

Steps:

1. Assign randomly n genes to K clusters.
2. For each iteration, do:
   i. For each cluster C_p, $1 \leqslant p \leqslant K$, calculate average correlation value $AVGC_{pk}$ (Eq. (2)) for each $\mathbf{x}_k$ in C_p.
   ii. For each cluster C_p, $1 \leqslant p \leqslant K$, select a gene $\mathbf{x}_i$ in C_p, if $AVGC_{pi} > AVGC_{pj}$, for all $\mathbf{x}_j$ in C_p and $j \neq i$.
   iii. Place a copy of the selected gene $\mathbf{x}_i$ from pth cluster to a new pth cluster $CNEW_p$.
   iv. For each $\mathbf{x}_k$ in $\left(\bigcup_{p=1}^{K} C_p - \bigcup_{p=1}^{K} CNEW_p\right)$, do:
      a. For each cluster $CNEW_p$, $1 \leqslant p \leqslant K$, calculate average correlation value $AVGC_{pk}$ (Eq. (2)).
      b. If $AVGC_{pk} > AVGC_{qk}$, for each q, $1 \leqslant q \leqslant K$, and $q \neq p$ then place a copy of $\mathbf{x}_k$ to a new pth cluster $CNEW_p$.
   v. If $\bigcup_{p=1}^{K} (CNEW_p - C_p) = \phi$ then no change occurs in the clusters obtained in the previous iteration of Step 2, i.e., if $CNEW_1 = C_1$, $CNEW_2 = C_2 \ldots$ and $CNEW_K = C_K$, then STOP, otherwise for each p, $1 \leqslant p \leqslant K$, set $C_p = CNEW_p$. Set $CNEW_p = \phi$, for each p and go to Step 2.

Table 1
Terms used in the algorithm ACCA.

Used term	Explanation
$\mathbf{x}_i$	ith gene in a dataset
C_p	pth cluster
$Corr(\mathbf{x}_i, \mathbf{x}_j)$	Pearson's correlation coefficient value between gene $\mathbf{x}_i$ and $\mathbf{x}_j$
$AVGC_{pi}$	Average correlation value of gene $\mathbf{x}_i$ with respect to all other elements in a cluster C_p
$CNEW_p$	New pth cluster from previous pth cluster C_p
ϕ	Null set

3. Results

The effectiveness of ACCA is demonstrated on two synthetic and nine gene expression datasets. These gene expression datasets deal with five yeasts (http://yfgdb.princeton.edu/download/) and four mammals (http://www.ncbi.nlm.nih.gov/sites/entrez). The superior performance of ACCA over other clustering algorithms, viz., Bansal's minimizing disagreement (MIND) [4], K-means (Euclidian) [30,25,44], K-means (Pearson) [30,25,35], PAM [25], DIANA [25,13], Fuzzy c-means (FCM) [17,6,7], GK [24], EM clustering [19], SOM [40], GA clustering [37], CLICK [42], δ-biclusters [11], and DCCA [8] is also observed using several indices. To compare performance of δ-biclusters [11] with other clustering algorithms we select 10 biclusters for each dataset generated by δ-biclusters. Bicluster generated by δ-biclusters [11] that contains all the genes in a dataset is not considered for selection. For comparison, we have considered $K \in [2, 10]$, K being an integer, for all the algorithms and selected the K-value for which results with the highest biological significance are obtained. We have considered the highest z-score as an indication of high biological significance for each algorithm. The values of the parameters for all these algorithms, which we have chosen, are given in Table 5 in Supplementary material. Expression profile plots have been utilized for visualizing quality of the clusters. All these indices and datasets are described briefly in Supplementary material. Moreover, a discussion on various characteristics of the algorithm is provided.

3.1. Performance comparison using synthetic data

Before going into the detailed discussion of the results on real life gene expression data, here we demonstrate superior performance of ACCA over some existing algorithms using an artificial dataset ADS_1 (Fig. 1). ADS_1 contains 115 three-dimensional samples distributed in three clusters. The dataset has been generated in such a way that the samples in three clusters are sparse in one of x, y, z dimensions, respectively. They form compact clusters in the other two directions. The samples have been generated by random members and by visual inspection. The range of the random numbers corresponding one of x, y, z directions for samples in a cluster is high, while the ranges corresponding to other two being low. Fig. 2 shows the results obtained by ACCA, DCCA, PAM, K-means (Pearson), EM clustering, GA clustering, CLICK, SOM, and GK. It is clear from Fig. 2 that ACCA, DCCA, PAM, K-means (Pearson), EM clustering, GA clustering, CLICK, SOM, and GK were able to obtain these three clusters successfully. On the other hand, MIND (Fig. 8 in Supplementary material), K-means (Euclidian) (Fig. 9 in Supplementary material), Fuzzy c-means (Fig. 10 in Supplementary material) and DIANA (Fig. 11 in Supplementary material) were unable to obtain desired clusters for the ADS_1 dataset.

Using the artificial dataset ADS_2 (Fig. 3), we demonstrate the superior performance of ACCA over DCCA, a recently developed algorithm by Bhattacharya and De [8]. ADS_2 contains 241 three-dimensional samples (genes) distributed in two clusters. The dataset has been generated in such a way that there exists no negative correlation between any two genes. The dataset is used to show the superior performance of ACCA over DCCA. Here, the samples have the values of x, y and z that form relationships $x \leqslant y \ll z$ and $x \leqslant y \leqslant z$ for clusters 1 and 2, respectively. DCCA was unable to obtain two clusters for ADS_2 dataset as there is no negative correlation present between any pair of genes. Fig. 4 shows that ACCA was able to produce proper clustering solution for the dataset ADS_2.

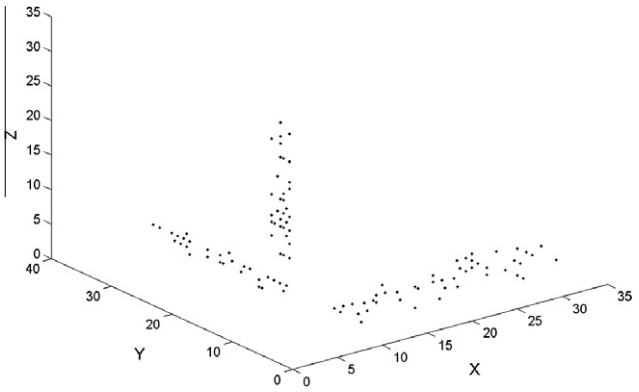

Fig. 1. Dataset ADS_1.

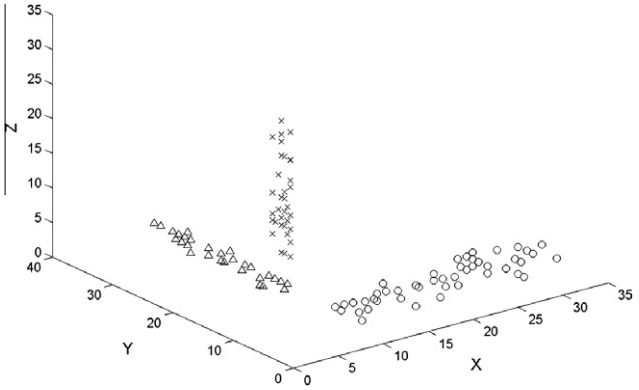

Fig. 2. Clustered output of ACCA, DCCA, PAM, K-means (Pearson), EM clustering, GA clustering, CLICK, SOM, and GK on ADS_1.

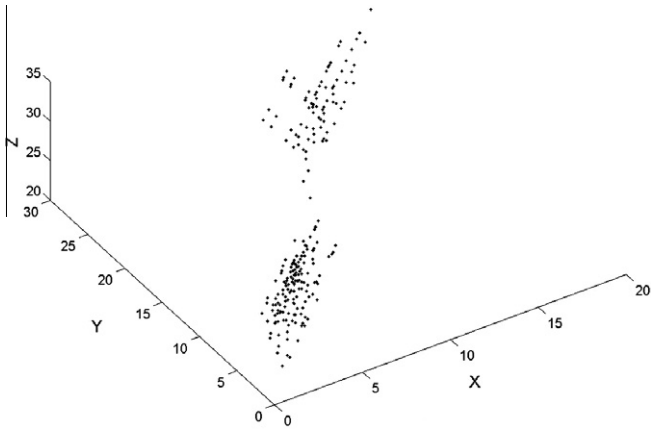

Fig. 3. Dataset ADS_2.

3.2. Performance comparison using real life data

Here we demonstrate the performance comparison of various clustering algorithms. All the clustering algorithms compared in this subsection use either Euclidean distance or Pearson's correlation coefficient as the proximity measure. When Euclidean distance is selected as proximity measure, the standardization process $x'_{il} = \frac{x_{il} - \bar{\mathbf{x}}_i}{\sigma_{\mathbf{x}_i}}$ is applied, where x_{il} is the lth sample of gene $\mathbf{x}_i$, while $\bar{\mathbf{x}}_i$ and $\sigma_{\mathbf{x}_i}$ are the mean and standard deviation of $\mathbf{x}_i$, respectively. On the other hand, this standardization process has not been applied while using Pearson correlation coefficient as the similarity measure. Under this situation, it can be proved that Euclidian distance bears a monotonic relation with Pearson's correlation coefficient. Thus, we can expect the effectiveness of a clustering algorithm to be equivalent whether Euclidean distance or Pearson's correlation coefficient is chosen as the proximity measure.

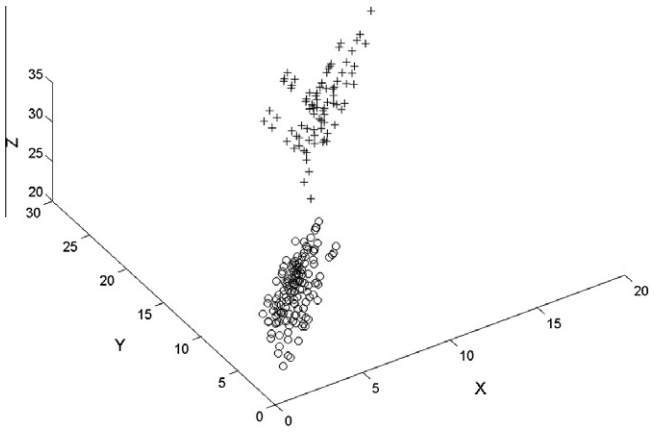

Fig. 4. Clustered output of ACCA on ADS_2.

3.2.1. Performance comparison using z-score

For performance comparison using biological information, we have used z-score. A higher value of z indicates that genes would be better clustered by functions, indicating a more biologically relevant clustering result. For comparing the results using z-score, Tables 2 and 3 show the highest z-scores corresponding to these algorithms for all the datasets considered here.

It may be mentioned here that z-score [38,21] is calculated by investigating the relation between a clustering result and the functional annotation of the genes in the cluster. To calculate z-score for five yeast datasets, Gibbons ClusterJudge [38,21] tool has been used. Saccharomyces Genome Database (SGD) annotation of the yeast genes, along with the gene ontology developed by the Gene Ontology Consortium [3,29] has been used by ClusterJudge for the calculation of z-score. ClusterJudge only supports yeast datasets. For GDS958, GDS1423 and GDS2745, corresponding annotation datasets GPL339, GPL96 and GPL97 (http://www.ncbi.nlm.nih.gov/sites/entrez) have been used. We have considered GDS958 knocked out samples and wild-type samples separately for clustering.

It has been found from Tables 2 and 3 that the best z-scores are 22.9 for ACCA (for $K = 5$), 20.44 for K-means (Euclidian) (for $K = 4$), 21.04 for PAM (for $K = 4$), 2.95 for DIANA (for $K = 8$), 19.6 for FCM (for $K = 4$), 21.78 for GK (for $K = 9$), 20.8 for K-means (Pearson) (for $K = 5$), 19.08 for EM clustering (for $K = 4$), 13.26 for SOM (for $K = 9$), 21.75 for GA clustering (for $K = 4$), 9.12 for CLICK (for $K = 9$), and -0.65 for δ-biclusters (for $K = 10$) calculated on Yeast ATP dataset. MIND produces three clusters for Yeast ATP dataset and the corresponding z-score is 4.56. Similar findings were obtained for the other datasets too.

Tables 2 and 3 show that z-scores corresponding to ACCA for $K \in [2, 10]$ (K being an integer) for these nine gene expression datasets are much higher than those of the other algorithms. Regarding the comparison between ACCA and DCCA, z-scores for the former algorithm are higher than that of the latter (Table 4). Thus the results obtained by ACCA are much more biologically relevant to that generated by the others.

3.2.2. Functional enrichment: analysis and comparison using P-values

For gene expression data analysis, P-value represents the probability of observing at least a given number of genes, in a cluster, from a specific GO functional category. A specific GO functional

Table 2
Various comparative scores on different clustering algorithms for yeast datasets.

Dataset	Method	Total clusters (K)	z-Score	Enriched clusters	Enriched attributes	Enriched transcription factors
Yeast ATP	ACCA	5	22.9	5	28	5
	MIND	3	4.56	2	2	0
	K-means (Euclidian)	4	20.44	3	14	2
	K-means (Pearson)	5	20.8	4	16	2
	PAM	4	21.04	3	16	2
	DIANA	8	2.95	0	0	0
	FCM	4	19.6	3	15	1
	GK	9	21.78	5	25	2
	EM clustering	4	19.08	2	13	2
	SOM	46	13.26	2	9	2
	GA clustering	4	21.75	3	23	2
	CLICK	3	9.12	1	4	1
	δ-Biclusters	10	−0.65	0	0	0
Yeast PHO	ACCA	9	28	4	91	9
	MIND	3	0.86	0	0	0
	K-means (Euclidian)	3	9.34	2	19	3
	K-means (Pearson)	9	10	6	37	4
	PAM	3	14.5	2	20	3
	DIANA	3	0.62	2	5	0
	FCM	2	8.46	2	14	2
	GK	10	8.94	3	21	3
	EM clustering	3	12.3	2	20	2
	SOM	52	2.6	3	12	4
	GA clustering	4	17.3	3	25	3
	CLICK	15	1.83	1	15	3
	δ-Biclusters	10	0.05	0	0	0
Yeast AFR	ACCA	4	25.9	4	63	6
	MIND	5	10.4	2	18	0
	K-means (Euclidian)	9	15.44	4	56	3
	K-means (Pearson)	9	15.42	4	51	3
	PAM	10	16.11	4	57	4
	DIANA	4	1.62	1	7	0
	FCM	7	14.41	3	55	2
	GK	7	16.82	4	61	4
	EM clustering	7	14.1	3	39	3
	SOM	46	1.36	1	8	2
	GA clustering	9	16.72	5	59	4
	CLICK	12	1.17	1	13	2
	δ-Biclusters	10	0.13	0	0	0
Yeast AFRt	ACCA	7	27.9	5	76	7
	MIND	5	15.7	3	46	2
	K-means (Euclidian)	9	16.01	3	58	4
	K-means (Pearson)	7	16.2	4	64	4
	PAM	9	16.12	4	61	4
	DIANA	9	0.27	0	0	0
	FCM	2	15.71	2	56	3
	GK	7	16.67	5	71	3
	EM clustering	7	15.5	4	41	3
	SOM	49	4.21	2	11	3
	GA clustering	9	16.59	6	55	4
	CLICK	12	1.47	1	10	2
	δ-Biclusters	10	−0.32	0	0	0
Yeast Cho et al.	ACCA	9	41.1	5	121	12
	MIND	6	39.2	3	115	6
	K-means (Euclidian)	4	39.68	3	106	6
	K-means (Pearson)	9	37.8	4	97	7
	PAM	4	40.03	3	110	8
	DIANA	3	18.56	2	42	3
	FCM	2	34.56	2	90	5
	GK	9	39.06	5	103	5
	EM clustering	3	38.62	2	79	4
	SOM	57	26.96	4	55	3
	GA clustering	9	40.54	5	116	7
	CLICK	16	15.23	2	28	4
	δ-Biclusters	10	4.08	1	16	3

category is said to be "enriched" if the corresponding P-value is less than a predefined threshold value. A low P-value indicates that the genes belonging to the enriched functional categories are biologically significant in the corresponding clusters. In this paper, only functional categories with P-value $<5.0 \times 10^{-7}$ are reported in order to restrict the size of the paper.

The enriched functional categories for each cluster obtained by the ACCA (only for $K \in \{2, 3, \ldots, 10\}$, which produces the best solution in terms of z-score) on nine datasets are listed in Tables 6–29 in Supplementary material. The functional enrichment of each GO category in each of the clusters was calculated by its P-value. To compute the P-value, we employed the software

Table 3
Various comparative scores on different clustering algorithms for mammalian datasets.

Dataset	Method	Total clusters (K)	z-Score	Enriched clusters	Enriched attributes
GDS958 wild-type	ACCA	10	11.7	5	36
	MIND	5	1.56	2	6
	K-means (Euclidian)	9	8.32	4	29
	K-means (Pearson)	10	8.2	4	32
	PAM	10	10.26	4	33
	DIANA	10	1.9	0	0
	FCM	10	9.08	4	32
	GK	10	10.63	5	34
	EM clustering	3	9.61	2	24
	SOM	32	1.12	1	3
	GA clustering	9	9.91	4	33
	CLICK	3	0.73	1	4
	δ-Biclusters	10	0.04	0	0
GDS958 knocked out	ACCA	10	9.6	4	32
	MIND	4	1.39	0	0
	K-means (Euclidian)	10	7.18	3	23
	K-means (Pearson)	10	7.2	3	27
	PAM	10	8.11	3	29
	DIANA	10	0.12	0	0
	FCM	10	6.66	4	23
	GK	10	8.57	4	30
	EM clustering	4	6.97	2	17
	SOM	31	0.85	0	0
	GA clustering	10	8.42	3	30
	CLICK	3	0.79	1	5
	δ-Biclusters	10	−1.34	0	0
GDS1423	ACCA	7	31.2	7	959
	MIND	7	12.4	3	126
	K-means (Euclidian)	8	29.44	7	807
	K-means (Pearson)	7	30	6	812
	PAM	7	30.85	7	842
	DIANA	7	2.1	2	66
	FCM	9	25.74	7	701
	GK	9	27.34	7	774
	EM clustering	5	24.34	3	268
	SOM	26	3.41	2	102
	GA clustering	7	30.35	7	816
	CLICK	2	0.17	0	0
	δ-Biclusters	10	4.36	3	109
GDS2745	ACCA	6	23	6	151
	MIND	4	3.4	2	67
	K-means (Euclidian)	5	20.53	4	136
	K-means (Pearson)	5	21.4	5	140
	PAM	5	21.67	4	141
	DIANA	6	1.06	2	40
	FCM	8	19.17	6	129
	GK	7	21.83	7	143
	EM clustering	5	18.76	4	116
	SOM	42	2.13	2	43
	GA clustering	5	22.55	4	144
	CLICK	2	0.27	0	0
	δ-Biclusters	10	2.8	1	30

Funcassociate (http://llama.med.harvard.edu/cgi/func/funcassociate) [5]. Tables 2–4 show the total number of functionally enriched attributes found in all the clusters in a clustering result. Higher the number of functionally enriched attributes better is the result. Similarly, enriched clusters column in Tables 2–4 show how many clusters of a clustering result have at least one functionally enriched attribute. Higher the number better is the result.

Enrichment of GO categories in clusters obtained by ACCA for the Yeast ATP dataset is listed in Table 6 in Supplementary material. Similarly, for Yeast PHO, Yeast AFR, Yeast AFRt and Yeast Cho et al. datasets, enriched categories are listed in Tables 7–12 in Supplementary material, respectively. For GDS958 wild-type dataset, the corresponding information is listed in Table 13 in Supplementary material and that of Knockedout dataset in Table 14 in Supplementary material. Similarly, for GDS1423 and GDS2745, the corresponding tables are Tables 15–29 in Supplementary material.

Analysis: Of the five clusters obtained for the Yeast ATP dataset (Table 6), the cluster C_2 contains several enriched categories on 'cytosolic ribosome'. The highly enriched category in cluster C_2 is the 'cytosolic ribosome (sensu Eukaryota)/80S ribosome' with P-value of 5.7×10^{-12}. In the case of the Yeast PHO dataset (Tables 7 and 8), the cluster C_8 contains several enriched categories on 'biogenesis'. The highly enriched categories in cluster C_8 are the 'ribosome biogenesis' with P-value of 9.4×10^{-56}, the 'cytoplasm organization and biogenesis' and the 'ribosome biogenesis and assembly' with P-value of 6.00×10^{-55} each. The cluster C_9 contains several enriched categories on 'ribosome'; more specifically, 'structural constituent of ribosome/ribosomal protein' with P-value of 5.6×10^{-39} as the highly enriched category. The GO category 'cytosolic ribosome (sensu Eukaryota)/80S ribosome' is also highly enriched in this cluster with P-value of 3.2×10^{-37}. For the Yeast AFR dataset (Table 9), the cluster C_1 contains several enriched

Table 4
Various comparative scores on ACCA and DCCA.

Dataset	Genes/conditions	Method	Total clusters (K)	z-Score	Enriched clusters	Enriched attributes	Enriched transcription factors
Yeast ATP	6215/3	ACCA	5	22.9	5	28	5
		DCCA	5	21.9	3	28	4
Yeast PHO	6013/8	ACCA	52	30	23	129	11
		DCCA	52	29.8	8	113	9
Yeast AFR	6184/8	ACCA	67	29.3	21	92	6
		DCCA	67	26.2	10	89	6
Yeast AFRt	6190/7	ACCA	41	32.7	11	122	7
		DCCA	41	31.4	7	107	6
Yeast Cho et al.	6457/17	ACCA	138	55.6	36	188	19
		DCCA	138	49.4	18	187	16
GDS958 wild-type	22,690/6	ACCA	39	19.6	19	55	
		DCCA	39	18.7	16	50	
GDS958 knocked out	22,690/6	ACCA	40	18.4	14	60	
		DCCA	40	17.9	11	57	
GDS1423	22,283/4	ACCA	14	39.7	14	1014	
		DCCA	14	37.1	14	1000	
GDS2745	22,645/6	ACCA	43	30.9	39	246	
		DCCA	43	30.7	32	202	

categories on 'ribosome'. The highly enriched categories in the cluster C_1 is the 'cytosolic ribosome (sensu Eukaryota)/80S ribosome' with P-value of 5.8×10^{-29}. The cluster C_4 contains several enriched categories on 'biogenesis'. The 'ribosome biogenesis' with P-value of 1.7×10^{-25} is the most enriched category in the cluster C_4.

As in the above datasets, for the Yeast AFRt dataset (Table 10), the cluster C_2 contains several enriched categories on 'biogenesis'. The highly enriched categories in cluster C_2 are the 'cytoplasm organization and biogenesis' and the 'ribosome biogenesis and assembly' with P-value of 1.7×10^{-51} each.

In the case of the Yeast Cho et al. dataset (Tables 11 and 12), the cluster C_1 contains several enriched categories on 'biogenesis'. The highly enriched categories in cluster C_1 are the 'ribosome biogenesis' with P-value of 9.3×10^{-62}, the 'cytoplasm organization and biogenesis' and the 'ribosome biogenesis and assembly' with P-value of 2.6×10^{-58} each. The cluster C_2 contains several enriched categories on 'ribosome'. The highly enriched category in cluster C_2 is the 'structural constituent of ribosome/ribosomal protein' with P-value of 4.3×10^{-79}. Two other highly enriched categories in cluster C_2 are the 'ribosome' with P-value of 1.7×10^{-78} and the 'cytosolic ribosome (sensu Eukaryota)/80S ribosome' with P-value of 1.3×10^{-75}.

The categories 'ribosome' (in C_2 for Yeast ATP, C_9 for Yeast PHO, C_1 for Yeast AFR, C_2 for Yeast AFRt and in C_2 for Yeast Cho et al. datasets) and 'biogenesis' (in C_2 for Yeast ATP, C_8 for Yeast PHO, C_4 for Yeast AFR, C_2 for Yeast AFRt and in C_1 for Yeast Cho et al. datasets) are enriched in at least one of the clusters for all the yeast datasets. This similarity in results from different yeast datasets shows consistency of ACCA.

In the case of the GDS958 wild-type dataset (Table 13), the highly enriched category in cluster C_1 is the 'motor activity' with P-value of 1.9×10^{-15}. The highly enriched categories in cluster C_8 are the 'MHC class II receptor activity' with P-value of 3.4×10^{-14} and the 'hydrolase activity' with P-value of 4.2×10^{-14}. In the case of the GDS958 IL-13 Knockedout dataset (Table 14), the highly enriched categories in cluster C_3 are the 'RNA binding' with P-value of 4×10^{-14} and the 'DNA binding' with P-value of 7.7×10^{-14}.

For GDS1423 (Tables 15–27) and GDS2745 datasets (Tables 28 and 29), all clusters are found enriched. The highly enriched category, for GDS1423, in cluster C_6 is the 'multicellular organismal process' with P-value of 1.2×10^{-82}. The cluster C_1 obtained from the GDS2745 dataset (Tables 28 and 29) contains several enriched categories on 'intracellular organelle'. The highly enriched category in cluster C_3 is the 'intracellular membrane-bound organelle' with P-value of 1.3×10^{-34}.

From the results of Tables 6–29 in Supplementary material, we see that the clusters obtained by ACCA are highly enriched in functional categories.

Comparisons: Here we compare the ability of detecting functionally enriched clusters/categories by the aforesaid clustering algorithms. Tables 2 and 3 show that five out of five clusters produced by ACCA of Yeast ATP dataset, contain functionally enriched categories. Similarly, for GK (Tables 2 and 3), five out of nine clusters of Yeast ATP dataset are functionally enriched, and total number of enriched categories for clusters generated by ACCA (28) is greater than that generated by GK (25). For K-means (Euclidian), PAM and Fuzzy c-means (in Tables 2 and 3), only three out of four clusters contain functionally enriched categories while for MIND two out of three clusters contain functionally enriched categories. DIANA, and δ-biclusters could not find any enriched functional category. For K-means (Pearson), EM clustering, SOM, GA clustering and CLICK (in Tables 2 and 3) functionally enriched categories are vary few compare to ACCA. This result, for Yeast ATP dataset, clearly shows ACCA produces better clustering solution than the other clustering algorithms considered in our analysis. Similar investigations were carried out for the other datasets using the aforesaid algorithms. In all the cases, the numbers of enriched attributes corresponding to ACCA are the highest among those of the other algorithms, and in most of the cases, ACCA provides higher number of enriched clusters (Tables 2 and 3).

Regarding the comparative analysis of ACCA and DCCA in identifying enriched attributes, we have set K to the number of clusters obtained by the DCCA. Then ACCA was run using these K-values on these nine datasets. It has been found that the numbers of enriched clusters and attributes obtained by ACCA are higher than that of DCCA for almost all the cases (Table 4).

3.2.3. Analysis of transcription factor binding sites

As in Section 3.2.2, we can determine P-values corresponding to the fact that a given number of genes in a cluster include a transcription factor. Tables 2 and 4 show the total number of transcription factors found enriched in all the clusters in a clustering result, and here again a higher number corresponds to a better result.

We have considered PRIMA available in EXPANDER [43] for analysis of transcription factor binding sites corresponding to the clusters of all considered algorithms for yeast datasets. Number of enriched transcription factors for each cluster of yeast datasets is found based on P-values. Tables 2 and 4 show that ACCA has resulted in the highest number of significant transcription factors compared to that obtained by the other algorithms. Higher the number of significant transcription factors, better is the algorithm.

In this paper, only transcription factors with P-value $<1.0 \times 10^{-4}$ are reported as significance.

3.2.4. Performance comparison by visualizing expression profile plots

The superior capability of ACCA over other clustering algorithms considered here, in grouping genes with similar pattern of variation in their expression values, can also be visualized from the expression profile plots. Figs. 5–7 in Supplementary material are such plots corresponding to ACCA, DCCA and K-means (Euclidian) for the dataset GDS2745. Figs. 12–22 in Supplementary material corresponding to the other algorithms. From these figures, it is evident that ACCA is able to capture the varying pattern in expression profiles far better than the other clustering algorithms. It is interesting to note that although both ACCA, DCCA and MIND are based on Pearson's correlation coefficient, ACCA performs the best over the others as depicted by z-scores (Tables 2–4), P-value (Tables 2–4) and expression profile plots (Figs. 5, 6 and 12 in Supplementary material). Regarding comparison between ACCA and DCCA, expression profile plots of ACCA (Fig. 5 in Supplementary material) and DCCA (Fig. 6 in Supplementary material) depict comparable results but as depicted by z-scores (Table 4) and P-value (Table 4) ACCA is able to produce better clustering results than DCCA.

3.3. Some important characteristics of ACCA

Here we provide some important characteristics of ACCA based on the results obtained for nine gene expression datasets.

Comparisons with Spearman's rank correlation coefficient and Euclidean distance: ACCA is a general algorithm and any pair wise correlation measure can be used as a similarity measure instead of Pearson's correlation coefficient. We have compared the results using Spearman's rank correlation coefficient and Euclidean distance. For example, the z-scores for Yeast ATP dataset, with Spearman's rank correlation and Euclidean distance are 20.4 and 18.6, while that using Pearson's correlation is 22.9. Similarly the numbers of functionally enriched attributes in the clusters, for Yeast ATP dataset, with Spearman's rank correlation and Euclidean distance have been found to be 25 and 17, respectively, while that using Pearson's correlation is 28. Moreover, the numbers of significant transcription factor, for Yeast ATP dataset have been found to be 3 both with Spearman's rank correlation and Euclidean distance, while that using Pearson's correlation is 5. Thus Pearson's correlation coefficient results in the best for Yeast ATP dataset.

Sensitivity to initial clustering: Sensitivity of ACCA to initial clustering has been tested by running ACCA multiple times for the same K value, and have plotted the resulting z-score, total number of functionally enriched attributes and total number of functionally enriched transcription factors for each run of ACCA. Plots for Yeast ATP dataset is shown in Fig. 23 in Supplementary material. From Fig. 23, we observe that all these parameter-values are closed for all the run of ACCA. The same things happen for all the other datasets too.

Main difference of ACCA over the other algorithms: The clustering algorithms like K-means, PAM and EM decide on inclusion of samples to the clusters based on their similarity values with the central elements of the clusters. On the other hand, ACCA does this based on the correlation values with all the elements in all the clusters. Thus the chance of miss clustering by K-means, PAM and EM is higher than that by ACCA.

Some other differences between ACCA and the other algorithms including DCCA [8] are: (i) ACCA is a partitional clustering algorithm whereas DCCA is a hierarchical clustering algorithm. (ii) In cluster updation step (Step 2.iv), ACCA computes the similarity of each gene with other genes already assigned to all the new clusters formed during the current iteration. On the other hand, for algorithms including DCCA, K-means, PAM, this similarity is determined based on the clustering result obtained in the preceding iteration. That is, similarity computation during updation step in ACCA is based on the modified clusters that are about to be formed in the current iteration, as opposed to that in DCCA and others, where this computation uses the clusters obtained in the preceding iteration. (iii) ACCA terminates its execution when there is no change in the sets of clusters in two successive iterations. For DCCA, lack of a single negative correlation in any cluster indicates its termination.

Time complexity: Upper bound of the execution time of ACCA is $O(n^2)$ for a dataset of n genes and m samples. For example, ACCA takes about 30 s to generate five clusters for Yeast ATP dataset in a server with 2 GHz Quad core processor and 2GB RAM. Upper bound of the execution time of the algorithm DCCA [8] is $O(n^3)$. Compared to ACCA, some of the other existing clustering algorithms take similar time. For example, upper bound of the execution time of the algorithm of DIANA [25,13] is $O(n^3)$, while that for PAM [25], GA clustering [37], and CLICK [42] are $O(n^2)$. For the K-means [30,25,44], EM clustering [19], SOM [40], Fuzzy c-means (FCM) [17,6,7] and GK [24] upper bound is $O(n)$. Upper bound for a single iteration of δ-biclusters [11], is $O(nm)$.

4. Conclusions

We have presented here a novel clustering algorithm, called average correlation clustering algorithm (ACCA), which is able to obtain clustering solution from gene expression dataset with very high biological significance. ACCA places genes in a cluster, having more common transcription factors. The expression values of these genes change in a similar way. Conventional clustering algorithms are used to place genes with similar expression level in the same clusters without monitoring similarity or dissimilarity in expression pattern that changes over samples. Co-regulated genes are generally expected to follow the same expression pattern, i.e., the same type of changes in expression values over samples while they may not have the same level of expression. For this reason, ACCA may be able to place co-regulated genes in the same clusters much more efficiently than the other conventional algorithms.

Moreover, the conventional clustering algorithms use single or multiple cluster representative points, and membership of a gene to a cluster depends only on these representative points of the cluster. Selection of cluster representative points may have effect on clustering results. On the other hand, in ACCA, a cluster is represented by all its members and membership of a gene in the cluster depends on all the members of the cluster. This makes ACCA more accurate compared to the conventional clustering algorithms. Like some other algorithms, ACCA also belongs to the category of partitional clustering algorithms. ACCA, like most of the clustering algorithms, cannot guarantee global optimum. However, ACCA computes membership values of elements (genes) based on their correlation values with every element in the clusters. Thus the results obtained by ACCA may be closer to global optimum compared to those obtained by K-means, EM or PAM as they do the same task based only on central elements of the clusters.

Analysis of the results show that clustering solution obtained by ACCA is much more biologically significant than that obtained by some other algorithms, viz., MIND, K-means (Euclidian), PAM, DIANA, Fuzzy c-means, K-means (Pearson), EM clustering, GA clustering, CLICK, SOM, δ-biclusters, and GK. The results of ACCA is more or less independent of the initial clustering. Although ACCA, unlike DCCA, requires the expected number of clusters as an input, it is able to provide better clustering solution than DCCA and all other algorithms considered in this paper. Moreover, execution time of the algorithm ACCA ($O(n^2)$) is considerably low compare to DCCA

($O(n^3)$). DCCA is able to obtain clustering solution if and only if at least one *negative* edge is present. If the input dataset contains a set of data points such that all the pairs are positively correlated, i.e., have *positive* edges among them, DCCA will fail to obtain clustering results although ACCA will work efficiently. However, ACCA will not work if a dataset contains less than three samples. In this case, calculated correlation value will be either +1 or −1.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.jbi.2010.02.001.

References

[1] Alon N, Makarychev K, Makarychev Y, Naor A. Quadratic forms on graphs. In: Proceedings of the 37th STOC; 2005. p. 634–43.
[2] Arai K, Barakbah AR. Hierarchical k-means: an algorithm for centroids initialization for k-means. Rep Fac Sci Eng Saga Univ 2007;36:25–31.
[3] Ashburner M, Ball C, Blake J, Botstein D, Butler H, Cherry J, et al. Gene ontology: tool for the unification of biology. Nat Genet 2000;25(1):25–9.
[4] Bansal N, Blum A, Chawla S. Correlation clustering. Mach Learn 2004;56: 89–113.
[5] Berriz FG, King OD, Bryant B, Sander C, Roth FP. Characterizing gene sets with funcassociate. Bioinformatics 2003;19(18):2502–4.
[6] Bezdek JC. Pattern recognition with fuzzy objective function algorithms. New York: Plenum Press; 1981.
[7] Bezdek JC, Ehrlich R, Full W. FCM: the fuzzy c-means clustering algorithm. Comput Geosci 1984;10:191–203.
[8] Bhattacharya A, De RK. Divisive correlation clustering algorithm (DCCA) for grouping of genes: detecting varying patterns in expression profiles. Bioinformatics 2008;24(11):1359–66.
[9] Charikar M, Guruswami V, Wirth A. Clustering with qualitative information. In: Proceedings of the 44th FOCS; 2003. p. 524–33.
[10] Charikar M, Wirth A. Maximizing quadratic programs: extending Grothendieck's inequality. In: Proceedings of the 45th FOCS; 2004. p. 524–33.
[11] Cheng Y, Church GM. Biclustering of expression data. Proc Int Conf Intell Syst Mol Biol 2000;8:93–103.
[12] Cohen W, Richman J. Learning to match and cluster large high-dimensional data sets for data integration. In: Proceedings of the eighth ACM SIGKDD; 2002. p. 475–80.
[13] Datta S, Datta S. Evaluation of clustering algorithms for gene expression data. BMC Bioinform 2006;7:s17.
[14] Demaine ED, Emanuel D, Fiat A, Immorlica N. Correlation clustering in general weighted graphs. Theor Comput Sci 2006;361:172–87.
[15] Demaine ED, Immorlica N. Correlation clustering with partial information. In: Proceedings of the RANDOM-APPROX; 2003. p. 1–13.
[16] Dembele D, Kastner P. Fuzzy c-means method for clustering microarray data. Bioinformatics 2003;19:973–80.
[17] Dunn JC. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. J Cybern 1973;3:32–57.
[18] Everitt BS, Landau S, Leese M. Cluster analysis. London: Hodder Arnold; 2001.
[19] Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. J Am Stat Assoc 2002;97:611–31.
[20] Getz G, Levine E, Domany E. Coupled two-way clustering analysis of gene microarray data. Proc Natl Acad Sci USA 2000:12079–84.
[21] Gibbons F, Roth F. Judging the quality of gene expression-based clustering methods using gene annotation. Genome Res 2002;12(10):1574–81.
[22] Gordon AD. Classification. Boca Raton (FL): CRC Press; 1999.
[23] Gun AM, Gupta MK, Dasgupta B. Fundamentals of statistics. Kolkata: The World Press Private Limited; 2005.
[24] Gustafson EE, Kessel WC. Fuzzy clustering with a fuzzy covariance matrix. Proc IEEE Conf Decision Control 1979:761–6.
[25] Han J, Kamber M. Data mining: concepts and techniques. Los Altos (CA): Morgan Kaufman; 2001.
[26] Hartigan JA. Direct clustering of a data matrix. J Am Stat Assoc 1972;67(337):123–9.
[27] Heyer LJ, Kruglyak S, Yooseph S. Exploring expression data: Identification and analysis of coexpressed genes. Genome Res 1999;9:1106–15.
[28] Ihmels J, Friedlander G, Bergmann S, Sarig O, Ziv Y, Barkai N. Revealing modular organization in the yeast transcriptional network. Nat Genet 2002;31:370–7.
[29] Issel-Tarver L, Christie K, Dolinski K, Andrada R, Balakrishnan R, Ball C, et al. Saccharomyces genome database. Methods Enzymol 2002;350:329–46.
[30] Jain AK, Dubes RC. Algorithms for clustering data. Englewood Cliffs (NJ): Prentice-Hall; 1988.
[31] Jain NC, Indrayan A, Goel LR. Monte carlo comparison of six hierarchical clustering methods on random data. Pattern Recogn 1986;19:95–9.
[32] Kim DW, Lee KH, Lee D. Detecting clusters of different geometrical shapes in microarray gene expression data. Bioinformatics 2005;21(9):1927–34.
[33] Kluger Y, Basri R, Chang JT, Gerstein M. Spectral biclustering of microarray cancer data: co-clustering genes and conditions. Genome Res 2003;13(4): 703–16.
[34] Li K, Wang L, Hao L. Comparison of cluster ensembles methods based on hierarchical clustering. Proc Int Conf Comput Intell Natl Comput 2009: 499–502.
[35] Loganantharaj R, Cheepala S, Clifford J. Metric for measuring the effectiveness of clustering of DNA microarray expression. BMC Bioinform 2006;7:s5.
[36] Lukashin AV, Fuchs R. Analysis of temporal gene expression profiles: clustering by simulated annealing and determining the optimal number of clusters. Bioinformatics 2001;17:405–14.
[37] Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique. Pattern Recogn 2000;33:1455–65.
[38] Press W, Flannery B, Teukolsky S, Vetterling W. Numerical recipes – the art of scientific computing. Cambridge: Cambridge University Press; 2003.
[39] Qin J, Lewis DP, Noble WS. Kernel hierarchical gene clustering from microarray expression data. Bioinformatics 2003;19:2097–104.
[40] Reich M, Ohm K, Angelo M, Tamayo P, Mesirov JP. Genecluster 2.0: an advanced toolset for bioarray analysis. Bioinformatics 2004;20:1797–8.
[41] Shao J, Tanner SW, Thompson N, Cheatham TE. Clustering molecular dynamics trajectories: 1. Characterizing the performance of different clustering algorithms. J Chem Theory Comput 2007;2:2312–34.
[42] Sharan R, Maron-Katz A, Shamir R. Click and expander: a system for clustering and visualizing gene expression data. Bioinformatics 2003;19:1787–99.
[43] Tanay A, Sharan R, Shamir R. Discovering statistically significant biclusters in gene expression data. Bioinformatics 2002;18:S136–44.
[44] Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. Systematic determination of genetic network architecture. Nat Genet 1999;22:281–5.
[45] Teyra J, Paszkowski-Rogacz M, Anders G, Pisabarro MT. Scowlp classification: Structural comparison and analysis of protein binding regions. BMC Bioinform 2008;9:9.
[46] Wang H, Wang W, Yang J, Yu PS. Clustering by pattern similarity in large data sets. In: Proceedings of the ACM SIGMOD; 2002. p. 394–405.
[47] Xu Y, Olman V, Xu D. Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. Bioinformatics 2002;18(4):536–45.
[48] Yang J, Wang W, Wang H, Yu P. δ-Clusters: capturing subspace correlation in a large data set. In: Proceedings of the 18th IEEE international conference on data engineering; 2002. p. 517–28.
[49] Yang J, Wang W, Wang H, Yu P. Enhanced biclustering on expression data. In: Proceedings of the third IEEE conference on bioinformatics and bioengineering; 2003. p. 321–27.
[50] Zhang Y, Luxon BA, Casola A, Garofalo RP, Jamaluddin M, Brasier AR. Expression of respiratory syncytial virus-induced chemokine gene networks in lower airway epithelial cells revealed by cdna microarrays. J Virol 2001;75(19):9044–58.