BIOINFORMATICS

Vol. 24 no. 11 2008, pages 1359–1366

Divisive Correlation Clustering Algorithm (DCCA) for grouping of genes: detecting varying patterns in expression profiles

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

[1]Department of Computer Science and Engineering, Netaji Subhash Engineering College, Garia and
[2]Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India

ABSTRACT

Motivation: Cluster analysis (of gene-expression data) is a useful tool for identifying biologically relevant groups of genes that show similar expression patterns under multiple experimental conditions. Various methods have been proposed for clustering gene-expression data. However most of these algorithms have several shortcomings for gene-expression data clustering. In the present article, we focus on several shortcomings of conventional clustering algorithms and propose a new one that is able to produce better clustering solution than that produced by some others.

Results: We present the Divisive Correlation Clustering Algorithm (DCCA) that is suitable for finding a group of genes having similar pattern of variation in their expression values. To detect clusters with high correlation and biological significance, we use the correlation clustering concept introduced by Bansal *et al*. Our proposed algorithm DCCA produces a clustering solution without taking number of clusters to be created as an input. DCCA uses the correlation matrix in such a way that all genes in a cluster have highest average correlation with genes in that cluster. To test the performance of the DCCA, we have applied DCCA and some well-known conventional methods to an artificial dataset, and nine gene-expression datasets, and compared the performance of the algorithms. The clustering results of the DCCA are found to be more significantly relevant to the biological annotations than those of the other methods. All these facts show the superiority of the DCCA over some others for the clustering of gene-expression data.

Availability: 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 one of the most important tasks that deals with finding a structure in a collection of unlabeled data. A loose definition of clustering could be 'the process of organizing objects into groups whose members are similar in some way' (Han and Kamber, 2001). A cluster is therefore a collection of objects that are *similar* among themselves and *dissimilar* to the objects belonging to other clusters. Clustering techniques use various distance measures for determining *similarity/dissimilarity* between a pair of objects and decide whether they belong to the same or different clusters. Euclidean and Maholanobis distances are commonly used distance measures in this regard.

Generally, clustering algorithms could be either hierarchical or partitional. Some of the problems with conventional hierarchical and partitional clustering methods are: (i) These algorithms find clusters containing co-expressed genes. They cannot determine a group of genes having similar pattern of variations in the expression profiles. In other words, the clusters obtained by these algorithms contain genes with similar expression values. (ii) They need the number of clusters we want to create, as an input. Although DB index, Dunn index (Han and Kamber, 2001; Jain and Dubes, 1988; Mitra and Acharya, 2003) or any other cluster validity indices could be used for determination of an optimal number of clusters for a given dataset, this number may be found to be different for different cluster validity indices. (iii) Conventional hierarchical and partitional clustering algorithms use either Euclidean or Maholanobis distances as distance measures. The Euclidean norm-based methods find mainly spherical shape of clusters (Kim *et al*., 2005) while methods based on the Maholanobis distance detecting mainly ellipsoidal ones (Kim *et al*., 2005), even if these shapes of clusters may not be present in a dataset. (iv) For large dataset, these algorithms may result in large miss clustering. Moreover hierarchical clustering algorithms like AGNES or DIANA (Han and Kamber, 2001; Jain and Dubes, 1988; Mitra and Acharya, 2003), may result in one single large cluster and several singletons.

In order to overcome some of the drawbacks of conventional clustering algorithms, new clustering algorithms have been developed for gene-expression data analysis. Xu *et al*. (2002) have proposed a new framework for representing a set of multi-dimensional gene-expression data as a minimum spanning tree (MST). Based on the MST representation, they have implemented a number of efficient clustering algorithms, including two with guaranteed global optimality. Kim *et al*. (2005) have used Gustafson–Kesel (GK) (Gustafson and Kessel, 1979) clustering method for microarray gene-expression data for

detecting clusters of different shapes in a dataset. Sharan *et al.* (2003), have presented a new clustering algorithm, called CLICK. CLICK uses graph-theoretic and statistical techniques to identify tight groups (kernels) of highly similar genes that are likely to belong to the same true cluster. Several heuristic procedures are then used to expand the kernels into the full clusters. Qin *et al.* (2003) have described a generalization of the hierarchical clustering algorithm named as kernel hierarchical clustering algorithm, and then evaluated the utility of the kernel hierarchical clustering algorithm using both internal and external validation. Lukashin and Fuchs (2001) have proposed a new clustering algorithm for clustering of gene-expression data based on the method of simulated annealing. Dembele and Kastner (2003) have developed a method for selection of parameters for Fuzzy c-means (FCM) algorithm when it is applied to gene-expression data clustering.

There also exist several biclustering algorithms in this regard. These include greedy biclustering algorithms of Cheng and Church (2000) and Ben-Dor *et al.* (2002), iterative algorithms of Getz *et al.* (2000) and Ihmels *et al.* (2004), SAMBA of Tanay *et al.* (2002), Flexible Overlapped biClustering (FLOC) Kluger *et al.* (2003), a graph theoretic algorithm of Alexe *et al.* (2002). Murali and Kasif (2003) have defined biclusters as conserved gene expression motif i.e. xMOTIF, and devised an algorithm to find largest xMOTIF. Prelic *et al.* (2006) have compared performance of different biclustering algorithms, and proposed a fast divide-and-conquer biclustering algorithm (Bimax).

Apart from different approaches of clustering and biclustering, Kim and Tidor (2003) have applied the notion of non-negative matrix factorization (NMF) to the analysis of gene-array experiments. NMF is capable of recognizing similarity between sub-portions of the data corresponding to localized features in expression space. It is to be mentioned here that the method is not suitable for low-dimensional data.

Correlation clustering is a new clustering method introduced by Bansal *et al.* (2004), which is basically based on the notion of graph partitioning. Here the quality of clusters is measured in terms of certain parameters, namely the number of *agreements* and the number of *disagreements*. First of all, a graph is constructed from input data by considering genes as nodes and correlation between the genes as edges. There are two types of edges, namely *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 *negative* edges between genes in different clusters. The number of *positive* edges between genes indicates 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 (Alon *et al.*, 2005; Charikar *et al.*, 2003; Charikar and Wirth, 2004; Demaine and Immorlica, 2003; Demaine *et al.*, 2006) 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 can be obtained by simply deleting all *negative* edges and output the connected components of the remaining graph (Cohen and Richman, 2002). It has been proved that if no perfect clustering exists, no algorithm, based on correlation coefficient can find an optimal clustering in polynomial time (Bansal *et al.*, 2004). There are two different approaches (Bansal *et al.*, 2004) for correlation clustering. Both these approaches create K number of clusters without taking K as an input. One approach is based on minimization of *disagreement* while the other is based on maximization of *agreement*.

Bansal *et al.*, 2004 has proved that the problem of minimizing *disagreement* or equivalently maximizing *agreement* is NP-complete. They have provided a constant factor approximation algorithm to the problem of minimizing *disagreements*, and a polynomial-time approximation scheme (PTAS) for maximizing *agreements* (Bansal *et al.*, 2004). Both these algorithms are based on graph partitioning. Main problem of these two algorithms is that they can only work on a given unweighted complete graph with *positive/negative* labels on the edges. Another major problem with Bansal *et al.*'s (2004) correlation clustering algorithm is that they have considered only sign of the correlation coefficient but not the magnitude. This may deteriorate the quality of clusters in terms of biological relevance.

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 article, a new clustering algorithm, called *divisive correlation clustering algorithm* (DCCA). This is a hierarchical clustering method but differs from the concept of conventional hierarchical algorithms. Unlike hierarchical clustering method, DCCA produces clusters with nearly uniform size based on input patterns and can repair that defects occur in a clustering step to produce proper clustering solution. DCCA uses Pearson correlation coefficient as the similarity measure. The common advantage of DCCA over conventional hierarchical and partitional clustering methods is that it can produce K clusters from an input dataset without taking K as an input. DCCA uses concepts of correlation clustering but the algorithm differs from that in (Bansal *et al.*, 2004).

DCCA considers the value of Pearson correlation coefficients among all pairs of genes. All the pairs of genes with negative correlation coefficient values between them should be in different clusters. In each iteration, the algorithm selects a cluster having a pair of gene $(\mathbf{x}_i, \mathbf{x}_j)$ with the most negative correlation coefficient between them. Then this selected cluster is partitioned into two disjoint clusters. Partitioning is done in such a way that the genes $\mathbf{x}_i$ and $\mathbf{x}_j$ placed in different clusters. The data points (genes) having larger correlation coefficient, with the gene $\mathbf{x}_i$ compared to that of $\mathbf{x}_j$ are placed in the cluster that contains $\mathbf{x}_i$. The other data points (genes) are placed in the cluster that contains the gene $\mathbf{x}_j$. The placement of each gene is checked whether they are placed in the most appropriate cluster or not. If placement is inappropriate then it is changed over the set of clusters. Placement checking and correction steps iterate until all genes are placed in appropriate clusters. Partitioning continues until all the pairs of the genes inside the clusters are only positively correlated. DCCA generates clusters where genes in a cluster have similar pattern of variation in their

expression profiles. Unlike conventional clustering algorithms, here a cluster may contain genes with both high and low expression values.

In our study the superior capability of clustering by DCCA over a number of algorithms namely Bansal's Minimizing Disagreement (MIND) in (Bansal *et al.*, 2004), *K*-means (Han and Kamber, 2001; Jain and Dubes, 1988; Mitra and Acharya, 2003), PAM (Han and Kamber, 2001; Mitra and Acharya, 2003), DIANA (Han and Kamber, 2001), FCM (Bezdek, 1981; Bezdek *et al.*, 1984; Dunn, 1973), GK (Gustafson and Kessel, 1979) and an NMF-based algorithm in (Kim and Tidor, 2003) is demonstrated through experiments with one artificial dataset and nine gene-expression datasets.

2 DIVISIVE CORRELATION CLUSTERING ALGORITHM

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 $C_1, C_2, \ldots, C_p, \ldots, C_K$. DCCA uses Pearson correlation coefficient for measuring similarity/dissimilarity between expression patterns of two genes $\mathbf{x}_i$ and $\mathbf{x}_j$, which is defined as

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

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 and 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. $\mathrm{Corr}(\mathbf{x}_i, \mathbf{x}_j) > 0$ represents that $\mathbf{x}_i$ and $\mathbf{x}_j$ are positively correlated with the degree of correlation as its magnitude. On the other hand, $\mathrm{Corr}(\mathbf{x}_i, \mathbf{x}_j) < 0$ represents that $\mathbf{x}_i$ and $\mathbf{x}_j$ are negatively correlated with value $|\mathrm{Corr}(\mathbf{x}_i, \mathbf{x}_j)|$. 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 for two genes is similar.

As mentioned in Section 1, the problem with Euclidean and Maholanobis distances (Kim *et al.*, 2005) is that they impose a fixed geometrical structure and find clusters of that shape even if they are not present. The Euclidean norm-based methods find mainly spherical shape of clusters, whereas the Maholanobis distance-based methods find mainly ellipsoidal ones, even if those shapes of clusters may not be present in a dataset. Pearson correlation coefficient is used as a measure of similarity/dissimilarity to cluster genes with similar expression patterns; genes with opposite expression patterns are assigned to different clusters. Before describing the algorithm in details, we define the following terms and measurements used in this regard.

Attraction: For two genes $\mathbf{x}_i$ and $\mathbf{x}_j$, if $\mathrm{Corr}(\mathbf{x}_i, \mathbf{x}_j)$ is greater than zero then there is an attraction between $\mathbf{x}_i$ and $\mathbf{x}_j$.

Repulsion: For two genes $\mathbf{x}_i$ and $\mathbf{x}_j$, if $\mathrm{Corr}(\mathbf{x}_i, \mathbf{x}_j)$ is less than zero then there is a repulsion between $\mathbf{x}_i$ and $\mathbf{x}_j$.

Attraction/repulsion value: Magnitude of $\mathrm{Corr}(\mathbf{x}_i, \mathbf{x}_j)$ is the strength of attraction or repulsion.

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

$$\mathrm{AVG}C_{pi} = \frac{1}{n_p}\sum_{\substack{\mathbf{x} \in C_p \\ \mathbf{x} \neq \mathbf{x}_i}} \mathrm{Corr}(\mathbf{x}_i, \mathbf{x}) \quad (2)$$

where n_p is the number of data points in $C_p - \{\mathbf{x}_i\}$. Thus $\mathrm{AVG}C_{pi}$ indicates that the average correlation 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.

DCCA considers a pair of repulsive genes that should be in different clusters as their functional behavior is opposite. Initially all the genes are considered in a single cluster. In each iteration, algorithm selects a cluster having a pair of gene $(\mathbf{x}_i, \mathbf{x}_j)$ with the largest repulsion (i.e. with the most negative repulsion value). Then this selected cluster is partitioned into two disjoint clusters. Partitioning is done in such a way that $\mathbf{x}_i$ and $\mathbf{x}_j$ are placed in two different clusters. Data points (genes) having larger attraction with $\mathbf{x}_i$ compared to $\mathbf{x}_j$ are placed in the cluster that contains $\mathbf{x}_i$. Otherwise, they are placed in the cluster that contains $\mathbf{x}_j$. Such partitioning may cause miss placement of genes as they are placed in clusters based on only attraction value with two genes $\mathbf{x}_i$ and $\mathbf{x}_j$. At this point, average correlation values for each gene $\mathbf{x}_k$ with respect to each cluster is calculated and $\mathbf{x}_k$ is placed into cluster with which $\mathbf{x}_k$ has the highest average correlation value. Partitioning continues until there is no repulsion present inside a cluster. The algorithm stops when no repulsion exists between any pair of genes inside any cluster. DCCA ensures that a gene $\mathbf{x}_i$ belongs to the cluster C_p, iff $\mathrm{AVG}C_{pi} > \mathrm{AVG}C_{qi}$, for all $q \neq p$. The algorithm also ensures that all pairs of genes in any cluster are only positively correlated.

Algorithm

Input: 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.

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

Steps:

1. Initially, consider all the genes in one cluster. Set number of cluster $K = 1$.
2. For each iteration, do:
   i. For each cluster C_p, calculate Pearson correlation coefficient [Equation (1)] between all pairs of genes in C_p.
   ii. If no *repulsion* exists between a pair of genes inside any cluster then STOP, otherwise perform Step iii.
   iii. Identify a cluster C for which a pair of genes $\mathbf{x}_i, \mathbf{x}_j$ have the most negative *repulsion* value among all the clusters.
   iv. Replace cluster C with two clusters C_p and C_q, and increase number of clusters K by one. Place gene $\mathbf{x}_i$ to C_p and $\mathbf{x}_j$ to C_q. For all the other genes $\mathbf{x}_k$ in C, compare $Corr(\mathbf{x}_i, \mathbf{x}_k)$ and $Corr(\mathbf{x}_j, \mathbf{x}_k)$.

If $Corr(\mathbf{x}_i, \mathbf{x}_k) > Corr(\mathbf{x}_j, \mathbf{x}_k)$ then place $\mathbf{x}_k$ to C_p, otherwise place $\mathbf{x}_k$ to C_q.

v. For each $\mathbf{x}_k$ in X, do:

   a. For each cluster C_p, $1 \le p \le K$, calculate average correlation value $AVGC_{pk}$ [Equation (2)].

   b. If $AVGC_{pk} > AVGC_{qk}$, for each q, $1 \le q \le K$, and $p \ne q$ then place a copy of $\mathbf{x}_k$ to new pth cluster $CNEW_p$.

vi. If $\bigcup_{p=1}^{K}(CNEW_p - C_p) = \phi$ then no change occurs in the clusters obtained in the previous iteration of Step v, i.e. $CNEW_1 = C_1, CNEW_2 = C_2, \ldots, CNEW_K = C_K$, then go to Step 2. Otherwise for each p, $1 \le p \le K$, set $C_p = CNEW_p$. Set $CNEW_p = \phi$, for each p. Then go to step v.

DCCA ensures that increase and decrease in expression values of all genes in a cluster across samples occur in the similar way. This form of resulting clusters also helps us in identifying group of genes that changes their behavior in a similar way from normal samples to diseased samples. If we consider a dataset containing normal and diseased samples then applying DCCA over the dataset will produce a set of clusters, where each cluster contains co-expressed genes both in normal and diseased condition. If gene $\mathbf{x}_k$ and $\mathbf{x}_j$ belong to the same cluster and $\mathbf{x}_k$ is over expressed in diseased samples then $\mathbf{x}_j$ is also over expressed in disease samples, and vice versa. We can identify clusters containing over/under expressed genes in diseased condition, and thus we will be able to identify group of genes potentially responsible for a particular disease. However, the issue of selecting genes potentially responsible for a particular disease, has not been considered here. Conventional clustering algorithms cannot guarantee absence of repulsion inside a cluster or highest average attraction between genes inside clusters. Due to this reasons, the DCCA is able to cluster genes with similar behavior together, with higher degree of accuracy than other conventional clustering algorithms. DCCA has another advantage over conventional clustering algorithms that it can create K number of clusters based on input data only, without taking K as an input.

2.1 Comparative analysis of the performance of DCCA over some existing algorithms using a synthetic dataset

Before going into the detailed discussion of the results on real life gene-expression data, here we demonstrate superior performance of DCCA over some existing algorithms using an artificial dataset ADS (Fig. 1). ADS contains 115 3-D samples distributed in three clusters. The values of these samples in three clusters vary mostly in x, y and z directions, respectively. Figure 2 shows results for DCCA, PAM and GK. It is clear from Figure 2 that DCCA, PAM and GK were able to obtain these three clusters successfully. On the other hand, MIND (Supplementary Fig. 3), K-means (Supplementary Fig. 4), FCM (Supplementary Fig. 5) and DIANA (Supplementary Fig. 6) were unable to obtain desired clusters for the ADS dataset.

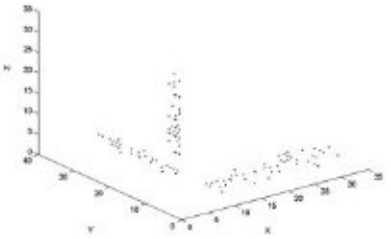

Fig. 1. Dataset ADS.

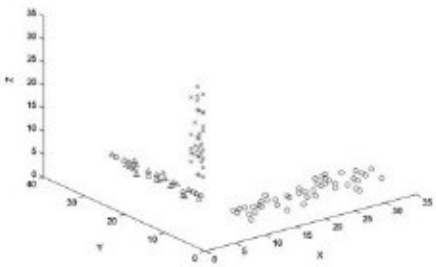

Fig. 2. Clustered output of DCCA, PAM and GK.

3 RESULTS

The effectiveness of the DCCA is demonstrated on nine gene expression datasets. These datasets deal with five yeast (http://yfgdb.princeton.edu/download/yeast_datasets/) and four mammalian datasets (http://www.ncbi.nlm.gov/projects/geo/gds). Clustering results produced by DCCA for five yeast datasets are shown in Supplementary Figures 7–11 that are generated using TreeView software (http://rana.lbl.gov/EisenSoftware.htm). The superior performance of DCCA over other clustering algorithms namely MIND in Bansal *et al.* (2004), K-means (Han and Kamber, 2001; Jain and Dubes, 1988; Mitra and Acharya, 2003), PAM (Han and Kamber, 2001; Mitra and Acharya, 2003), DIANA (Han and Kamber, 2001), FCM (Bezdek, 1981; Bezdek *et al.*, 1984; Dunn, 1973), GK (Gustafson and Kessel, 1979) algorithms and an NMF-based algorithm (Kim and Tidor, 2003) is also observed using several indices (described in the Supplementary Material). The datasets used for comparative analysis are also described in the Supplementary Material.

Figure 7 in Supplementary Material shows five clearly distinct clusters produced by DCCA for Yeast ATP. For Yeast PHO, Figure 8 in Supplementary Material shows 52 different clusters produced by DCCA. Similarly, for Yeast AFR, Yeast AFRt, Yeast data obtained by Cho *et al.*, the numbers of clusters are (Supplementary Figs. 9–11) 67, 41 and 138, respectively. The number of clusters obtained by DCCA is 39 for Wild Type and 40 for IL-13 knocked out mouse asthma data, while 14 for GDS1423 and 43 for GDS2745.

3.1 Performance comparison

For performance comparisons, we have used z-score. Table 1 provides the values of z-score computed on the clusters obtained by the aforesaid algorithms using the datasets. z-score (Gibbons and Roth, 2002; Press *et al.*, 2003) 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 (Gibbons and Roth, 2002; Press *et al.*, 2003) tool is used. Saccharomyces Genome Database (SGD) annotation of the yeast genes, along with the gene ontology developed by the Gene Ontology Consortium (Ashburner *et al.*, 2000; Issel-Tarver *et al.*, 2002) 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.gov/projects/geo/gds) have been used. We consider GDS958 knocked out samples and wild type samples separately for clustering. A higher value of z indicates that genes would be better clustered by function, indicating a more biologically relevant clustering result.

Table 1 shows that z-scores corresponding to DCCA for all nine datasets are much larger than that corresponding to the other algorithms. This shows the results obtained by DCCA are much more biologically relevant than that generated by the others.

3.2 Functional enrichment: analysis and comparisons

The enriched functional categories for each cluster obtained by the DCCA on nine datasets are listed in Supplementry Tables 3–29. Some of the enriched functional categories for Yeast ATP Dataset obtained by MIND, K-means, PAM, FCM and GK are provided in Supplementary Table 30. 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 Funcassociate (Berriz *et al.*, 2003). P-value represents the probability of observing the number of genes from a specific GO functional category within each cluster. A low P-value indicates that the genes belonging to the enriched functional categories are biologically significant in the corresponding clusters. In the present article, only functional categories with P-value $<5.0 \times 10^{-7}$ are reported in order to restrict the size of the article.

3.2.1 Analysis Of the five clusters obtained for the Yeast ATP dataset (Supplementary Table 3), the highly enriched categories in cluster C_3 are the 'non-membrane-bound organelle' and the 'intracellular non-membrane-bound organelle' with P-value of 1.1×10^{-11} each. The cluster C_4 contains several enriched categories on 'cytosolic ribosome'. The highly enriched category in cluster C_4 is the 'cytosolic ribosome (sensu Eukaryota)/80S ribosome' with P-value of 5.2×10^{-14}. In the case of the Yeast PHO dataset (Supplementary Tables 4 and 5), 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 1.5×10^{-63}, the 'cytoplasm organization and biogenesis' and the 'ribosome biogenesis and assembly' with P-value of 3.4×10^{-63} each. The cluster C_3 contains several enriched categories on 'ribosome'. The cluster C_3 contains 'cytosolic ribosome' with P-value of 5.9×10^{-39} as the highly enriched category. The GO category 'structural constituent of ribosome/ribosomal protein' is also highly enriched in this cluster with P-value of 1.4×10^{-35}. The cluster C_{19} contains several enriched categories on 'biosynthesis'. The highly enriched category in cluster C_{19} is the

Table 1. z-scores on the clusters obtained by various algorithms

Dataset	Genes/conditions	Method	z-score	K
Yeast ATP	6215/3	DCCA	21.9	5
		MIND	4.56	3
		K-means	18.3	5
		PAM	18.9	5
		DIANA	−0.825	5
		FCM	18.1	5
		GK	19.6	5
Yeast PHO	6013/8	DCCA	29.8	52
		MIND	0.862	3
		K-means	20.2	52
		PAM	18	52
		DIANA	9.02	52
		FCM	19.5	52
		GK	23.6	52
Yeast AFR	6184/8	DCCA	26.2	67
		MIND	10.4	5
		K-means	21.5	67
		PAM	19.92	67
		DIANA	3.66	67
		FCM	23.6	67
		GK	22.7	67
Yeast AFRt	6190/7	DCCA	31.4	41
		MIND	15.7	5
		K-means	26.2	41
		PAM	27.4	41
		DIANA	2.56	41
		FCM	28.4	41
		GK	30.6	41
Yeast Cho *et al.*	6457/17	DCCA	49.4	138
		MIND	39.2	6
		K-means	44.6	138
		PAM	46.5	138
		DIANA	18.8	138
		FCM	35.8	138
		GK	39	138
GDS958 Wildtype	22690/6	DCCA	18.7	39
		MIND	1.56	5
		K-means	9.1	39
		PAM	10.3	39
		DIANA	−0.915	39
		FCM	12.9	39
		GK	15.3	39
GDS958 Knocked out	22690/6	DCCA	17.9	40
		MIND	1.39	4
		K-means	10.7	40
		PAM	10.1	40
		DIANA	−0.831	40
		FCM	11.3	40
		GK	14.6	40
GDS1423	22283/4	DCCA	37.1	14
		MIND	12.4	7
		K-means	33.6	14
		PAM	35.4	14
		DIANA	3.43	14
		FCM	29.2	14
		GK	31.6	14
GDS2745	22645/6	DCCA	30.7	43
		MIND	3.4	4
		K-means	26.3	43
		PAM	28.5	43
		DIANA	4.1	43
		FCM	24.6	43
		GK	29.4	43

'biosynthesis/anabolism' with P-value of 2.5×10^{-25}. The GO category 'cellular biosynthesis' is also highly enriched in this cluster with P-value of 4.4×10^{-19}.

For the Yeast AFR dataset (Supplementary Tables 6 and 7), the highly enriched category in cluster C_4 is the 'biosynthesis/anabolism' with P-value of 1.1×10^{-9}. The GO category 'cellular biosynthesis' is also highly enriched in this cluster with P-value of 1.9×10^{-9}. The cluster C_{11} contains several enriched categories on 'biogenesis'. The 'ribosome biogenesis' with P-value of 4.2×10^{-13}, the 'cytoplasm organization and biogenesis' and the 'ribosome biogenesis and assembly' with P-value of 1.1×10^{-11} each are some enriched categories in the cluster C_{11}. The cluster C_{30} contains several enriched categories on 'ribosome'. The highly enriched categories in the cluster C_{30} are the 'cytosolic ribosome (sensu Eukaryota)/80S ribosome' with P-value of 1.7×10^{-14} and the 'ribosome' with P-value of 1.4×10^{-12}.

As in the above datasets, for the Yeast AFRt dataset (Supplementary Tables 8 and 9), the cluster C_4 contains several enriched categories on 'biogenesis'. The highly enriched categories in cluster C_4 are the 'cytoplasm organization and biogenesis' and the 'ribosome biogenesis and assembly' with P-value of 6.1×10^{-33} each, the 'ribosome biogenesis' with P-value of 2.2×10^{-32}. The cluster C_{17} contains several enriched categories on 'ribosome'. The highly enriched categories in cluster C_{17} are the 'ribonucleoprotein complex/RNP' with P-value of 1.4×10^{-28} and the 'ribosome' with P-value of 2.1×10^{-28}.

In the case of the Yeast Cho *et al.* dataset (Supplementary Tables 10–12), 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 4.1×10^{-24}, the 'cytoplasm organization and biogenesis' and the 'ribosome biogenesis and assembly' with P-value of 1.9×10^{-23} each. The cluster C_{128} contains several enriched categories on 'ribosome'. The highly enriched category in cluster C_{128} is the 'cytosolic ribosome' with P-value of 1×10^{-132}. Two other highly enriched categories in cluster C_{128} are the 'ribosome' with P-value of 3.2×10^{-108} and the 'structural constituent of ribosome/ribosomal protein' with P-value of 2.5×10^{-99}.

The categories 'ribosome' (in C_4 for Yeast ATP, C_3 for Yeast PHO, C_{30} for Yeast AFR, C_{17} for Yeast AFRt and in C_{128} for Yeast Cho *et al.* datasets) and 'biogenesis' (in C_3 for Yeast ATP, C_1 for Yeast PHO, C_{11} for Yeast AFR, C_4 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. Similarly the category 'biosynthesis' (in C_4 for Yeast ATP, C_{19} for Yeast PHO, C_4 for Yeast AFR, C_{17} for Yeast AFRt and in C_{128} for Yeast Cho *et al.* datasets) is also enriched in at least one of the clusters for all the yeast datasets. This similarity in results from different datasets shows consistency of DCCA.

In the case of the GDS958 Wildtype dataset (Supplementary Table 13), the highly enriched categories in cluster C_3 are the 'motor activity' with P-value of 1.9×10^{-15}, the 'transmembrane receptor protein serine/threonine kinase activity' and the 'transforming growth factor beta receptor activity' with P-value of 2.5×10^{-15} each. The highly enriched categories in cluster C_{27} are the 'hydrolase activity' with P-value of 2.4×10^{-15} and the 'MHC class II receptor activity' with P-value of 2.7×10^{-15}. In the case of the GDS958 IL-13 Knockedout dataset (Supplementary Table 14), the highly enriched categories in cluster C_4 are the 'hydrolase activity' with P-value of 2.1×10^{-16} and the 'receptor activity' with P-value of 5.7×10^{-15}. The GO category 'structural constituent of ribosome' is highly enriched in cluster C_8 with P-value of 1.1×10^{-9} and in cluster C_{10} with P-value of 7.3×10^{-8}.

For the GDS1423 dataset (Supplementary Tables 15–26), all 14 clusters are found enriched with a total of 1000 enriched attributes. The highly enriched category in cluster C_8 is the 'multicellular organismal process' with P-value of 3.3×10^{-94}.

The cluster C_1 obtained from the GDS2745 dataset (Supplementary Tables 27–29) contains several enriched categories on 'intracellular organelle'. The highly enriched category in cluster C_1 is the 'intracellular membrane-bound organelle' with P-value of 1.0×10^{-23}.

From the results of Tables 3–29 in Supplementary Material, we see that the clusters obtained by the DCCA shows a high enrichment of functional categories.

3.2.2 Comparisons Here we describe the ability of detecting functionally enriched clusters/categories by the aforesaid clustering algorithms. Table 3 in Supplementary Material shows three out of five clusters produced by DCCA of Yeast ATP dataset, contain functionally enriched categories. Similarly, for GK (Supplementary Table 30), three out of five clusters of Yeast ATP dataset are functionally enriched, but total number of enriched categories for clusters generated by DCCA (28) are greater than those generated by GK (25). For MIND, K-means, PAM and FCM, among five clusters generated from Yeast ATP dataset (Supplementary Table 30), only two clusters contain functionally enriched categories. DIANA could not find any enriched functional category. This result, for Yeast ATP dataset, clearly shows that DCCA 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, DCCA provides greater number of enriched categories (Table 2) compared to the other algorithms.

We have also found that the NMF technique by Kim and Tidor (2003) is able to obtain 87 enriched attributes for Yeast Cho *et al.* dataset, whereas DCCA is able obtain 187 enriched attributes for the same dataset indicating superiority of DCCA over NMF-based technique. In order to restrict the size of the article, we have not included the detailed results.

4 CONCLUSIONS

We have presented here a novel clustering algorithm, called DCCA, which is able to obtain clustering solution from gene-expression dataset with very high biological significance. DCCA is able to detect clusters containing genes with similar variation in pattern of expression profiles, without taking the expected number of clusters as an input. The algorithm continues clustering until all clusters contain only positively correlated sets of genes. Like some other algorithms, DCCA also belongs to the category of hierarchical divisive clustering algorithms. Analysis of the results shows that clustering solution obtained by DCCA is more biologically significant than that obtained by some other algorithms, namely, MIND, K-means, PAM, DIANA, FCM, GK and an NMF-based

Table 2. Comparative score of various clustering algorithms with respect to ability of detecting functionally enriched clusters/categories. Here we have considered the categories for which P-value $< 5.0 \times 10^{-7}$

Dataset	Method	Total clusters	Enriched clusters	Enriched attributes
Yeast ATP	DCCA	5	3	28
	MIND	3	2	2
	K-means	5	2	11
	PAM	5	2	13
	DIANA	5	0	0
	FCM	5	2	15
	GK	5	2	25
Yeast PHO	DCCA	52	8	113
	MIND	3	0	0
	K-means	52	6	89
	PAM	52	6	81
	DIANA	52	2	17
	FCM	52	5	73
	GK	52	7	106
Yeast AFR	DCCA	67	10	89
	MIND	5	2	18
	K-means	67	7	71
	PAM	67	6	62
	DIANA	67	0	0
	FCM	67	8	69
	GK	67	8	78
Yeast AFRt	DCCA	41	7	107
	MIND	5	3	46
	K-means	41	6	91
	PAM	41	7	99
	DIANA	41	0	0
	FCM	41	6	86
	GK	41	7	101
Yeast Cho *et al.*	DCCA	138	18	187
	MIND	6	3	115
	K-means	138	14	143
	PAM	138	15	157
	DIANA	138	4	39
	FCM	138	15	107
	GK	138	14	131
GDS958 Wildtype	DCCA	39	16	50
	MIND	5	2	6
	K-means	39	11	36
	PAM	39	13	41
	DIANA	39	0	0
	FCM	39	14	43
	GK	39	14	46
GDS958 Knocked out	DCCA	40	11	57
	MIND	4	0	0
	K-means	40	9	41
	PAM	40	9	44
	DIANA	40	0	0
	FCM	40	9	49
	GK	40	10	51
GDS1423	DCCA	14	14	1000
	MIND	7	3	126
	K-means	14	13	856
	PAM	14	14	931
	DIANA	14	2	79
	FCM	14	11	788
	GK	14	11	814
GDS2745	DCCA	43	32	202
	MIND	4	1	16
	K-means	43	30	179
	PAM	43	30	187
	DIANA	43	1	19
	FCM	43	29	174
	GK	43	31	192

algorithm. Despite these benefits of the DCCA, several issues require further investigations. First, the computational cost of the DCCA for repairing any misplacement occurring in clustering step needs to be reduced. Second, the quality of the clusters obtained by DCCA depends on the choice of correlation coefficient. In this article, we have used Pearson correlation coefficient as a similarity measure. However, other measures with the similar properties could be used for further study. Third, DCCA will not work if dataset contains less than three samples. In this case calculated correlation value will be either $+1$ or -1. Fourth, the concept of DCCA needs to be modified in order to develop a suitable biclustering algorithm.

Conflict of Interest: none declared.

REFERENCES

Alexe,G. *et al.* (2002) Consensus algorithms for the generation of all maximal bicliques. In *Technical Report TF-DIMACS*.

Alon,N. *et al.* (2005) Quadratic forms on graphs. In *37th ACM Symposium on Theory of Computing (STOC)*.

Ashburner,M. *et al.* (2000) Tool for the unification of biology. The gene ontology consortium. *Nat. Genet.*, **25**, 25–29.

Bansal,N. *et al.* (2004) Correlation clustering. *Mach. Learn., Special Issue*, **56**, 89–113.

Ben-Dor,A. *et al.* (2002) Discovering local structure in gene expression data: the order-preserving submatrix problem. *Proceedings of the Sixth International Conference on Computational Biology (RECOMB'02)*, pp. 49–57.

Berriz,F.G. *et al.* (2003) Characterizing gene sets with funcassociate. *Bioinformatics*, **19**, 2502–2504.

Bezdek,J.C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.

Bezdek,J.C. *et al.* (1984) FCM: Fuzzy c-means algorithm. *Comput. Geosci.*, **10**, 191–203.

Charikar,M. and Wirth,A. (2004) Maximizing quadratic programs: extending grothendieck's inequality. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 524–533.

Charikar,M. *et al.* (2003) Clustering with qualitative information. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 524–533.

Cheng,Y. and Church,G.M. (2000) Biclustering of expression data. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **8**, 93–103.

Cho,R.J. *et al.* (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, **2**, 65–73.

Cohen,W. and Richman,J. (2002) Learning to match and cluster large high-dimensional data sets for data integration. In *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Demaine,E.D. and Immorlica,N. (2003) Correlation clustering with partial information. In *Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 7th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM-APPROX 2003)*. Princeton, New Jersey, pp. 1–13.

Demaine,E.D. *et al.* (2006) Correlation clustering in general weighted graphs. *Theoret. Comput. Sci.*, **361**, 172–187.

Dembele,D. and Kastner,P. (2003) Fuzzy c-means method for clustering microarray data. *Bioinformatics*, **19**, 973–980.

Dunn,J.C. (1973) A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *J. Cybernet.*, **3**, 32–57.

Getz,G. *et al.* (2000) Coupled two-way clustering analysis of gene microarray data. *Proc. Natl Acad. Sci USA*, 12079–12084.

Gibbons,F. and Roth,F. (2002) Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Res.*, **12**, 1574–1581.

Gustafson,E.E. and Kessel,W.C. (1979) Fuzzy clustering with a fuzzy covariance matrix. In *Proceedings of the IEEE CDC*. San Diego, California, pp. 761–766.

Han,J. and Kamber,M. (2001) *Data Mining: Concepts and Techniques*. Morgan Kaufmann, CA, USA.

Ihmels,J. *et al*. (2004) Defining transcription modules using large-scale gene expression data. *Bioinformatics*, **20**, 1993–2003.

Issel-Tarver,L. *et al*. (2002) Saccharomyces genome database. *Methods Enzymol*, **350**, 329–346.

Jain,A.K. and Dubes,R.C. (1988) *Algorithms for Clustering Data*. Prentice Hall, New Jersey.

Kim,D.W. *et al*. (2005) Detecting clusters of different geometrical shapes in microarray gene expression data. *Bioinformatics*, **21**, 1927–1934.

Kim,P.M. and Tidor,B. (2003) Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome Res.*, **13**, 1706–1718.

Kluger,Y. *et al*. (2003) Spectral biclustering of microarray cancer data: co-clustering genes and conditions. *Genome Res.*, 703–716.

Lukashin,A.V. and Fuchs,R. (2001) Analysis of temporal gene expression profiles:clustering by simulated annealing and determining the optimal number of clusters. *Bioinformatics*, **17**, 405–414.

Mitra,S. and Acharya,T. (2003) *Data Mining: Multimedia, Soft Computing, and Bioinformatics*. John Wiley, New York.

Murali,T.M. and Kasif,S. (2003) Extracting conserved gene expression motifs from gene expression data. In *Proceeding of the Pacific Symposium on Biocomputing*, pp. 77–88.

Prelic,A. *et al*. (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, **22**, 1122–1129.

Press,W. *et al*. (2003) *Numerical Recipies – The Art of Scientific Computing*. Cambridge University Press, Cambridge.

Qin,J. *et al*. (2003) Kernel hierarchical gene clustering from microarray expression data. *Bioinformatics*, **19**, 2097–2104.

Sharan,R. *et al*. (2003) Click and expander: a system for clustering and visualizing gene expression data. *Bioinformatics*, **19**, 1787–1799.

Tanay,A. *et al*. (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, S136–S144.

Xu,Y. *et al*. (2002) Clustering gene expression data using a graph-theoretic approach: an application of minmum spanning trees. *Bioinformatics*, **18**, 536–545.