A pattern matching approach for clustering gene expression data

Rosy Das*
Department of Computer Science and Engineering,
Tezpur University,
Napaam, Assam, 784028, India
E-mail: rosy8@tezu.ernet.in
*Corresponding author

Jugal Kalita
Department of Computer Science,
University of Colorado,
Colorado Springs, CO 809933, USA
E-mail: jkalita@uccs.edu

Dhruba K. Bhattacharyya
Department of Computer Science and Engineering,
Tezpur University,
Napaam, Assam, 784028, India
E-mail: dkb@tezu.ernet.in

Abstract: Identifying groups of genes with similar expression time courses is crucial in the analysis of gene expression time series data. This paper proposes a regulation-based clustering approach, PatternClus, for clustering gene expression data. The method also identifies sub-clusters based on an order preserving ranking approach. The clustering method was experimented in light of real life datasets and the proposed method has been established to perform satisfactorily. PatternClus was compared to some of the well-known clustering algorithms (k-means and hierarchical algorithm) and was found to give better results in terms of z-score measure of cluster validation. An incremental version of PatternClus is also presented here which helps in identifying clusters incrementally where the database is continuously increasing.

Keywords: gene expression; microarray; regulation pattern; matching; clustering; sub-cluster; incremental clustering.


Biographical notes: Rosy Das is an Assistant Professor in the Department of Computer Science and Engineering, Tezpur University, Tezpur, India. Her research interests include clustering and bioinformatics.
1 Introduction

The advent of DNA microarray technologies has enabled the monitoring of expression levels of a large number of genes across different experimental conditions. DNA chips measure the expression level of thousands of genes, perhaps all genes of an organism, within a number of different experimental conditions (samples) (Stekel, 2006). The conditions may correspond to different time points, different organs, different experimental conditions or may have come from cancerous or healthy tissues, or even from different individuals. Visually this kind of data, which is widely known as gene expression data or simply expression data is difficult and extracting knowledge of biological relevance from such data is highly challenging.

Microarrays measure the activity (expression level) of the genes under varying conditions. Expression level is estimated by measuring the amount of mRNA for that particular gene. Microarray data or gene expression data is organised together as a gene expression matrix with rows corresponding to genes and columns to conditions. This can be viewed as a $G \times T$ matrix where each of the $G$ rows represents a gene (or a clone, ORF, etc.) and each of the $T$ columns represents a condition. Each entry in the matrix represents the expression level of a gene under a condition. It can either be an absolute value (e.g., Affymetrix GeneChip) or a relative expression ratio (e.g., cDNA microarrays). A row/column is sometimes referred to as the 'expression profile' of the gene/condition. If two genes are related (having similar functions or are co-regulated), their expression profiles should be similar (e.g., low Euclidean distance or high correlation).

According to Jiang et al. (2003b), most data mining algorithms developed for gene expression data deal with the problem of clustering. Clustering methods can be used to group either genes or conditions. There is a third way of clustering gene expression data which is by grouping subsets of genes under subsets of conditions. Such discovery of local patterns in gene expression data is known as ‘biclustering’ or subspace clustering (Jiang et al., 2003b). We will, however, consider the first objective in this paper.

Clustering gene expression data identifies subsets of genes that behave similarly along a course of time (conditions, samples, etc.). Genes in the same cluster have similar expression patterns. A detailed survey on clustering gene expression data is given in Jiang et al. (2003b). Co-expressed genes can be grouped into clusters based on their expression patterns (Jiang et al., 2003b). In such gene-based clustering (objective 1), the genes are treated as the objects, while the samples are the features. In sample-based clustering (objective 2), the samples can be partitioned into homogeneous groups where
the genes are regarded as features and the samples as objects. Both the **gene-based** and **sample-based** clustering approaches search for exclusive and exhaustive partitions of objects that share the same feature space. The third category, that is **subspace clustering**, captures clusters formed by a subset of genes across a subset of samples. For subspace clustering algorithms, either genes or samples can be regarded as objects or features. In this paper, we use a gene-based clustering approach.

## 2 Related work

Clustering identifies subsets of genes that behave similarly along a course of time. Genes in the same cluster have similar expression patterns. Gene expression data clustering techniques can be categorised as follows: **partitioning**, **hierarchical**, **density-based**, **model-based** and **graph-based**. A detailed survey on clustering gene expression data is given in Jiang et al. (2003b).

### 2.1 Partitional approaches

K-means (Tavazoie et al., 1999) is a typical partition-based clustering algorithm which divides the data into pre-defined number of clusters in order to optimise a predefined criterion. The major advantages of it are its simplicity and speed, which allows it to run on large datasets. However, it may not yield the same result with each run of the algorithm. It often can be found incapable of handling outliers and is not suitable to detect clusters of arbitrary shapes. A self-organising map (SOM) (Tamayo et al., 1999) is more robust than K-means for clustering noisy data. It requires the number of clusters and the grid layout of the neuron map as user input. Specifying the number of clusters in advance is difficult in case of gene expression data. Moreover, partitioning approaches are restricted to data of lower dimensionality, with inherent well-separated clusters of high density. But, gene expression datasets may be high dimensional and often contain intersecting and embedded clusters. Quality threshold (QT) clustering (Heyer et al., 1999) is an alternative method of partitioning data, invented for gene clustering. It requires more computing power than k-means, but does not require specifying the number of clusters a priori, and always returns the same result when run several times. The distance between a point and a group of points is computed using complete linkage, i.e., as the maximum distance from the point to any member of the group (Eisen et al., 1998). Hierarchical structure can also be built based on SOM such as self-organising tree algorithm (SOTA) (Dopazo and Carazo, 1997). Recently, several new algorithms such as Herrero et al. (2001) and Tomida et al. (2002) have been proposed based on the SOM algorithm. These algorithms can automatically determine the number of clusters and dynamically adapt the map structure to the data distribution. Herrero et al. (2001) extend the SOM by a binary tree structure. At first, the tree only contains a root node connecting two neurons. After a training process similar to that of the SOM algorithm, the dataset is segregated into two subsets. Then the neuron with less coherence is split in two new neurons. This process is repeated level by level, until all the neurons in the tree satisfy some coherence threshold. Other examples of SOM extensions are fuzzy adaptive resonance theory (Fuzzy ART) (Tomida et al., 2002) which provide some approaches to measure the coherence of a neuron. The output map is adjusted by splitting the existing neurons or adding new neurons into the map, until the coherence of each neuron in the map satisfies a user-specified threshold.
2.2 Hierarchical approaches

Hierarchical clustering generates a hierarchy of nested clusters. These algorithms are divided into agglomerative and divisive approaches. Unweighted pair group method with arithmetic mean (UPGMA), presented in Eisen et al. (1998), adopts an agglomerative method to graphically represent the clustered dataset. However, it is not robust to noise. In Alon et al. (1999), the genes are split through a divisive approach, called the deterministic-annealing algorithm (DAA). The divisive correlation clustering algorithm (DCCA) (Bhattacharya and De, 2008) uses Pearson’s correlation as the similarity measure. All genes in a cluster have highest average correlation with genes in that cluster. Hierarchical clustering not only groups together genes with similar expression patterns but also provides a natural way to graphically represent the dataset allowing a thorough inspection. However, a small change in the dataset may greatly change the hierarchical dendrogram structure. Another drawback is its high computational complexity.

2.3 Density-based approaches

Density-based clustering identifies dense areas in the object space. Clusters are highly dense areas separated by sparsely dense areas. A kernel density clustering method for gene expression profile analysis is reported in Shu et al. (2003). It assumes no parametric statistical model and does not rely on any specific probability distribution. Hyper-spherical uniform kernels of variable radius are used and density estimate of the data points are found. The method is robust and less sensitive to outliers. However, accurate density estimation and assignment of cluster membership require multiple data points in near neighbourhoods and thus density estimation is less accurate when cluster size is small. In Jiang et al. (2003a), the authors propose the density-based hierarchical clustering (DHC) method that uses a density-based approach to identify co-expressed gene groups from gene expression data. It considers clusters as high dimensional dense areas where the genes are ‘attracted’ to each other. DHC uses two-level hierarchical structures (attraction tree and density tree) to organise the cluster structure of the dataset. The attraction tree reflects relationships among genes in the dense area. Each node in the attraction tree represents a gene and its parent is the attractor of it. The highest density gene becomes the root of the tree. The attraction tree becomes complicated for large datasets and hence the cluster structure is summarised in a density tree. Each node of the density tree represents a dense area. Initially the whole dataset is considered a single dense area represented by the root node of the density tree. This dense area is then split into several sub-dense areas based on some criteria where each sub-dense area is represented by a child node of the root node. The sub-dense areas are further split till each sub-dense area contains a single cluster. DHC is suitable for detecting highly connected clusters but is computationally expensive and is dependent on two global parameters. Density-based approach discovers clusters of arbitrary shapes even in presence of noise (Ester et al., 1996). However, density-based clustering techniques suffer from high computational complexity with the increase in dimensionality (even if spatial index structure is used) and input parameter dependency.

An alternative technique is to define the similarity of points in terms of their shared nearest neighbours. This idea was first introduced by Jarvis and Patrick (1973). In Chung et al. (2004), a $k$-nearest neighbour-based density estimation technique has been exploited. The density-based algorithm proposed by Chung et al. (2004) works in three
phases: density estimation for each gene, rough clustering using core genes and cluster refinement using border genes. Density of a gene is calculated by the sum of similarities among its \( k \) nearest neighbours. Core genes are high density genes and the method proceeds by clustering core genes to form the rough clusters. Once the rough clusters are formed, the border genes are assigned to the most relevant cluster. In Syamala et al. (2006), the authors present a density and shared nearest neighbour-based clustering method. The similarity measure used is that of Pearson’s correlation and the density of a gene is given by the sum of its similarities with its neighbours. The shared nearest neighbours of the dense genes are found and merged into the same cluster. The merging is done efficiently using a data structure called the P-tree (Perrizo, 2001).

### 2.4 Model-based approaches

Model-based approaches provide a statistical framework to model the cluster structure in gene expression data. The expectation maximisation (EM) algorithm (Dempster et al., 1977) discovers good values for its parameters iteratively. It can handle various shapes of data and can be very expensive since a large number of iterations may be required. In Travis and Huang (2009), a signal shape similarity method is used to cluster genes using a variational Bayes expectation maximisation algorithm (Beal and Ghahramani, 2003). A model-based approach provides an estimated probability that a data object will belong to a particular cluster. Thus, a gene can have high correlation with two totally different clusters. However, the model-based approach assumes that the dataset fits a specific distribution which is not always true.

### 2.5 Graph-based approaches

In graph-based clustering algorithms, graphs are built as combinations of objects, features or both, as the nodes and edges, and partitioned by using graph theoretic algorithms. Graph theoretic algorithms are also used for the problem of clustering cDNAs based on their oligo-nucleotide fingerprints (Hartuv et al., 1999). Cluster identification via connectivity kernels (CLICK) (Sharan and Shamir, 2000) is suitable for subspace and high dimensional data clustering. The cluster affinity search technique (CAST) by Ben-Dor et al. (1999) takes as input the pairwise similarities between genes and an affinity threshold. It does not require a user-defined number of clusters and handles outliers efficiently. But, it faces difficulty in determining a good threshold value. In CAST, the size and number of clusters produced is directly affected by the fixed user-defined parameter \( t \) and hence, apriori knowledge of the dataset is required. To overcome this problem, E-CAST (Bellaachia et al., 2002) calculates the threshold value dynamically based on the similarity values of the objects that are yet to be clustered.

### 2.6 Soft computing approaches

Fuzzy c-means (Bezdek, 1981) and genetic algorithms (GA) [such as Bandyopadhyay et al. (2007) and Maulik et al. (2009)] have been used effectively in clustering gene expression data. The fuzzy c-means algorithm requires the number of clusters as an input parameter. The GA-based algorithms have been found to detect biologically relevant clusters but are dependent on proper tuning of the input parameters.

The current information explosion, fuelled by the availability of World Wide Web and the huge amount of microarray experiments conducted has led to the ever-increasing
volume of data. There is therefore a need to introduce incremental clustering so that updates can be clustered in an incremental manner.

2.7 Incremental clustering approaches

In Ester et al. (1998), the authors present an incremental clustering approach based on the DBSCAN (Ester et al., 1996) algorithm. The main idea behind the algorithm is that the insertion or deletion of an object affects the current clustering only in the neighbourhood of this object. Density connections may surface or get removed depending on whether an object is added or deleted respectively. Incremental DBSCAN yields the same result as DBSCAN executed over the whole updated database. In Can (1993), the authors propose an incremental clustering for dynamic processing. In Tao and Sarabjot (2008), the authors present a one pass clustering algorithm for relational datasets. Rough set theory has been employed in the incremental approach for clustering interval datasets in Asharaf et al. (2006). It groups the given dataset into a set of overlapping clusters by employing a rough variant of the Leader algorithm. The algorithm generates cluster abstractions in a single scan and is robust to outliers.

In Charikar et al. (1997), an incremental clustering model for information retrieval applications has been presented. In Cheung et al. (1996) and Feldman et al. (1997), efficient methods for modifying a set of association rules has been reported. Though a lot of research has been done on incremental clustering for other application domains, yet incremental clustering in gene expression data has not been exploited much. However, some amount of research has been devoted to this field. In Lu et al. (2004b), an incremental genetic k-means algorithm (IGKA) has been presented. IGKA calculates the objective value total within-cluster variation (TWCV) and clusters centroids incrementally whenever the mutation probability is small. IGKA converges to the global optimum. In the genetic k-means algorithm (GKA) proposed in Krishna and Murty (1999), a GA is hybridised with the k-means algorithm and therefore GKA converges to the global optimum faster than other GAs. In Lu et al. (2004a), the authors present a faster version of GKA (FGKA) which efficiently evaluates the TWCV, avoids illegal string termination overhead and simplifies the mutation operator. IGKA inherits all the advantages of FGKA and outperforms FGKA when the mutation probability is small. The cost of calculating the centroids in FGKA is more expensive when the mutation probability is small than when it is calculated incrementally in IGKA. The hybrid genetic k-means algorithm (HGKA) in Lu et al. (2004b) combines the advantages of both IGKA and FGKA and obtains an even better performance. However, it is very difficult to obtain the threshold value which is dataset dependent. In Ruiz et al. (2006), an incremental gene selection algorithm which reduces search space complexity using a wrapper-based method is presented. This method works on the ranking directly. In BIRS (Ruiz et al., 2006), genes are first ranked w.r.t. an evaluation measure. Then, the set of genes is updated with by crossing the ranking from the beginning to the last ranked gene. Classification accuracy with the first gene in the list is obtained and it is marked as selected. The classification rate is again obtained and the second gene is selected depending on whether the classification accuracy is significantly better. The process is repeated till the last gene on the ranked list is processed. The algorithm returns the best subset formed and it does not contain irrelevant or redundant genes.
2.8 Discussion

Based on our selected survey following observations have been made:

- Clustering approaches are highly dependent on the similarity measure being used. However, it can be observed that the inherent high dimensional space of gene expression data renders most of the similarity measures ineffective w.r.t. quality clusters.

- Most of the clustering algorithms are input parameter dependent and these parameters play an important role in quality cluster detection.

- Incremental clustering over gene expression data has not been exploited much. PatternClus is free from the initial guess of the number of clusters. Also, it does not use any of the existing proximity measures as the regulation matching approach of it can itself be found capable of identifying coherent patterns. Moreover, it is also independent of any input parameters. An incremental version of PatternClus is also presented to handle incremental data.

3 The proposed method

In PatternClus, the gene expression data is normalised and a discretisation technique is applied to discretise the data. Clustering is then run on the discretised data.

The gene expression data is normalised to have mean 0 and standard deviation 1. Expression data having a low variance across conditions as well as data having more than three-fold variation are filtered. Discretisation is then performed on this normalised expression data. Discretisation uses the regulation information, i.e., up- or down-regulation in each of the conditions for a particular gene. Here, let \( G \) be the set of all genes and \( T \) be the set of all conditions. The discretisation is done as follows:

1. The discretised value of gene \( g_i \) at condition, \( t_1 \) (i.e., the first condition)

\[
\xi_{g_i, t_1} = \begin{cases} 
1 & \text{if } \varepsilon_{g_i, t_1} > 0 \\
0 & \text{if } \varepsilon_{g_i, t_1} = 0 \\
-1 & \text{if } \varepsilon_{g_i, t_1} < 0
\end{cases}
\]

2. The discretised values of gene \( g_i \) at conditions \( t_j \) (\( j = 1, \ldots (T - 1) \)), i.e., at the rest of the conditions \( (T - \{t_1\}) \)

\[
\xi_{g_i, t_{j+1}} = \begin{cases} 
1 & \text{if } \varepsilon_{g_i, t_j} < \varepsilon_{g_i, t_{j+1}} \\
0 & \text{if } \varepsilon_{g_i, t_j} = \varepsilon_{g_i, t_{j+1}} \\
-1 & \text{if } \varepsilon_{g_i, t_j} > \varepsilon_{g_i, t_{j+1}}
\end{cases}
\]

where \( \xi_{g_i, t_j} \) is the discretised value of gene \( g_i \) at condition \( t_j \) (\( j = 1, \ldots (T - 1) \)). The expression value of gene \( g_i \) at condition \( t_j \) is given by \( \varepsilon_{i, j} \). We see in the above computation that the first condition, \( t_1 \), is treated as a special case and it’s discretised value is directly based on \( \varepsilon_{i, t_1} \), i.e., the expression value at condition \( t_1 \). For the rest of the conditions the discretised value is calculated by comparing its expression value with that of the previous value. This helps in finding whether the gene is up- (1) or -down...
A pattern matching approach for clustering gene expression data

(-1) regulated at that particular condition. Each gene will now have a regulation pattern ($\phi$) of 0, 1, and -1 across the conditions or time points. Figures 1 and 2 shows an example dataset and the profile plot respectively and Figure 3 illustrates its regulation pattern.

**Figure 1** Example dataset

<table>
<thead>
<tr>
<th></th>
<th>1.1061</th>
<th>1.066</th>
<th>1.1061</th>
<th>-0.42027</th>
<th>-1.47775</th>
<th>0.110567</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.41575</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.716305</td>
<td>1.56754</td>
<td>0.2436</td>
<td>-0.27982</td>
<td>-0.48637</td>
<td>-1.29406</td>
<td></td>
</tr>
<tr>
<td>0.278431</td>
<td>1.673058</td>
<td>0.144933</td>
<td>-0.15183</td>
<td>-0.66517</td>
<td>-1.2794</td>
<td></td>
</tr>
<tr>
<td>0.2378</td>
<td>1.066</td>
<td>1.066</td>
<td>-0.72985</td>
<td>-1.42057</td>
<td>0.306216</td>
<td></td>
</tr>
<tr>
<td>0.63311</td>
<td>0.978443</td>
<td>0.978443</td>
<td>-0.31655</td>
<td>-1.13672</td>
<td>-1.13672</td>
<td></td>
</tr>
<tr>
<td>-0.144</td>
<td>0.975743</td>
<td>0.975743</td>
<td>-0.0244</td>
<td>-0.16078</td>
<td>-1.75191</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2** Profile plot of the example dataset (see online version for colours)

![Profile plot](image)

**Figure 3** Discretised matrix

```
2 1 0 2 2 1
1 1 2 2 2 2
1 1 2 2 2 2
2 1 0 2 2 1
1 1 0 2 2 0
2 1 0 2 2 2
```
The clustering of genes proceeds with the discretised matrix by finding the maximal matching genes with respect to regulation pattern. The cluster expansion starts with an arbitrary gene (the initiator) and finds the maximal matching regulation pattern w.r.t. it. From the remaining genes, those genes having the same maximal matching w.r.t. the initiator are grouped into the same cluster and then sub-cluster identification starts. Sub-cluster identification, looks for order preserving genes in a particular cluster. Genes in a cluster are grouped into a sub-cluster by analysing their variational patterns with respect to time, dosage, patient age, etc. Sub-clusters reveal more information about functionally similar genes than is possible with the clustering of their intensity values alone. The process of cluster identification continues with the next gene from the set of unclassified genes. The process continues till all the genes are classified. The identification of clusters is based on the following definition:

**Definition 1:** Let \( \varphi_{g_i} \) and \( \varphi_{g_j} \) be the regulation pattern of two genes \( g_i \) and \( g_j \). Then, the matching \( (M) \) between \( g_i \) and \( g_j \) will be given by the \( \text{reg}. \text{score} \) between the two regulation patterns, i.e., \( M(g_i, g_j) = \text{reg}. \text{score}(\varphi_{g_i}, \varphi_{g_j}) \), where \( \text{reg}. \text{score} \) is the number of condition-wise common regulation values excluding condition 1.

**Definition 2:** A pair of genes \((g_i, g_j)\) is called maximally matched \((MM)\) if \( \forall g_j \in G - g_i \, M(g_i, g_j) \geq \delta \) and maximal.

**Definition 3:** For a pair of maximally matched gene pair \((g_i, g_j)\) over conditions \( (t = 2, 3, \cdots, T - 1) \), maximally matched regulation pattern (MMRP) \( \varphi'_{g_i} \) and \( \varphi'_{g_j} \) are defined over the subset of conditions where both \( \varphi_{g_i} \) and \( \varphi_{g_j} \) match.

MMRP of a gene \( g_i \) is computed w.r.t. another gene \( g_j \) (\( g_i \) and \( g_j \) are maximally matched) as follows. Here, \( t \) refers to the conditions \( (t = 2, 3, \cdots, T - 1) \).

\[
\varphi'_{g_i} = \varphi'_{g_j} = \begin{cases} 
1 & \text{if } \varphi_{g_i,t} = \varphi_{g_j,t} = 1 \\
0 & \text{if } \varphi_{g_i,t} = \varphi_{g_j,t} = 0 \\
-1 & \text{if } \varphi_{g_i,t} = \varphi_{g_j,t} = -1 \\
x & \text{otherwise}
\end{cases}
\]

### 3.1 Basis of the clustering approach

The regulation matching and the order preserving are the two fundamental basis based on which the clustering technique is designed. Order preserving finds the sub-clusters within a cluster using rank order. Sub-clusters consist of those genes whose expressions vary across stages in the same way.

#### 3.1.1 Regulation matching

For a particular gene \( g_i \), the maximal matching regulation pattern is found. All those genes having the same maximally matched regulation pattern w.r.t. \( g_i \) are grouped into the same cluster.

#### 3.1.2 Order preserving

The experiments in \( T \) (total set of conditions) represent distinct stages in the progress of a disease or in a cellular process, and the expression levels of all genes in \( G \) vary across the stages in the same way. This algorithm looks for order-preserving subsets (or
A pattern matching approach for clustering gene expression data

sub-matrices) in which the expression levels of all genes induce the same linear ordering of the experiments. Our work builds on Ben-Dor et al. (2002) in the following way. For a condition set \( t \in T \) and a gene \( g \in G \), the conditions in \( T \) can be ordered in a way so that the expression values are sorted in ascending order. By order ranking, we search for the expression levels of genes within a cluster which induce ordering of the experiments (conditions).

Each gene will have a rank which will give the permutation order of that gene across conditions \( t \in T \). The rank is calculated according to the expression values of a gene across conditions, i.e., the elements of the rank pattern are given by their ranking in ascending order of their expression values. The rank computation of a gene is a two step process:

1. for each gene \( g_i \), find \( \psi'_{g_i} \),
2. rank \( g_i \) in ascending order according to the expression values where \( \psi'_{g_i,t} \neq x \).

For ease of understanding, the above concepts will be elaborated using the example given in Figure 1. Here, the rows will represent the genes \( g_1, g_2, \cdots, g_6 \) and the columns will represent the corresponding conditions (excluding condition 1 as stated before).

\[
\begin{align*}
\varphi_{g_1} &= 2 \ 1 \ 0 \ 2 \ 2 \ 1 \\
\varphi_{g_2} &= 1 \ 1 \ 2 \ 2 \ 2 \ 2 \\
\varphi_{g_3} &= 1 \ 1 \ 2 \ 2 \ 2 \ 2 \\
\varphi_{g_4} &= 2 \ 1 \ 0 \ 2 \ 2 \ 1 \\
\varphi_{g_5} &= 1 \ 1 \ 0 \ 2 \ 2 \ 0 \\
\varphi_{g_6} &= 2 \ 1 \ 0 \ 2 \ 2 \ 2
\end{align*}
\]

Matching among pairs of genes are:

\[
\begin{align*}
M(g_1, g_2) &= 3 & M(g_1, g_3) &= 3 & M(g_1, g_4) &= 5 \\
M(g_1, g_5) &= 4 & M(g_1, g_6) &= 4 & M(g_2, g_3) &= 5 \\
M(g_2, g_4) &= 3 & M(g_2, g_5) &= 3 & M(g_2, g_6) &= 4 \\
M(g_3, g_4) &= 3 & M(g_3, g_5) &= 3 & M(g_3, g_6) &= 4 \\
M(g_4, g_5) &= 4 & M(g_4, g_6) &= 4 & M(g_5, g_6) &= 4
\end{align*}
\]

Suppose \( \delta = 4 \), then maximal matching of pairs of genes are:

\[
\begin{align*}
MM(g_1, g_4) &= 5 & MM(g_1, g_5) &= 4 & MM(g_1, g_6) &= 4 \\
MM(g_2, g_3) &= 5 & MM(g_2, g_6) &= 4 & MM(g_3, g_6) &= 4 \\
MM(g_4, g_6) &= 4 & MM(g_4, g_6) &= 4 & MM(g_5, g_6) &= 4
\end{align*}
\]

MMRP is:

\[
\begin{align*}
\psi'_{g_1} &= 1 \ 0 \ 2 \ 2 \ 1 \\
\psi'_{g_2} &= 1 \ 0 \ 2 \ 2 \ 1 \\
\psi'_{g_3} &= 1 \ 2 \ 2 \ 2 \ 2 \\
\psi'_{g_4} &= 1 \ 2 \ 2 \ 2 \ 2 \\
\psi'_{g_5} &= 1 \ 0 \ 2 \ 2 \ x \\
\psi'_{g_6} &= 1 \ 0 \ 2 \ 2 \ x
\end{align*}
\]

From, the above example it is clear that the MMRPs of \( g_1 \) and \( g_4 \) are same; \( g_2 \) and \( g_3 \) are same and \( g_5 \) and \( g_6 \) are same.
Genes 1, 4, 2 and 3 have the MMRP over conditions 2, 3, 4, 5, and 6. Rank order over these four conditions are computed w.r.t. their expression values \( e_{i,j}, i = 1, 4, 2, 3 \) and \( j = 2, 3, 4, 5, 6 \), where \( i \) refers to gene \( i \) and \( j \) refers to condition \( j \) and ranks as follows:

\[
\text{Rank}(g_1) = 4 \quad 4 \quad 2 \quad 1 \quad 3 \\
\text{Rank}(g_4) = 4 \quad 4 \quad 2 \quad 1 \quad 3 \\
\text{Rank}(g_2) = 5 \quad 4 \quad 3 \quad 2 \quad 1 \\
\text{Rank}(g_5) = 5 \quad 4 \quad 3 \quad 2 \quad 1 \\
\]

Similarly, genes 5 and 6 have the MMRP over conditions 2, 3, 4 and 5 and ranks obtained are as follows:

\[
\text{Rank}(g_3) = 3 \quad 3 \quad 2 \quad 1 \\
\text{Rank}(g_6) = 3 \quad 3 \quad 2 \quad 1 \\
\]

**Definition 4 (Cluster):** A cluster \( C_i \) is a non-empty subset of genes having the same MMRP, i.e., \( \forall g_i, g_j, \psi_{g_i} = \psi_{g_j} \), where \( g_i, g_j \in C_i \).

**Definition 5 (Sub-clusters):** If \( g_i, g_j \in C_i \) and \( \text{Rank}(g_i) = \text{Rank}(g_j) \), then \( g_i, g_j \) will form a sub-cluster \( S_i \). Here \( S_i \subset C_i \) and \( \mid S_i \mid \geq 2 \).

**Definition 6 (Noise):** A gene \( g \) is said to be a noise gene if its regulation pattern does not match with any of the genes belonging to the clusters so formed.

The clustering of PatternClus is discussed next.

### 3.2 The clustering procedure

Cluster identification starts with an arbitrary gene and finds the maximal matching regulation pattern with the other unclassified genes. For the regulation pattern matching, two genes are matched w.r.t. the regulation across the conditions starting from condition 2. Genes having the maximal matching regulation pattern in a particular iteration of the clustering process are grouped into the same cluster. The expansion process then continues until all genes are classified. The execution of PatternClus includes the following steps:

a) Start with an arbitrary unclassified gene \( g_i \) and find its regulation pattern.

b) From the remaining unclassified genes, find the maximal matching regulation pattern w.r.t. \( g_i \) for a given \( \delta \).

c) Classify all those genes having the same maximal matching regulation pattern with the same cluster_id as that of \( g_i \).

d) Repeat steps a to d till all the genes are classified.

e) Find sub-clusters.

For a cluster \( C_i \), the MMRP \( \psi' \) of all genes \( \in C_i \) is same which is also the MMRP of \( C_i \), i.e., \( \psi'_{C_i} \). The process of finding sub-clusters is detailed below.

#### 3.2.1 Finding the sub-clusters

For sub-cluster identification the rank of the genes in a cluster are considered as follows:
a. For each cluster \(C_i\) find \(\psi_{g_i}^t\).

b. Rank each \(g_i \in C_i\) in ascending order according to the expression values where \(\psi_{g_i,t}^t \neq x\).

c. Sub-clusters are formed with those genes \(g_k \in C_i\) \(k = 1, 2, \ldots \|C_i\|\) having the same rank.

Assuming \(G\) be the set of genes and \(C\) be the set of clusters, the following lemmas and observations are trivial.

Lemma 1: \(\forall g_i, g_j \in G\) where \(g_i \in C_1\) and \(g_j \in C_2\), \(\psi_{g_i}^t \neq \psi_{g_j}^t\).

Lemma 2: If \(g_i\) is a noise gene, then \(g_i \notin C\) where \(C\) is the set of all clusters.

Observation 1: Suppose a cluster \(C_{Gl}\) contains the set of genes \(Gl\), \(Gl \subset G\) and \(Gl\) match in the conditions \(t, t \subset (T - 1)\) and \(t \geq 2\). \(C_{Gl}\) is a cluster if the regulation pattern \(\phi\) of genes in \(Gl\) across \(t\) are same.

Observation 2: For the cluster \(C_{Gl}\) containing genes \(Gl\) and conditions \(t\), if there is an ordering (permutation) of \(t\) such that the expression values of all genes in \(Gl\) have the same rank, then the genes are all part of cluster \(C_{Gl}\) else the cluster \(C_{Gl}\) contains sub-clusters \(S_{C_{Gl_1}}, S_{C_{Gl_2}}, \ldots S_{C_{Gl_k}}\) where \(\|S_{C_{Gl_i}}\| \geq 2, i = 1, 2, \ldots k\) and rank of all genes within a sub-cluster are same (according to definition 5).

Once cluster identification process has terminated, each cluster can be represented by a cluster profile which will be given by the MMRP of that cluster.

4 Performance evaluation

**PatternClus** was implemented in Java in Windows environment running on a HP workstation. The method was tested with the two datasets as given below:

- **Dataset 1**: In the data from DeRisi et al. (1997), the authors used DNA microarrays to study the temporal gene expression of 6,400 genes in *Saccharomyces cerevisiae* during the metabolic shift from fermentation to respiration. Expression levels were measured at seven time points during the diauxic shift. The full dataset can be downloaded from the Gene Expression Omnibus website, (http://www.ncbi.nlm.nih.gov/geo/query).

- **Dataset 2**: In Cho et al. (1998), they used the temperature sensitive mutant strain CDC28-13 to produce a synchronised cell culture of the *Saccharomyces cerevisiae* from which 17 samples were taken at ten minute intervals and hybridised to Affymetrix chips. The final data was downloaded from http://scdp.stanford.edu/yeast_cell_cycle/full_data.html. Cho’s dataset is widely available and has functional classification that allows validation of clustering results. This dataset contains 6,218 genes at 17 time points.

- **Dataset 3**: The dataset used is from the study of Wen et al. (1998) where the authors study the relationship among gene expression patterns of genes involved in the rat central nervous system (CNS) measured during the development of the rat’s CNS. Gene expression patterns for 112 genes were measured at nine different developmental time points. This yields a 112 \(\times\) 9 matrix of gene expression data.
4.1 Results

We compared our algorithm with that of the K-means (Tavazoie et al., 1999) and hierarchical clustering (UPGMA) (Eisen et al., 1998) algorithms. Some of the clusters obtained by our method on the full Dataset 1 is shown in Figure 4. The parameter $\delta$ is used as a tuning parameter for maintaining the cluster quality; higher values of $\delta$ give more finer cluster results. From our experiments we have found that the probable range of $\delta = \left[ \frac{3}{4} T, T \right]$ for the datasets used in this paper. However, this range has been found to work for some other datasets also.

Figure 4 Clusters obtained by the proposed method on Dataset 1 (see online version for colours)

We also experimented our method over a reduced form of Dataset 1. The dataset was reduced by filtering out the low variance and low entropy genes from the data. Figures 5 and 6 show some of the clusters obtained by k-means and hierarchical algorithm on the reduced dataset. We note here that the clusters obtained by our algorithm are detected automatically and unlike k-means no input parameter for number of clusters is needed. We have tested k-means with $k = 16, 20, 30, 40$. Since our method gave a total of 48 clusters for the reduced dataset, we also tested k-means algorithm for $k = 48$. Similarly, UPGMA algorithm was tested for various values of cutoff and also for cutoff = 48. The sub-clusters obtained from the reduced form of Dataset 1 for some of the clusters of Figure 4 are shown in Figure 5. The sub-cluster formation is based on the rank of expression values in the subset of conditions that match according to MMRP. Due to space constraint we could not show all the clusters and their sub-clusters. Some of the clusters obtained from Dataset 2 using PatternClus is shown in Figure 6. As can be seen from the figure, the sub-clusters gives the embedded clusters in the clusters. This helps in finding the finer clustering of a dataset or when it is important to know if the similarly expressed genes undergo some distinct stages in the experiment. Figure 7
A pattern matching approach for clustering gene expression data

shows a sample output of some clusters of Dataset 1 with genes inserted incrementally. The inserted genes are shown in red colour (grey for black and white images) with filled circles at the time points.

Finally, to validate the cluster results a cluster validity measure like z-score was used and the results were compared with the different clustering algorithms.

Figure 5 Clusters obtained by kmeans for k = 48 (see online version for colours)

4.2 Cluster quality

To assess the quality of our method, we need an objective external criterion. A statistical rating of the relative gene-expression activity in each cluster and GO term has been done. In order to validate our clustering result, we employ z-score (Gibbons and Roth, 2002) as the measure of agreement. A higher value of z indicates that genes would be better clustered by function, indicating a more biologically relevant clustering result. z-score (Gibbons and Roth, 2002) is calculated by investigating the relation between a clustering result and the functional annotation of the genes in the cluster. We have used Gibbons ClusterJudge (Gibbons and Roth, 2002) tool to calculate the z-score.
To test the performance of the clustering algorithm, we compare clusters identified by \textit{PatternClus} with the results from k-means and UPGMA using z-score. In this paper, the reported z-score was averaged over ten repeated experiments. The result of applying the z-score on the full \textit{Dataset 1} is shown in Table 1. In this table, the proposed algorithm has been compared with the well-known agglomerative hierarchical algorithm, UPGMA. Table 1 clearly shows that \textit{PatternClus} outperforms UPGMA w.r.t. the cluster quality. The z-score values obtained from clustering the reduced \textit{Dataset 1} is given in Table 2. As can be seen in the table, our method performs better than K-means and hierarchical clustering. We note here that unlike k-means our method does not require the number of clusters as an input parameter. It detects the clusters present in the dataset automatically and gives the rest as noise. Also, UPGMA requires the parameter \textit{cutoff} as input to the algorithm.

\textbf{Figure 6} Clusters obtained by UPGMA using average linkage and at cutoff = 48 (see online version for colours)

In the incremental version of \textit{PatternClus}, the new genes are compared with the cluster profiles to get the clustering of the updated database, the whole database need not be used. This considerably reduces the time required during cluster assignment.
Figure 7 Some of the cluster results for the reduced form of Dataset 1 (see online version for colours)

<table>
<thead>
<tr>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Cluster 3</th>
<th>Cluster 4</th>
<th>Cluster 5</th>
</tr>
</thead>
</table>

Figure 8 Sub-clusters: (a)-(d) for Cluster 1, (e)-(f) for Cluster 2, (g)-(h) for Cluster 3, (i)-(j) for Cluster 4 and (k)-(l) for Cluster 5 of Figure 4 (see online version for colours)

<table>
<thead>
<tr>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e)</td>
<td>(f)</td>
<td>(g)</td>
<td>(h)</td>
</tr>
<tr>
<td>(i)</td>
<td>(j)</td>
<td>(k)</td>
<td>(l)</td>
</tr>
</tbody>
</table>

5 Incremental version of PatternClus

PatternClus can be used for static gene expression data, however, due to the huge amount of microarray experiments whenever there are new gene expression data coming in, it is highly desirable to perform the updates (the clustering) of these newly arrived genes incrementally. PatternClus was modified to handle this situation incrementally. Here, the MMRP of the newly inserted gene was computed with each of the cluster profiles. The gene can be inserted to the cluster with whose cluster profile it is maximally matched. It has been found that the incremental algorithm yields the same result as PatternClus.
R. Das et al.

Figure 9  Some of the clusters obtained by *PatternClus* on Dataset 2 (see online version for colours)

![PatternClus clusters](image)

Figure 10  Incremental clustering: Some cluster results over Dataset 1 (see online version for colours)

![Incremental clustering](image)

Table 1  *z*-scores for *PatternClus* and hierarchical clustering using average linkage at cutoff = 162 and 156

<table>
<thead>
<tr>
<th>Algorithm used</th>
<th>No. of clusters</th>
<th><em>z</em>-score</th>
<th>Total no. of genes</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>PatternClus</em></td>
<td>156</td>
<td>10.9</td>
<td>6,089</td>
</tr>
<tr>
<td>UPGMA</td>
<td>162</td>
<td>5.76</td>
<td>6,089</td>
</tr>
<tr>
<td>UPGMA</td>
<td>156</td>
<td>5.99</td>
<td>6,089</td>
</tr>
<tr>
<td>kmeans</td>
<td>156</td>
<td>NA</td>
<td>6,089</td>
</tr>
</tbody>
</table>
Table 2  z-scores for PatternClus, k-means at k = 16 and 48 and hierarchical clustering using
average linkage at cutoff = 16 and 48

<table>
<thead>
<tr>
<th>Algorithm used</th>
<th>No. of clusters</th>
<th>z-score</th>
<th>Total no. of genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatternClus</td>
<td>48</td>
<td>7.02</td>
<td>614</td>
</tr>
<tr>
<td>k-means</td>
<td>16</td>
<td>−0.366</td>
<td>614</td>
</tr>
<tr>
<td>k-means</td>
<td>48</td>
<td>0.563</td>
<td>614</td>
</tr>
<tr>
<td>UPGMA</td>
<td>16</td>
<td>0.285</td>
<td>614</td>
</tr>
<tr>
<td>UPGMA</td>
<td>48</td>
<td>0.309</td>
<td>614</td>
</tr>
</tbody>
</table>

6 Conclusions

This paper presents a pattern matching approach for clustering gene expression data. The clusters obtained have been found satisfactory on z-score as well as visual inspection for three real datasets. A major attraction of PatternClus is being free from the initial guess about the number of clusters and non-dependency of input parameters. The regulation-based cluster expansion also overcomes the computational overhead usually linked with the other clustering approaches. The incremental version handles new gene data incrementally.

References


A pattern matching approach for clustering gene expression data


