

A Frequent Itemset–Nearest Neighbor Based Approach for Clustering Gene Expression Data

Rosy Das^{*†}, D. K. Bhattacharyya^{*} and J. K. Kalita[†]

^{*} Department of Computer Science and Engineering

Tezpur University, Tezpur, India

[†]Department of Computer Science,

University of Colorado at Colorado Springs, Colorado

[‡] Contact author: rosy8@tezu.ernet.in

Abstract—Microarray technology has enabled the monitoring of expression levels of thousands of genes across different experimental conditions. Identifying groups of genes that manifest similar expression patterns in such huge amounts of data is crucial in the analysis of gene expression time series. In this study, we present an integrated analysis of microarray data using association mining and clustering that discovers intrinsic grouping based on co-occurrence patterns in such data. A shared nearest neighbor approach is used to cluster the results of association mining to obtain the final clustering of the dataset. The method was used with real-life datasets and has been found to perform satisfactorily.

Index Terms—Gene expression, microarray, coherent pattern, association mining, clustering.

I. INTRODUCTION

A microarray experiment assesses a large number of DNA sequences under multiple conditions such as a time series during a biological process or a collection of different tissue samples at different time points. A gene expression dataset from a microarray experiment can be represented by an expression table, $D = \{a_{ij} \mid i = 1, \dots, p; j = 1, \dots, n\}$, where $a_{ij} \in R$ is the measured expression level of gene g_i in sample s_j . Each row in the expression table corresponds to a particular gene and each column to a particular sample or condition.

A. Similarity Measures

To identify genes or samples that have similar expression profiles, an appropriate similarity measure is required. Some commonly used distance metrics are Euclidean distance, Pearson’s correlation coefficient, Jackknife correlation and Spearman’s rank-order correlation coefficient [1]. In general, Euclidean distance and Pearson’s correlation coefficient are widely used as distance measures for clustering gene expression data [2]. However, Euclidean distance measure is not effective in reflecting functional similarities as well as interdependence among values. It can only account for closeness in values. Pearson’s correlation coefficient accounts for the overall shapes of genes, but is not robust to outliers. The authors have proposed a dissimilarity measure in [3] which handles the above mentioned problems.

B. Gene Expression Data Clustering Approaches

Data mining techniques have proven useful in understanding gene function, gene regulation, cellular processes and subtypes of cells. According to [1], most data mining algorithms developed for gene expression time series data deal with the problem of clustering. In the gene expression context, clustering is the process of identifying subsets of genes that behave similarly along a course of time under described test conditions. Genes in the same cluster have similar expression patterns. Two major challenges for clustering gene expression data are:

- 1) How to group genes with similar expression patterns (co-expressed genes) together, and
- 2) How to extract useful patterns intelligently from noisy datasets.

Gene expression data clustering techniques can be categorized as follows: *partitioning*, *hierarchical*, *density based*, *model based* or *graph based*.

Partitioning algorithms such as K-means [1] and Self Organizing Maps (SOMs) [4] have been widely used in gene expression datasets. However, the specification of the number of clusters in advance is a must in these approaches, which is difficult to pre-estimate in case of gene expression data.

Hierarchical clustering algorithms are divided into agglomerative and divisive approaches. Unweighted Pair Group Method with Arithmetic Mean (UPGMA), presented by [5], adopted an agglomerative method while in [6] the genes were split through a divisive approach called the deterministic-annealing algorithm (DAA). Hierarchical clustering provides a natural way to graphically represent the dataset. However, it suffers from high computational complexity and a small change in the dataset may greatly change the hierarchical dendrogram structure.

Density based clustering identifies dense areas in the object space. Clusters are hypothesized as high density areas separated by sparsely dense areas. In [7], the Density-based Hierarchical Clustering method (DHC) was proposed that uses a density-based approach to identify co-expressed gene groups from gene expression data. An alternative to direct similarity is to define the similarity of points in terms of their shared nearest neighbors. This idea was first introduced by Jarvis

and Patrick [8]. Density-based approach discovers clusters of arbitrary shapes even in presence of noise. However, density-based clustering techniques suffer from high computational complexity with increase in dimensionality even if spatial index structure is used, and input parameter dependency.

Model based approaches provide a statistical framework to model the cluster structure of gene expression data. The Expectation-Maximization (EM) algorithm [9] can handle various shapes of data and can be very expensive since large number of iterations may be required for the iterative refinement of the parameters. Some other model-based algorithms to cluster gene expression datasets are [10] which is based on Hidden Markov Model (HMM) and [11] based on statistical models where each cluster is represented by a spline curve and the clustering is computed using an EM-type algorithm. The model-based approach provides an estimated probability that a data object will belong to a particular cluster. Since gene expression data are “highly connected”, there may be instances in which a single gene has high correlation with two totally different clusters. Thus, the probabilistic feature of model-based clustering is much suitable for such datasets. However, the model-based approach assumes that the dataset fits a specific distribution which is not always true.

Graph theoretic techniques [12], [13], [14] are another useful category of clustering techniques used in the bioinformatics domain. In graph-based clustering algorithms, graphs are built as combinations of objects, features, or both as the nodes and edges. The graph is partitioned by using graph theoretic algorithms. CLuster Identification via Connectivity Kernels (CLICK) ([13]) is suitable for subspace and high dimensional data clustering and is extensively used to cluster gene expression data. The Cluster Affinity Search Techniques (CAST) by [12] does not require a user-defined number of clusters and handles outliers efficiently. But, it faces difficulty in determining a good threshold value. E-CAST [14] is an enhanced version of CAST which uses a dynamic threshold.

C. Association Rule Mining

Association rules were introduced in [15]. Association rules follow the form $X \Rightarrow Y$ where X and Y are disjoint sets of items (or itemsets). X is called the antecedent and Y the consequent of the rule. The intended meaning of such a rule is that data instances that contain X are likely to contain Y as well. The extent to which the rule applies to a given dataset can be measured using various metrics including support and confidence. The support of the rule is the probability of X and Y occurring together in an instance, $P(X \text{ and } Y)$. The confidence of the rule is the conditional probability of Y given X , $P(Y | X)$. Here, probability is taken to be the observed frequency in the underlying dataset. The Apriori algorithm [15] is a pioneering algorithm for association rule mining; it finds all frequent itemsets whose supports are above a threshold. The FP-tree [16] does not rely on candidate generation step and is therefore faster than the Apriori algorithm. Recently, association rules has been proposed to the analysis of gene expression data [17], [18], [19] in order to extract

associations and relationships among subsets of genes. This approach avoids some of the existing drawbacks of standard clustering algorithms and has been proved to be successful in extracting new and informative gene relationships. A major disadvantage of association rules discovery method is the large amount of rules that are generated, which becomes a major problem in many applications. In several studies, some post-processing pruning methods have been proposed to reduce the number of generated rules. For example, in the context of gene expression, Creighton and Hanash imposed constraints on the size of the rules, extracting only those formed by seven or more genes [17] while Tuzhilin and Adomavicius proposed several post-processing operators for selecting and exploring interesting rules from the whole set [18]. In [19], a method for the integrative analysis of microarray data based on the Association Rules Discovery data mining technique is presented. The approach integrates gene annotations and expression data to discover intrinsic associations among both data sources based on co-occurrence patterns. Filter options have been used to eliminate irrelevant and redundant associations. This option drastically reduces the number of associations to be examined. In [20], a new similarity measure has been proposed that can be applied together with hierarchical clustering and leads to grouped similar patterns. The mining part first constructs a compact data structure called Gene Profile tree (or GP-tree), from which the frequent co-regulated gene profiles are extracted.

D. Motivation

In this paper, we introduce a finer clustering method that integrates a traditional clustering technique with frequent itemset discovery. The gene expression dataset is encoded in binary with respect to correlated genes. Frequent itemset mining is then run on this data to discover the maximal frequent set(s). This maximal frequent set gives the core genes in a cluster. Cluster expansion then proceeds with this set of core genes using a shared neighbor approach. Frequent itemset mining has been applied to gene expression data. However, to the best of our knowledge, the approach reported in this paper has not yet been explored in the domain of gene expression datasets. The advantage of our method is that it produces finer clustering of the dataset. The advantage of using frequent itemset discovery is that it can capture relations among more than two genes while normal similarity measures can calculate the proximity between only two genes at a time.

II. A NEW METHOD INTEGRATING ASSOCIATION RULE MINING AND NEAREST NEIGHBOR CLUSTERING

Our innovative method works in three phases. In the first phase, the gene expression data D is transformed into a 0-1 transaction matrix. The second phase finds the maximal frequent itemset using a frequent itemset mining algorithm such as Apriori or FP-tree. The third phase is dedicated to the task of clustering using a shared nearest neighbor based approach. Below, we discuss these phases in detail.

A. Phase I: Transformation from gene expression matrix to Transaction matrix

The gene expression dataset D is a $p \times n$ matrix of expression values where p is the number of rows (genes) and n is the number of columns (time points) as shown in Equation 1. Pearson's correlation coefficients between the genes across time series is used to build a $p \times p$ similarity matrix for the whole dataset D . The dissimilarity measure given in [3] may also be used to generate the similarity matrix. We introduce some definitions as we proceed with the description of our method.

Definition 1: Nearest Neighbor of a gene

A gene g_i is the nearest neighbor of a gene g_j if the similarity between genes g_i and g_j is greater than or equal to some similarity threshold. $\rho(g_i, g_j) \geq \theta$, where θ is a similarity threshold and ρ is the pearson's correlation.

From the nearest neighbor lists, we build the $p \times p$ gene-gene transaction matrix, T , of zeroes and ones (Equation 2). For each gene g_i , a p -pattern of 0's and 1's is obtained with 1 if a gene g_j is neighbor of g_i and 0 otherwise as given in (Equation 3).

$$D = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pn} \end{bmatrix} \quad (1)$$

$$T = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1p} \\ t_{21} & t_{22} & \cdots & t_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ t_{p1} & t_{p2} & \cdots & t_{pp} \end{bmatrix} \quad (2)$$

$$T = t_{ij} = \begin{cases} 1 & \text{if } \rho(g_i, g_j) \geq \theta, \text{ where } i = 1, 2, \dots, p; \\ & j = 1, 2, \dots, p. \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Pruning: Those transactions are pruned to satisfy the following conditions:

- 1) In the transaction matrix, the value of t_{ij} , where $i = j$ is set to zero since the same gene does not contribute to frequent itemset generation.
- 2) In this transaction matrix if for a particular row i the value of t_{ij} across all j conditions are zero and the same applies for column j and all i rows, then that i^{th} row and j^{th} column both are discarded.

These two steps reduce the size of the transaction matrix considerably.

Phase II now uses this matrix, T , to calculate the frequent itemset using FP-tree.

B. Phase II: Maximal Frequent Itemset Generation

In this phase, we use the FP-tree to generate the maximal frequent itemset(s) (MFIS) at support threshold $s\%$. The gene-gene $p \times p$ transaction matrix, T is fed as input along with the user defined support threshold to get the frequent itemsets.

The maximal frequent itemset obtained from this phase gives us the set of core genes. The identification of core genes is done as follows.

- If only one MFIS is obtained at $s\%$ support, the genes within that set become the set of core genes for a particular cluster.
- If more than one MFIS is obtained at $s\%$ support and there is a chain of genes (items) from one MFIS to the other, the genes are merged together into the set of core genes for a particular cluster.
- If more than one MFIS is obtained at $s\%$ support and there is no chain of genes (items) from one MFIS to the other, each of the MFIS will give the set of core genes for different clusters.

This set of core genes are the seeds for cluster expansion which gives the finer clustering of the dataset. Different clustering approaches such as hierarchical or density based clustering can be applied on these core genes to get the final cluster. The next phase gives a detailed overview of the clustering process.

The following definitions provide the foundation for the clustering process.

Definition 2: Density of a gene

The density of a gene g_i is the number of nearest neighbors of that gene.

$$\text{Density}(g_i) = \sum_{j=1}^p t_{ij}, \text{ where } t_{ij} = 1 \quad (4)$$

Definition 3: Core genes

A set of genes $Cr = \{g_a, \dots, g_m\}$ is termed as core genes where $Cr = \{\cup MFIS\}$ is the maximal frequent itemset generated by the FP-tree algorithm.

Definition 4: Shared Neighbors of a gene

Suppose $Cr = \{g_a, \dots, g_m\}$ be the set of core genes. A gene g_k is said to be the shared neighbor of each of the genes in Cr if it satisfies the following:

$$\text{SharedNeighbor}, sn(Cr, g_k) = \begin{cases} \rho(g_a, g_k) \geq \beta \wedge \rho(g_b, g_k) \\ \geq \beta \wedge \cdots \wedge \rho(g_m, g_k) \geq \beta \end{cases} \quad (5)$$

where β is the shared neighbor threshold.

Definition 5: Noise genes

A gene g_k is said to be a noise gene, if it has no nearest neighbor gene g_m , where $g_m \in D$.

Lemma 1: Seeds selected for cluster expansion cannot be noise.

Proof: Assume that g_j is a noise gene and $g_j \in Cr$, i.e. the set of core genes. Now, since $g_j \in Cr$, it should satisfy $\rho(g_j, g_k) \geq \beta$ for any non-core gene $g_k \in D$. Thus, g_k is a shared neighbor of g_j and $\rho(g_j, g_k) \geq \theta$. But, according to Definition 5, it contradicts the initial assumption i.e. $g_j \notin \text{Set of Noise genes}$.

Therefore, a noise gene cannot be a seed for cluster expansion.

C. Phase III: Clustering

We have used a shared neighbor approach to expand the cluster from the finer clustering to obtain the final cluster. The clustering procedure is initiated from the core genes identified in Phase II. First, these genes are classified. From the unclassified genes, a gene g_k is selected which is nearest neighbor of each of the core genes ($Cr = g_a, g_b, \dots, g_m$). This can be found using the transaction matrix, T as follows: if $t_{ij} = 1$, for $i = k$ and $j = a, b, \dots, m$, then g_k is nearest neighbor to each of the core genes. This process eliminates the exhaustive neighbor search which most of the clustering algorithms use and is also computationally expensive. If g_k has similarities greater than a given *shared neighbor threshold* (β) with each of the core genes, g_k is classified with the same cluster ID as that of the core genes. This means that the core genes has g_k as their shared neighbor. Since g_k is shared neighbor of the core genes, we merge g_k into the same cluster along with the core genes. This process of cluster expansion is iterated until there are no more genes that can be merged into this cluster. The cluster thus obtained gives a final cluster. Once cluster expansion terminates, the row and column of the classified genes in the transaction matrix T are discarded from further consideration. This step reduces the number of items (genes) which have to be checked for itemset generation. The process then restarts phase II with the new compact transaction matrix T .

The steps in our method are given below.

- 1) Calculate the $p \times p$ similarity matrix using Pearson's correlation and generate the $p \times p$ gene-gene transaction matrix.
- 2) Generate the maximal frequent itemset using FP-tree algorithm.
- 3) Classify the maximal frequent itemset as core genes and give the same cluster_id to them.
- 4) Select a gene from the nearest neighbors of the core genes which is a shared neighbor of each of the core genes and classify this gene with the same cluster_id.
- 5) Repeat step 4 till no more genes satisfy the shared neighbor condition and increment the cluster_id.
- 6) Discard the rows and columns of the classified genes from the gene-gene transaction matrix and go to step 2.

III. RESULTS

Our method has been implemented in C++ in Linux platform and run on Pentium IV machine with a 256MB RAM and 1.6 GHz speed. The method was tested on the following two real datasets.

- 1) Dataset 1: The dataset used is from the study of [21] 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 112×9 matrix of gene expression data.

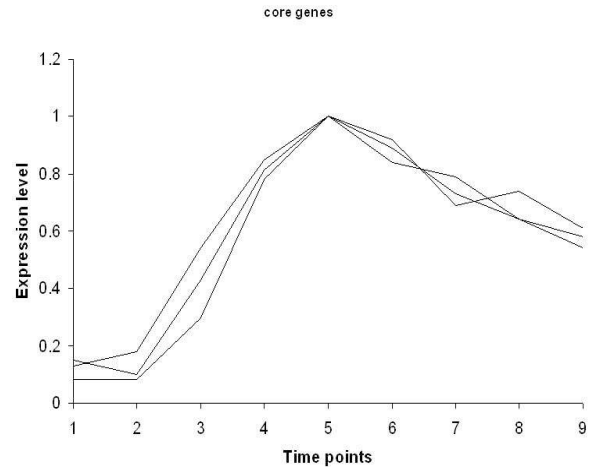


Fig. 1. The Core genes at $s=40\%$

- 2) Dataset 2: [22] use the temperature sensitive mutant strain CDC28-13 to produce a synchronized cell culture of the *Saccharomyces cerevisiae* from which 17 samples were taken at 10 minute intervals and hybridized to Affymetrix chips. The final data was downloaded from http://yscdp.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 6218 genes at 17 time points.

A. Standardization of the datasets

The dataset 1 is first normalized to have mean 0 and standard deviation 1. No other normalization was performed on this dataset. However, since the dataset 2 contains negative expression values, we shift the scale by +610. Expression levels were then normalized to have mean = 0 and variance = 1. Genes that have small variance over time were filtered out. There are many unimportant gene profiles in dataset 2 which make it difficult to extract the real cluster structures. Therefore a gene selection procedure as given in [23] is used to eliminate the irrelevant and redundant genes. This process reduces the gene set to 800 genes.

B. Experimental Results

We exhaustively tested our method on both the datasets. In dataset 1, the similarity matrix is first computed and the transaction matrix is obtained from it. The method is then executed and eight clusters are obtained. The agreement of the clusters with prior biological knowledge is quite good. Some of the clusters obtained are shown in Figure 2 and 4 and their respective core genes are shown in Figure 1 and 3. From these results, we can also conclude that the core genes gives the overall trend of the cluster. Therefore, this approach can also be used to detect the embedded clusters in the dataset. When the method was executed on dataset 2, the clusters obtained agree well with the functional classification of [22]. Of the different clusters obtained from dataset 2, two of them are shown in this paper. The first cluster along with its core

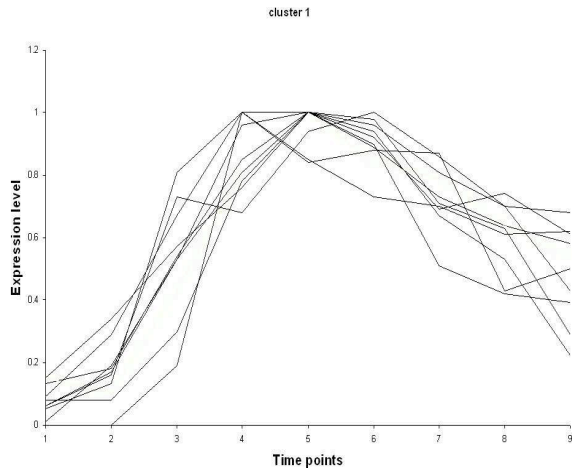


Fig. 2. The final cluster 1 obtained from the core genes

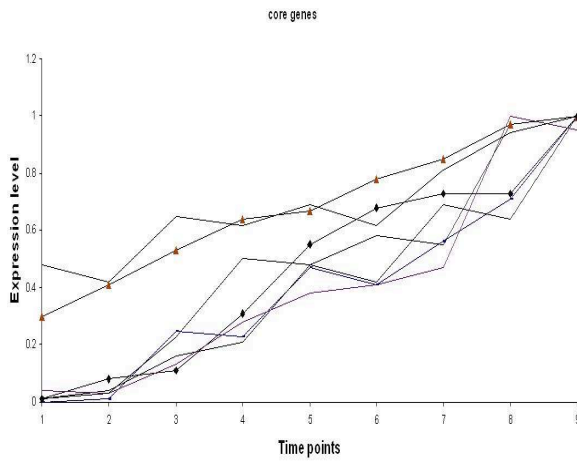


Fig. 3. The Core genes at $s=40\%$

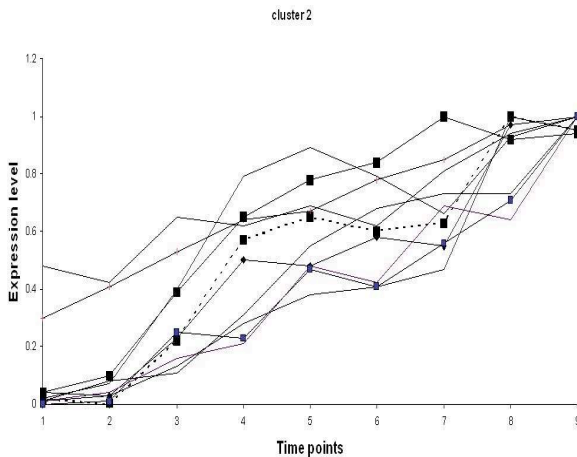


Fig. 4. The final cluster 2 obtained from the core genes

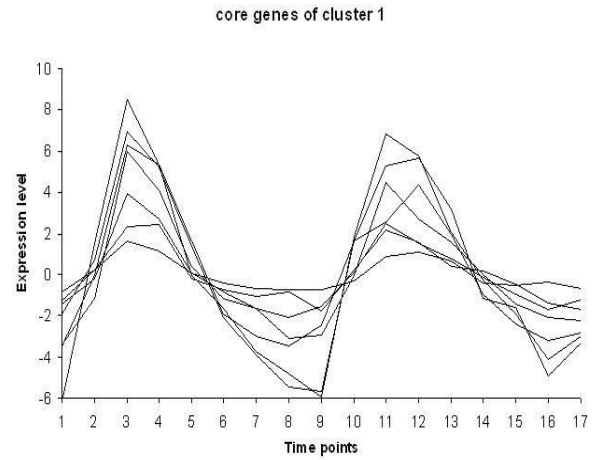


Fig. 5. The core genes of cluster 1

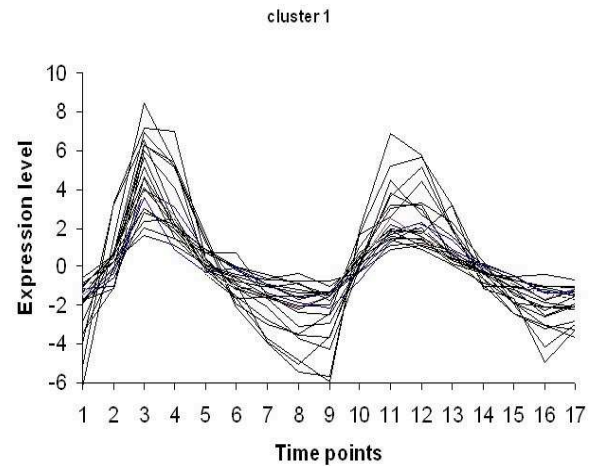


Fig. 6. Final cluster 1 based on the core genes of Figure 5

genes is shown in Figure 5 and Figure 6. The second cluster result is shown in Figure 7 and Figure 8.

C. Discussion

From our exhaustive experiments, it is seen that by varying the value of β , the quality of the clusters can be increased further. The support count in the frequent itemset generation has a pivotal role in the detection of the core genes. With the increase in the support count, more compact set of core genes can be obtained. Moreover, for high values of support count, frequent itemset generation also becomes faster. Taking these factors into count, more compact clusters may be obtained. To test the performance of the clustering algorithm, we compared the clusters obtained by our method with the ‘ground truth’, and the result was found satisfactory.

IV. CONCLUSION

This paper presents a method for clustering gene-expression time series data. The frequent itemset generation step gives the innermost or the fine clusters from the gene expression data and the shared neighbor clustering approach gives the

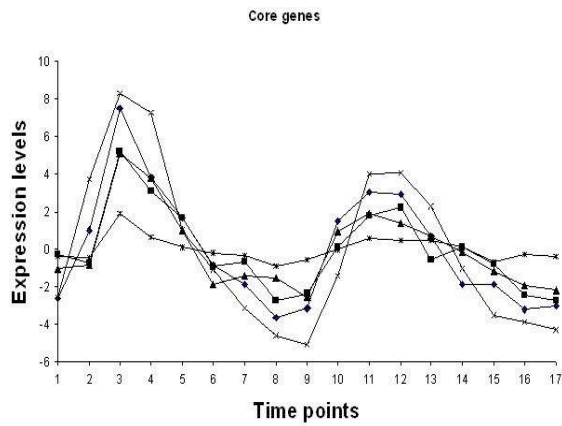


Fig. 7. The core genes of cluster 2

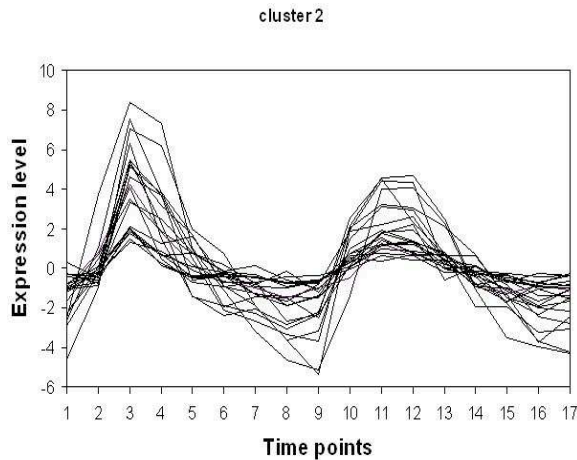


Fig. 8. The final cluster 2 based on the core genes of Figure 7

final clusters in the dataset. Compared with other clustering approaches, our method is better capable of identifying finer clusters of the dataset and may also be used to detect embedded clusters; this work is ongoing. Our experimental results have shown that the clusters obtained are similar to those obtained by [4] and [12]. In fact, on increasing the value of β , the quality of clusters improves.

However, work is ongoing to test the robustness of the method in terms of other gene expression datasets.

ACKNOWLEDGMENT

We thank Roland Somogyi, President of Biosystemix Ltd., for providing us with the Rat spinal cord development dataset.

REFERENCES

[1] Stekel, D. (2003). *Microarray Bioinformatics*, Cambridge University Press, Cambridge, UK.

[2] Jiang, D., Tang, C. and Zhang, A. (2003). 'Cluster Analysis for Gene Expression Data: A Survey', URL: www.cse.buffalo.edu/DBGROUP/bioinformatics/papers/survey.pdf.

[3] Das, R., Kalita, J.K. and Bhattacharyya, D.K. (2007). 'An effective dissimilarity measure for clustering gene expression time series data', *BIOT 2007*, Colorado.

[4] Tamayo, P., Slonim, D. et al. (1999) 'Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation', *Proc. of Natl. Acad. Sci, USA*, pp. 2907–2912.

[5] Eisen, M. B. et al. (1998). 'Cluster analysis and display of genome-wide expression patterns', *Proc. of Natl. Acad. Sci. USA*, 95, 14863–14868.

[6] Alon, U. et al. (1999). 'Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Array', *Proc. of Natl. Acad. Sci., USA*, Vol. 96, No. 12, pp.6745–6750.

[7] Jiang, D., Pei, J. and Zhang, A. (2003). 'DHC: A Density-based Hierarchical Clustering Method for Time Series Gene Expression Data', *Proc. of BIBLE2003: 3rd IEEE International Symposium on Bioinformatics and Bioengineering*, Bethesda, Maryland.

[8] Jarvis, R.A. and Patrick, E.A. (1973). 'Clustering using a similarity measure based on Shared Nearest Neighbors', *IEEE Transactions on Computers*, Vol.11

[9] Dempster, A. et al. (1977). 'Maximum likelihood from incomplete data via the EM algorithm', *Journal of the Royal Statistical Society*, pp. 1–38.

[10] Schliep, A. et al. (2003). 'Using hidden Markov models to analyze gene expression time course data', *Bioinformatics*, 19 (Suppl. 1), i255–i263.

[11] Bar-Joseph, Z. et al. (2002). 'A new approach to analyzing gene expression time series data', *Proc. of Sixth Annual International Conf. on Computational Biology*, pp. 39–48.

[12] Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999). 'Clustering gene expression patterns', *Journal of Computational Biology*, pp. 281–297.

[13] Sharan, R. and Shamir, R. (2000). 'CLICK: A clustering algorithm with applications to gene expression analysis', *Proc. of Eighth Int. Conf. on Intelligent Systems for Molecular Biology*, AAAI Press.

[14] Bellaachia, A. et al. (2002). 'E-CAST: A Data Mining Algorithm for Gene Expression Data', *BIOKDD02: Workshop on Data Mining in Bioinformatics (with SIGKDD02 Conference)*, pp. 49.

[15] Agrawal, R., Imielinski T. and Swami, A. (1993). 'Mining Association Rules between Set of Items in Large Databases'. In *Proc. ACM SIGMOD Conference on Management of Data*, pp.207–216.

[16] Han, E.H., Pei, J. and Yin, J. (2000). 'Mining frequent patterns without candidate generation', in *Proceedings of SIGMOD 2000*.

[17] Creighton, C., Hanash, S. (2003). 'Mining gene expression databases for association rules'. *Bioinformatics*, vol 19, pp. 79–86.

[18] Tuzhilin, A., Adomavicius, G., (2002). 'Handling very large numbers of association rules in the analysis of microarray data', in *Proceedings of the Eighth ACM SIGKDD International Conference on Data Mining and Knowledge Discovery*, pp. 396–404.

[19] Carmona-Saez, P., Chagoyen, M., Rodriguez, A., Trelles, Oswaldo., Carazo, J. M. and Pascual-Montano, A. (2006). 'Integrated analysis of gene expression by association rules discovery', *BMC Bioinformatics*, vol.7: 54.

[20] Gyenesei, A., Wagner, U., Barkow-Oesterreicher, S., Stolte, E. and Schlapbach, R., (2007). 'Mining co-regulated gene profiles for the detection of functional associations in gene expression data', *Bioinformatics*, Vol. 23 no. 15, pp. 1927–1935.

[21] Wen, X., Fuhrman, S., Michaels, G.S., Carr, D.B., Smith, S., Barker, J.L. and Somogyi, R. (1998). 'Large-scale temporal gene expression mapping of central nervous system development' *PNAS*, 95(1), pp. 334–339.

[22] Cho, R. J., Campbell, M., Winzler, E., et al. (1998). 'A genome-wide transcriptional analysis of the mitotic cell cycle', *Mol. Cell*, Vol. 2(1), pp. 65–73.

[23] Huang, D., Chow, W., Ma, E. and Li, J. (2005). 'Efficient selection of Discriminant genes from microarray gene expression data for cancer diagnosis', *IEEE Transactions on Circuits and Systems*, Vol. 52, No. 9, pp. 1909–1918.