Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computational Biology and Chemistry 27 (2003) 565–574
Computational Biology and Chemistry
www.elsevier.com/locate/compbiolchem

Software note

# Rival penalized competitive learning (RPCL): a topology-determining algorithm for analyzing gene expression data

T. Murlidharan Nair [a,*], Christina L. Zheng [a], J. Lynn Fink [a], Robert O. Stuart [b], Michael Gribskov [a]

[a] *San Diego Supercomputer Center, University of California at San Diego, 9500 Gilman Dr., La Jolla, CA 92093-0537, USA*
[b] *Department of Medicine and Pediatrics, Division of Nephrology-Hypertension, Cancer Center, University of California at San Diego, La Jolla, CA 92093, USA*

Received 18 September 2003; received in revised form 20 September 2003; accepted 20 September 2003

**Abstract**

DNA arrays have become the immediate choice in the analysis of large-scale expression measurements. Understanding the expression pattern of genes provide functional information on newly identified genes by computational approaches. Gene expression pattern is an indicator of the state of the cell, and abnormal cellular states can be inferred by comparing expression profiles. Since co-regulated genes, and genes involved in a particular pathway, tend to show similar expression patterns, clustering expression patterns has become the natural method of choice to differentiate groups. However, most methods based on cluster analysis suffer from the usual problems (i) dead units, and (ii) the problem of determining the correct number of clusters ($k$) needed to classify the data. Selecting the $k$ has been an open problem of pattern recognition and statistics for decades. Since clustering reveals similar patterns present in the data, fixing this number strongly influences the quality of the result. While there is no theoretical solution to this problem, the number of clusters can be decided by a heuristic clustering algorithm called rival penalized competitive learning (RPCL). We present a novel implementation of RPCL that transforms the correct number of clusters problem to the tractable problem of clustering based on the degree of similarity. This is biologically significant since our implementation clusters functionally co-regulated genes and genes that present similar patterns of expression. This new approach reveals potential genes that are co-involved in a biological process. This implementation of the RPCL algorithm is useful in differentiating groups involved in concerted functional regulation and helps to progressively home into patterns, which are closely similar.


*Keywords:* RPCL; Clustering; Gene expression

## 1. Introduction

The development of large-scale genome sequencing and high throughput microarray technology have made possible the simultaneous temporal assay of thousands of genes expressed under different conditions (drug treatment, environment) (Thieffry, 1999; Bowtell, 1999; Lipshutz et al., 1999; Lockhart et al., 1996). The result of an array assay under a single condition comprises thousands of data points. With a minimum of two conditions needed for comparison, it is easy to see how the data can grow uncontrollably when multiple conditions and/or time points are involved (Tusher et al., 2001). Most functional approaches suggest that genes with similar expression patterns play concerted roles, such as being involved in the same pathway (DeRisi et al., 1997; DeRisi and Iyer, 1999). Thus, grouping genes with similar expression levels is useful towards understanding their functional significance (Granjeaud et al., 1999).

The grouping of genes with similar expression levels at different time points is the mathematical equivalent of clustering a set of points in $n$ dimensions. Of the clustering methods, agglomerative hierarchical clustering has been the most widely used method in array data analysis (Eisen et al., 1998; Spellman et al., 1998; Takahashi et al., 2001). However, despite the frequent use of hierarchical clustering (Jain and Dubes, 1988; Kaufman and Rousseeuw, 1990; Han and Kamber, 2001), this method has a number of shortcomings. For instance, its strict sense of hierarchical descent, which

* Corresponding author. Tel.: +1-858-8220887; fax: +1-858-8220873.
*E-mail address:* nair@sdsc.edu (T.M. Nair).

1476-9271/$ – see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiolchem.2003.09.006

is very well suited for the evolution of species is not as well suited for the complex expression groups found in microarray data. The problem is further exacerbated because the massive amount of microarray data is not efficiently handled by hierarchical clustering methods in terms of compute time or the algorithm. The algorithm involves the computation of a distance or similarity matrix, which has an $O(N^2)$ complexity. Hierarchical clustering has also been noted to have complications with robustness, non-uniqueness, and inversions resulting in difficulty in interpreting the true hierarchy (Morgan and Ray, 1995). Further, the deterministic nature of this method causes data points to be clustered locally, wherein, once a data point is assigned to a cluster, it cannot be re-evaluated to be in another cluster.

These shortcomings have lead many groups to focus on machine learning methods to cluster microarray data (Tamayo et al., 1999; Sultan et al., 2002; Takahashi et al., 2001). Unsupervised learning methods like Kohonen's self-organizing map (SOM) (Kohonen, 1982, 1990) have the robustness to handle the complex groups and the large amounts of noise associated with microarray data. SOMs are a more desirable choice for microarray data due to their efficiency and speed over hierarchical clustering as well as their ability to deal with outliers and non-parametric distributions. Toronen et al. showed the feasibility and sensitivity of using SOMs in analyzing and visualizing microarray data (Toronen et al., 1999). SOMs have also been successfully implemented in a diverse range of microarray experiments such as in detecting yeast cell cycle periodicity (Tamayo et al., 1999) and in time course studies of LPS-induced expression of genitourinary inflammation (Saban et al., 2001).

Even with the many positive features of SOM with respect to its use in microarray data analysis, one of the major drawbacks of SOM is the need for a pre-determined grid of nodes before running the algorithm. Most groups have tried to circumvent this problem with trial and error. Furthermore, SOMs and other conventional clustering algorithms like the *k*-means clustering technique which also suffers from the inability to automatically select the optimal number of clusters, also suffers from the dead unit or stuck vector problem. A dead unit is defined as a node or center that never finds any input vector close to it. This mainly stems from the fact that the centers are improperly initialized. Thus, in such a case, there exist a center $w_c$, which does not represent any input vector and thus takes the status of a dead unit. This problem has been overcome, to a large extent, by employing the strategy of a conscience factor (DeSieno, 1988). In this, the winning rate of the frequently winning node is decreased, thereby giving a chance to the non-winning units. Ahalt et al., has used the conscience factor approach in the frequency sensitive competitive learning (FSCL) method (Ahalt et al., 1990). While the FSCL formalism solves the dead unit problem, it still does not provide an approach for automatically selecting the optimal nodes required. However, Xu et al., has proposed an unsupervised learning paradigm, which during the process of learning, optimally determines the number of nodes (Xu et al., 1993). This strategy known as the rival penalized competitive learning (RPCL) is a modification of FSCL, and it automatically arrives at the optimal number of nodes needed to represent the clusters in the input space. The number of nodes needed to optimally cluster the input space is arrived at by determining the number of nodes as a function of the degree of similarity between nodes.

In our implementation of RPCL the degree of similarity is embedded into a user-defined parameter $\tau$. The novel approach that was used to embed the similarity condition in $\tau$ is biologically relevant since the members of each cluster will have to satisfy the similarity criterion, which is a reflection of the pattern of expression and may be associated with a particular biological process. Further, in the original implementation of RPCL, Xu et al. retained the unused nodes in anticipation of new patterns. In contrast our implementation prunes these unused nodes at the end of the analysis. The performance of the algorithm was evaluated using simulated data, microarray data from yeast cell cycle studies and from kidney developmental stage specific gene expression studies. The results show that RPCL is a robust approach and can be used to reveal biologically relevant information. Further the results also point to the fact that the algorithm is capable of differentiating closely similar pockets of expression pattern.

## 2. Materials and methods

### 2.1. Simulated data

Datasets in a two dimensional space with predetermined number of clusters were generated with variance 0.1 and 0.01, centered at (−1, 0), (1, 0), (0, 1), (0, −1) (Fig. 1). Points were assigned to each center according to a Gaussian distribution. Since the dataset was generated randomly, there is an equal probability of picking points centered on any one of the four centers. A total of 100 points were associated with each center. The clusters in the dataset with variance of 0.01 are more distinct than the dataset with variance 0.1. These would be referred to as tight and sparse clusters, respectively.

### 2.2. Yeast cell cycle data

Yeast cell cycle data was obtained from http://www.genomics.stanford.edu. Processing of the data was modeled after Tamayo et al. The genes that did not show a two-fold relative change or an absolute change of 35 were filtered out and the expression levels were normalized within each of the two cell cycles (Tamayo et al., 1999).

### 2.3. Kidney gene expression data

Gene expression studies during kidney development were studied earlier by Stuart et al. (2001). This dataset consists of gene expression levels for 880 genes during different

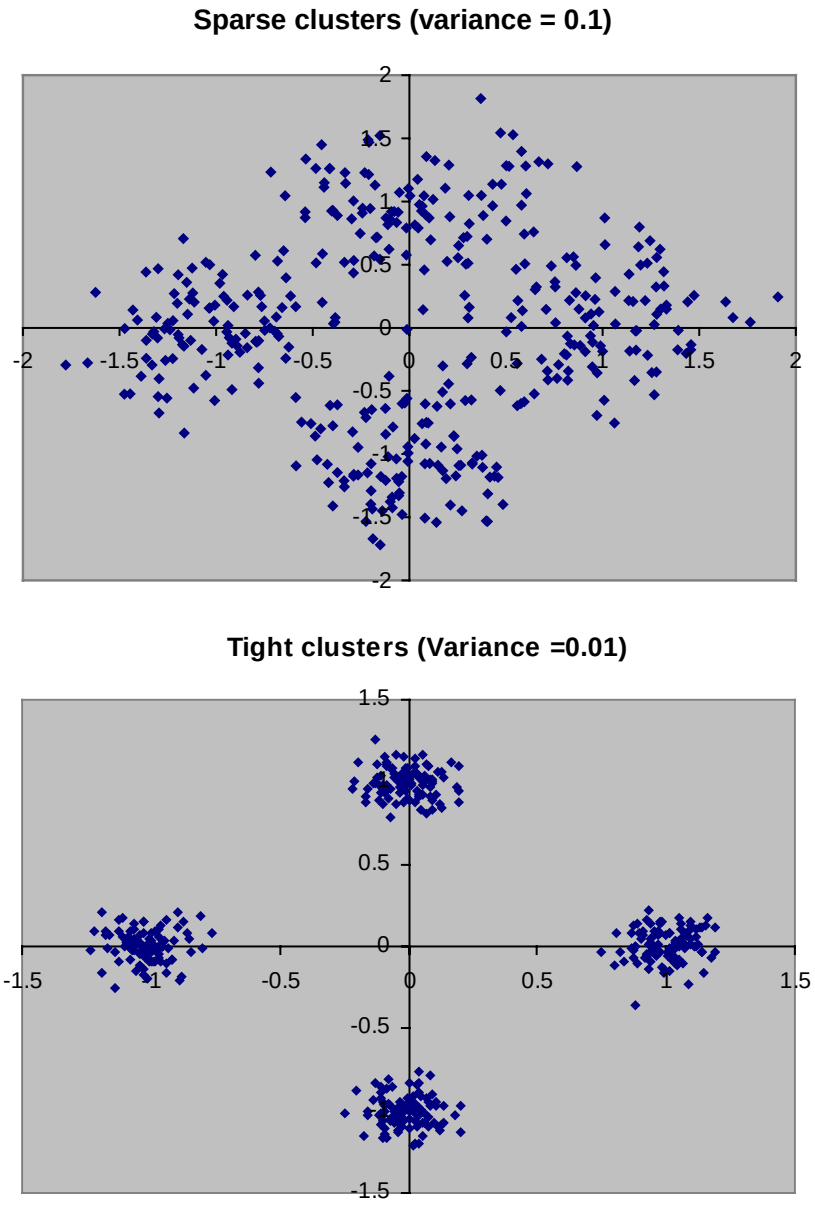


Fig. 1. Artificially generated dataset used to test the performance of the algorithm. The clusters are centered at (−1, 0), (0, 1), (1, 0) and (−1, 0).

stages of kidney development. The data was preprocessed by normalizing it to a mean of 0 and a variance of 1, before presenting it to the RPCL algorithm.

### 2.4. Rival penalized competitive learning algorithm

RPCL is an unsupervised learning strategy proposed by Xu et al. (1993, reviewed in Tambe et al., 1996), that automatically determines the optimal number of nodes. The principle underlying RPCL is that for each input vector not only is a winning node moved closer to adapt to the input but also a rival (second winner) is moved away or penalized. The rate at which the rival node is penalized is much smaller than the learning rate. The steps involved in the RPCL algorithm can be summarized as follows:

(i) For a given layer of nodes with outputs $k_i$, determine the winner and the rival node by randomly taking a sample $\vec{x}$ from the input space $I$. For each $i$ in $I$:

$$k_i = \begin{cases} 1 & \text{if } i = \text{c such that } \gamma_c||\vec{x}-\vec{w}_c||^2 \\ & \quad = \min_j(\gamma_j||\vec{x}-\vec{w}_j||^2) \\ -1 & \text{if } i = \text{r such that } \gamma_r||\vec{x}-\vec{w}_r||^2 \\ & \quad = \min_{j\neq c}(\gamma_j||\vec{x}-\vec{w}_j||^2) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\vec{w}_c$ is the weight vector, which wins the competition, $\vec{w}_r$ the weight vector of its rival, and $\gamma_j$ is the con-

science factor and is used to reduce the winning rate of the frequent winners. It is so called because a processing term that wins too often begins to "feel guilty" and prevents itself from winning excessively. It is useful to develop a set of equiprobable features or prototypes representing the input data. $\gamma$ is calculated as follows:

$$\gamma_j = \frac{n_j}{\sum_{i=1}^{k} n_i} \tag{2}$$

where $n_i$ refers to the cumulative number of times the node $i$ has won the competition.

(ii) Update the weight vector $w_i$ by

$$\Delta w_j(t+1) = w_j(t) + \Delta w_j \tag{3}$$

Here, $\Delta w_j$ is the magnitude by which the $j$th weight is adjusted when the $k$th input vector is applied to the network and is determined as follows:

$$\Delta w_j = \begin{cases} \alpha_c(\vec{x} - \vec{w}_i(t)) & \text{if } k_i = 1 \\ -\alpha_c(\vec{x} - \vec{w}_i(t)) & \text{if } k_i = -1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $0 \leq \gamma_c$ and $\gamma_r \leq 1$ denotes the learning rates for the nearest and next nearest center. The learning rate decreases with each iteration where $\alpha_r(t) \ll \alpha_c(t)$ must hold. The reduction was carried out following Chen et al. (1992):

$$\alpha(t) = \frac{\alpha(t-1)}{\sqrt{1 + \text{int}\left[\frac{t}{m}\right]}} \tag{5}$$

where, int[·] denotes the integer part of the argument. $t$ is the iteration number and $m$ is the number of clusters.

(iii) Prune any centers that are within a specified threshold $\tau$ of each other.

$$\text{if} ||\vec{w}_m - \vec{w}_n||^2 < \tau, \quad \text{then} \quad s_z = \sum_i ||\vec{w}_z - \vec{x}_i||^2 \tag{6}$$

where $z$ is $m$ or $n$ and $i$ includes all points found in $m$ and $n$. Then

$$\rho(\vec{s}_z) = \begin{cases} 1 & \text{if } s_i < s_j \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $\rho(\vec{s}_z)$ denotes whether a node is pruned ($\rho(\vec{s}_z) = 0$) or is not pruned ($\rho(\vec{s}_z) = 1$).

## 3. Results and discussion

Our implementation of the RPCL algorithm is a topology-determining clustering algorithm. That is, it determines the number of clusters needed to classify the data into biologically similar units. If the number of clusters at the start of the simulation under-represents the topology of the input space, the extra units needed are automatically generated and the points that are closer to these newly generated units are classified appropriately. On the other hand, if there are too many clusters at the start of the analysis, then the unused clusters are pruned away. The RPCL algorithm has been modified to converge on the correct number of clusters based on a similarity criterion we introduced, defined in $\tau$, and to include a node-pruning step wherein any two centers that are closer to each other in $n$-dimensional space by some threshold $\tau$ are assigned to a single cluster. This criterion ($\tau$) that defines a condition dictates the number of cluster formed. The elimination of the center depends on $\rho(\vec{s}_z)$. Centers are also pruned if they are driven away and never assigned a point. In cases where the number of nodes is greater than the number that is needed to represent the input space, the-pruning step in the algorithm prunes the extra and unused nodes converging to the optimal number of nodes.

The performance of the algorithm was first tested on a set of simulated data with known numbers of clusters. Fig. 1 represents the input space with four clusters. The tight clusters (variance 0.01) are very distinct compared to the sparse clusters (variance 0.1). Multiple simulations were performed to determine the uniqueness of the convergence to a particular solution. The learning and penalizing rates were set to 0.05 and 0.002, respectively at the beginning of the simulation and gradually reduced during the simulation following Eq. (5). This helps in reducing fluctuations during learning. The error profiles during training are given in Fig. 2. The profile clearly captures the decrease in the error as the number of clusters/nodes tends toward the optimal solution. The profile also shows that once the optimal number of clusters is reached, there is no further decrease in error. The results of the simulation are summarized in Table 1. In all simulations, the number of starting centers was assumed to be a minimum of 2. Simulations were performed with varying $\tau$ until the algorithm failed to produce the correct number of clusters.

In the case of the dataset with tight clusters, the RPCL algorithm correctly determined the number of clusters with a threshold of 0.5 while it produced nine clusters for the dataset with sparse clusters. This is a result of the dataset with a variance of 0.01 resulting in easily distinguishable clusters, and therefore a $\tau$ of 0.5 was sufficient to differen-

Table 1
Performance of the RPCL algorithm on artificially generated data

| Width | Number of clusters obtained | |
|---|---|---|
| | 0.01[a] | 0.1[a] |
| 0.5 | 4 | 9 |
| 1.0 | 4 | 4 |
| 1.2 | 4 | 4 |
| 1.3 | 4 | 4 |
| 1.4 | 4 | 3 |
| 1.45 | 3 | – |

[a] Variance.

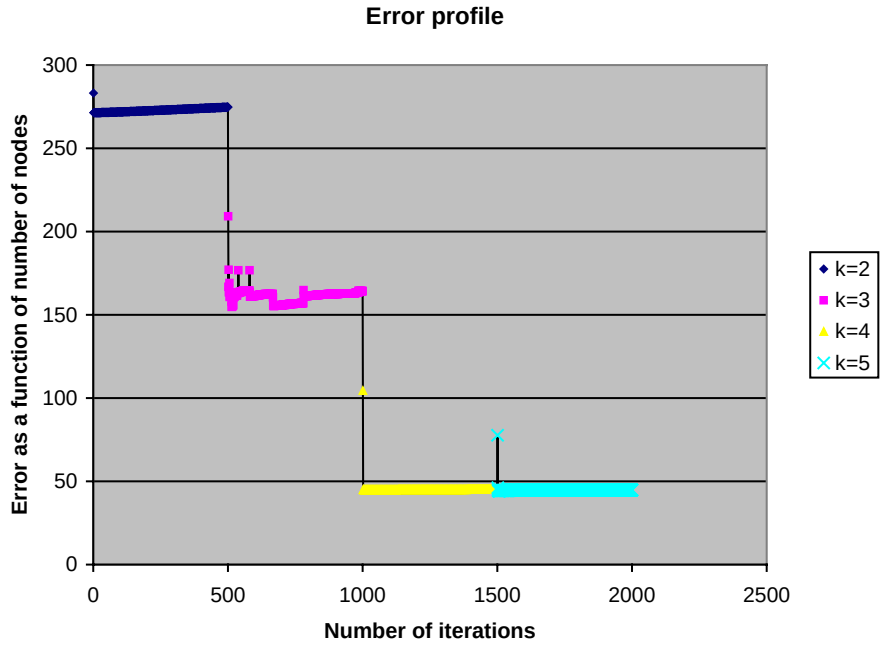


Fig. 2. Error profile for the simulations run on the artificially generated dataset with sparse clusters. *k* represents the number of nodes generated during the simulation.

tiate the tightly packed cluster. Another point worth mentioning is the percentage of correctly classified points. In the case of the tight clusters all the points were correctly classified for $\tau$ between 0.5 and 1.4, giving a performance of 100%. In the case of the sparse clusters, the classification was never 100% accurate. This is due to the overlap of points that are at the tail of the distribution. Fig. 3 depicts the points that were misclassified.

The movement of the centers is shown in Fig. 4(a). The penalization of the centers become more apparent when the number of starting centers is greater than the number of actual centers needed. This is an important step in the simulation as it helps in efficiently determining the optimal centers for the similarity criterion defined in $\tau$. The number of nodes will be automatically selected by driving away any extra units. Since these nodes are far away from the winning centers they are never assigned any points, and empty nodes are eventually pruned. The converged centers for the dataset with sparse clusters are shown in Fig. 4(b). The convergence of the RCPL algorithm to a unique solution even

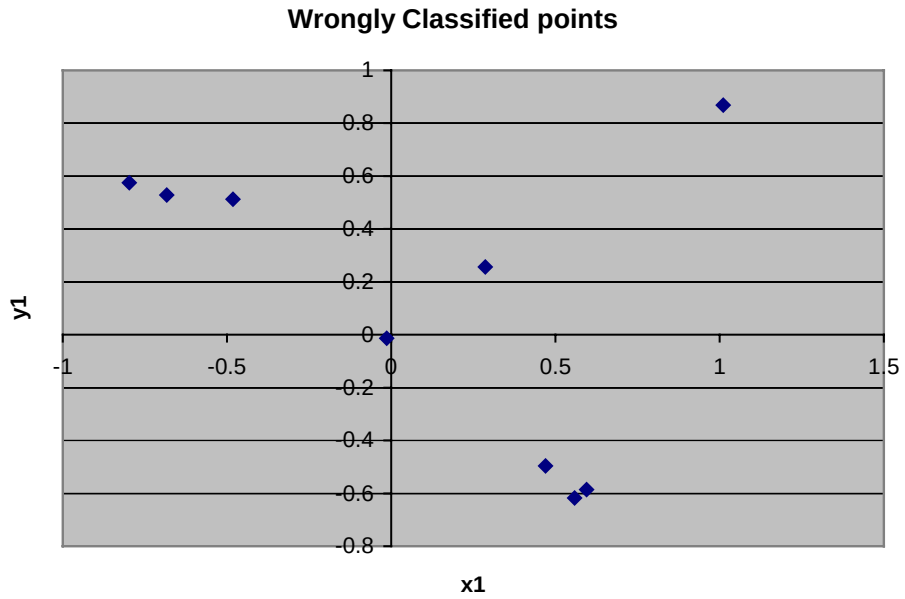


Fig. 3. The points from the dataset with sparse clusters that were wrongly classified by the RPCL algorithm.

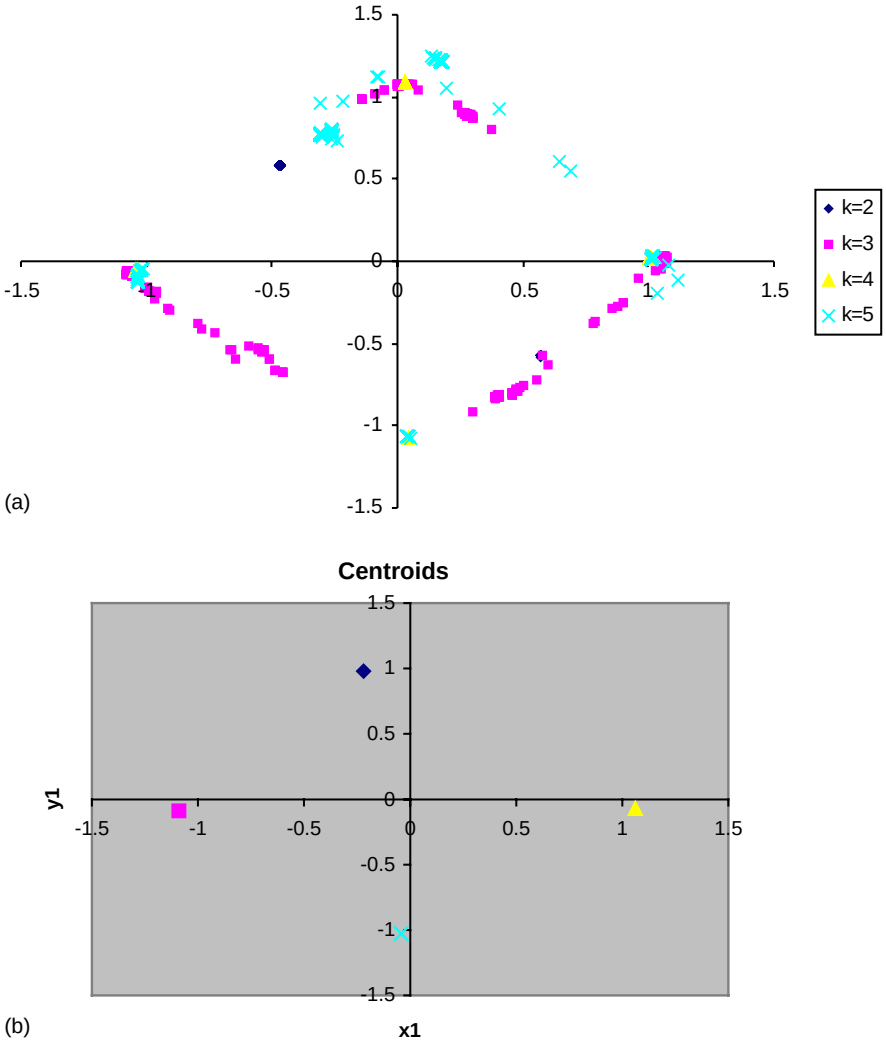


Fig. 4. (a) Movement of the centers during the simulation. $k$ represents the number of nodes. (b) Converged centers arrived at by RPCL at the end of the simulation.

with different $\tau$ and varying numbers of starting clusters, suggests the robustness of the approach and its potential application as a tool for analyzing gene expression data.

Next, the RPCL algorithm was applied to publicly available gene expression data. Two sets of data were chosen. One of the sets consists of expression data used to understand the genome-wide transcriptional analysis of the mitotic cell cycle in yeast (Cho et al., 1998). In this study, the fluctuation of gene expression levels were analyzed as the cells progressed through the cell cycle. The temporal changes in gene expression make this a particularly attractive system to understand genome-wide regulation of gene activity. Cho et al., manually analyzed their expression data to understand the genes involved in the different stages of cell cycle. For each phase of the cell cycle, genes with similar expression levels and patterns of expression were grouped together. This dataset was chosen for two reasons. Firstly, as a test case for evaluating the performance of the algorithm and secondly, in the hope of extracting some as yet unknown and interesting dynamics of gene expression as the cell progresses through the different phases.

The algorithm assigns a gene to a particular cluster based on its distance from the number of clusters involved at the time of classification. The number of clusters involved depends largely on $\tau$ which is specified at the beginning of the analysis. This parameter can be viewed as the distance that must be maintained between any two clusters. From a biological sense $\tau$ is equivalent to the difference in expression levels that need to be differentiated. This feature of the algorithm makes it particularly attractive as it allows for the collapse of two clusters into one or the differentiation of an already existing cluster into one or more clusters. Thus, $\tau$ helps to either differentiate gene expression patterns with fine differences or to integrate them as one based on their expression level.

Table 2
Number of clusters obtained for the yeast expression data

| Width | Yeast clusters obtained |
|---|---|
| 2.5 | 5 |
| 2.0 | 12 ± 3 |
| 1.5 | 24 ± 4 |
| 1.0 | 43 ± 4 |

Results of simulations on the yeast expression data are given in Table 2. The results reveal that the numbers of clusters are inversely proportional to $\tau$. A smaller value of $\tau$ results in the separation of more similar expression patterns while a larger $\tau$ value collapses two clusters and consequently decreases the number of clusters.

Genes associated with each cluster were compared with the manual analysis performed by Cho et al. The assignment of genes to the clusters as they get differentiated depending on the value of $\tau$ revealed a phenomenon of close functional association with the genes within the differentiated nodes. Analysis of the differentiation of the gene-cluster association for the genes involved in the S-phase of the cell cycle by Cho et al., is indicative of this fact. One set of RPCL simulations on the yeast cell cycle data with $\tau = 2$ and $\tau = 1$ converged to a total of 10 and 47 clusters respectively. Analyzing the gene-cluster association for the genes involved in the S-phase of the cell cycle revealed that the algorithm was able to differentiate them into finer functionally distinct clusters. RPCL distributed these genes into 5 and 11 nodes for $\tau$ of 2 and 1, respectively. Lowering the value of $\tau$ from 2 to 1 permitted the separation of genes involved in chromatin organization, cell wall organization and axial budding. Cho et al., manually assigned these genes to the same node while RPCL assigned these genes to different nodes. While these process occurs during the S-phase of the cell cycle, differentiating them into separate groups help in better understanding the finer differences in expression levels. Similarly, genes that were clustered by Cho et al., to be in the late G1 phase and the M-phase were also differentiated into smaller and more distinct groups of clusters. The differentiation of one of these clusters into seven individual clusters, as the value of $\tau$ decreased from 1.5 to 1, is shown in Fig. 5. It is evident from the similar expression patterns of clusters 9, 10 and 40 that they all originated from a single cluster. Furthermore, two other nodes, viz. 5 and 25, also have genes that were once part of a larger node. There were two other genes that were part of the large node that got assigned to two different nodes upon differentiation. Similar trends were obtained with subsequent runs.

The similarity of the expression patterns within a cluster is very apparent, and justifies the fact as to why they were once part of a single cluster or node. Analysis of other clusters also revealed that they were assigned genes that could be categorized to be involved in different cell specific processes. It was also revealed from the analysis that there were genes clustered together with no apparent functional correlation, as well as genes with no function assigned being grouped with genes with known similar functions. This could help in assigning function to these genes and provide basis for further investigation.

While the grouping of genes with known and unknown function is also achieved by other methods, it may not happen at such a fine scale. This mainly stems as a result of inappropriately setting the initial number of clusters. The similarity criterion between members in a cluster and the apparent differences between clusters is captured in the parameter $\tau$. The fine control of separation that can be obtained using the RPCL based approach aids in reducing the functional search space. The analysis of the gene expression patterns during the development and maturation of the rat kidney, which was the other system that we chose to evaluate the model, evinces this fact.

Stuart et al. had studied the gene expression patterns during kidney organogenesis (Stuart et al., 2001). In this study, the expression of over 8,740 rat genes were analyzed. Their analysis revealed five discrete patterns during nephrogenesis. These patterns were clustered into five distinct groups: (i) a group which constituted genes that presented high levels of expression in the early embryonic kidney linking them to translation and replication processes; (ii) a group that showed increased levels during mid-embryogenesis; (iii) another group that peaked during the neonatal period with the predominance of retrotransposon RNAs; (iv) a set of genes that steadily increased throughout the development; (v) a group that was weakly expressed throughout embryogenesis and had increased levels in the adult kidney. Analysis of the expression data revealed three very distinct patterns, which were converged to with a $\tau = 2.0$. These three were the most distinct patterns that were present in the data. It was rather easy for the algorithm to detect these patterns and all of the expression data were first assigned to any one of these patterns, depending on its distance from the centroid. It is however important to realize that developmental stage specific expression of genes could include a variety of functionally distinct genes that would certainly belong to more than three groups. It is also important to understand that two different functionally distinct gene might also have the same expression profile. Thus, within each group there may exist a set of sub groups that may provide additional functional signature to its members. Decreasing the value of $\tau$ can differentiate this additional signature. Upon running the simulations by setting the value of $\tau$ to 1.5, the algorithm converged to four patterns (Fig. 6). Further simulations, by lowering the value of $\tau$ helped in differentiating patterns with finer differences. Simulations with $\tau = 1.0$, converged to a set of five patterns. Fig. 6(c) shows the additional signatures that were separated were close to one of the patterns that were obtained when $\tau$ was set to 2.0. Furthermore, the analysis of genes assigned to the clusters revealed functional co-involvement, similar to those reported earlier (Stuart et al., 2001).

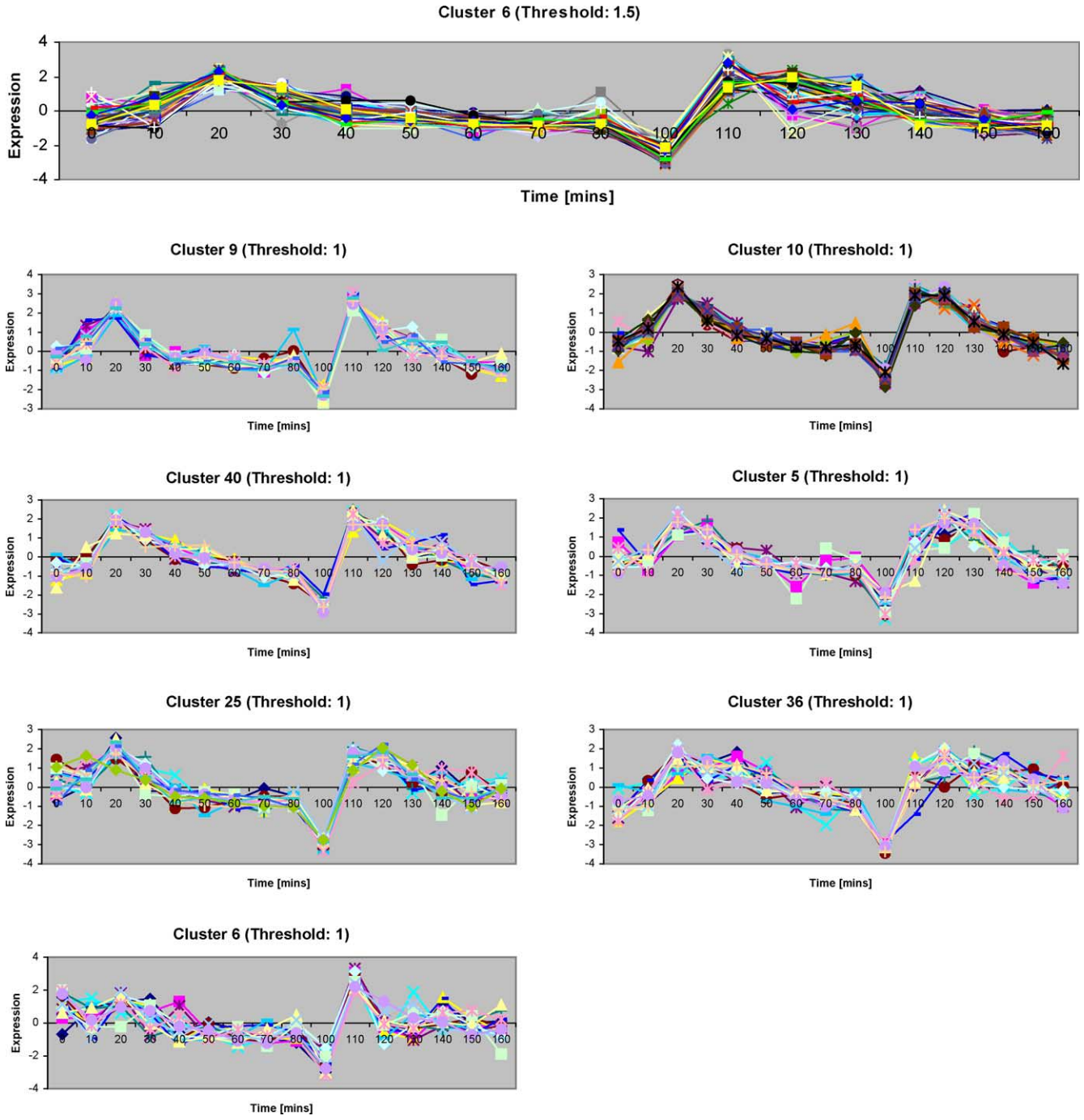


Fig. 5. This figure shows the expression trends of cluster 6 (threshold 1.5) and the trends of clusters containing those same genes when the threshold was raised to 1. All genes found in clusters 9, 10, and 40 are also found in cluster 6 (1.5) and the similar trend of the all four clusters reinforces that fact. Only half of the genes in clusters 25 and 5 are found in cluster 6 (threshold 1.5) and once again the trend of these two graphs compared to cluster 6 (threshold) clearly shows some similarity supporting the fact that some genes are shared while others are not. Cluster 6 and 36 (both of threshold 1) only share one gene each with cluster 6 (threshold 1.5) and once again the trends of the two with threshold of 1 compared to that of cluster 6 (threshold 1.5) clearly shows the majority of the genes in these two clusters are not found in cluster 6 (threshold 1.5).

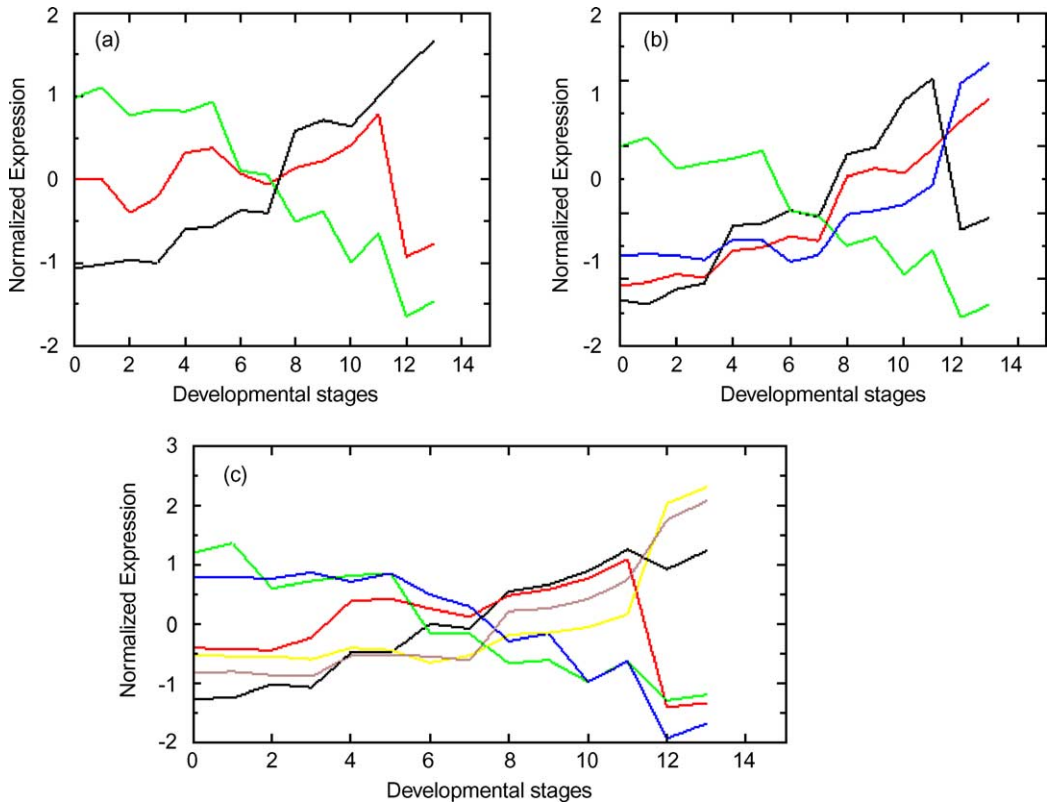


Fig. 6. Centroids of the clusters obtained for the kidney gene expression data. The value of $\tau$ used were (a) 2.0, (b) 1.5, and (c) 1.0. The development stages on the $x$-axis correspond to 0 = 13 embryonic days, 2 = 15 embryonic days, 4 = 17 embryonic days, 6 = 19 embryonic days, 8 = new born, 10 = 1 week and 12 = adult.

## 4. Conclusion

Selecting the correct number of clusters ($k$) for clustering is still an open problem that influences the accuracy and relevance of the clustering results. However, defining a condition that captures similarity between patterns as introduced in our implementation of RPCL, determines the correct number of clusters with the desired distinction between patterns. Since clustering is supposed to uncover similar groups, determining the number of groups heuristically aids in the interpretation of the results. Defining the width between clusters is equivalent to differentiating genes based on their level of expression. This is a more effective approach to arrive at functionally similar genes than by mapping it on to a fixed number of clusters where no similarity criteria are defined. Further, the fine separation obtained using RPCL also helps in reducing the functional search space.

It is important to mention that when clustering data in high dimension there is a possibility of multiple solutions. Thus, when patterns are not easily separable RPCL does produce multiple solutions but in all cases the solutions are within set constraints. An obvious question is: what is a good threshold to use? The answer depends on the data being analyzed and the specific questions being asked about the data. In most cases, it would be possible to mine the expression of a couple of genes and derive a value for $\tau$ from it. We hope the method described here will contribute towards inferring function of unknown genes from their expression profiles.

## Acknowledgements

This work is partly supported by the National Cancer Institute funding grant CA 88351 and NSF grants DBI-9975808 and DBI-0077378. ROS is supported by K08-DK02392. TMN thanks Drs. George Cunningham Virginia de Sa, N.V. Joshi and Sheila Podell for their comments and suggestions. This work is also part of the initial work for a proposed machine-learning project (LEGEND).

## References

Ahalt, S.C., Krishnamurthy, A.K., Chen, P., Melton, D.E., 1990. Competitive learning algorithms for vector quantization. Neural Netw. 3, 277–290.

Bowtell, D.L.D., 1999. Options available—from start to finish—for obtaining data from DNA microarrays II. Nat. Genet. 32 (Suppl.), 481–499.

Chen, S., Billings, S., Grant, P., 1992. Recursive hybrid algorithm for nonlinear system identification using radial basis function networks. Int. J. Contr. 55, 1051–1070.

Cho, R.J., Campbell, M.J., Winzeler, E.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T.G., Gabrielian, A.E., Landsman, D.,

Lockhart, D.J., Davis, R.W., 1998. A genome-wide transcriptional analysis of the mitotic cell cycle. Mol. Cell 2, 65–73.
DeRisi, J., Iyer, V.R., 1999. Genomics and array technology. Curr. Opin. Oncol. 11, 76–79.
DeRisi, J., Iyer, V., Brown, P., 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science 278, 680–686.
DeSieno, D., 1988. Adding a conscience to competitive learning. In: Proceedings of the IEEE International Conference on Neural Networks, vol. 1. San Diego, CA, pp. 117–124.
Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D., 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. U.S.A. 95, 14863–14868.
Granjeaud, S., Bertucci, F., Jordan, B.R., 1999. Expression profiling: DNA arrays in many guises. BioEssays 21, 781–790.
Han, J., Kamber, M., 2001. Data Mining. Morgan Kaufmann, Los Altos, CA.
Jain, A., Dubes, R., 1988. Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs, NJ.
Kaufman, L., Rousseeuw, P., 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York, NY.
Kohonen, T., 1982. Emergence of invariant-feature detectors in the adaptive-subspace self-organizing map. Biol. Cybernet. 43, 59–69.
Kohonen, T., 1990. The self-organizing map. Proc. IEEE 78, 1464–1480.
Lipshutz, R.J., Fodor, S.P.A., Gingeras, T.R., Lockhart, D.J., 1999. High density synthetic oligonucleotide arrays. Nat. Genet. Suppl. 21, 20–24.
Lockhart, D.J., Dong, H., Byrne, M.C., Follettie, M.T., Gallo, M.V., Chee, M., Mittmann, M., Wang, C., Kobayashi, M., Horton, Brown, E.L., 1996. Expression monitoring by hybridization to high-density oligonucleotide arrays. Nat. Biotechnol. 14, 1675–1680.
Morgan, B.J.T., Ray, A.P.G., 1995. Non-uniqueness and inversions in cluster-analysis. Appl. Stat.: J. R. Stat. Soc., Ser. C 44 (1), 117–134.
Saban, M., Hellmich, H., Nguyen, N., Winston, J., Hammond, T., Saban, R., 2001. Physiol. Genom. 5, 147–160.
Spellman, P.T., Sherlock, G., Zhang, M.Q., Botstein, D., Futcher, B., 1998. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. Mol. Biol. Cell 9, 3273–3297.
Stuart, R.O., Bush, K.T., Nigam, S.K., 2001. Changes in global gene expression patterns during development and maturation of the rat kidney. Proc. Natl. Acad. Sci. U.S.A. 98, 5649–5654.
Sultan, M., Wigle, D.A., Cumbaa, C.A., Maziarz, M., Glasgow, J., Tsao, M.S., Jurisica, I., 2002. Binary tree-structured vector quantization approach to clustering and visualizing microarray data. Bioinformatics 18 (1), 111S–119S.
Takahashi, M., Rhodes, D.R., Furge, K.A., Kanayama, H., Kagawa, S., Haab, B.B., The, B.T., 2001. Gene expression profiling of clear cell renal cell carcinoma: gene identification and prognostic classification. Proc. Natl. Acad. Sci. U.S.A. 98, 9754–9759.
Tambe, S.S., Kulkarni, B.D., Deshpande, P.B., 1996. Elements of Artificial Neural Networks with Selected Application in Chemical Engineering, and Chemical and Biological Sciences. SAC, Louisville, KY, USA.
Tamayo, P., Slonim, D., Mesirov, K., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E., Golub, T., 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc. Natl. Acad. Sci. U.S.A. 96, 2907–2912.
Thieffry, D., 1999. From global expression data to gene networks. BioEssays 21, 895–899.
Toronen, P., Kolehmainen, M., Wong, G., Castren, E., 1999. Analysis of gene expression data using self-organizing maps. FEBS 451, 142–146.
Tusher, V.G., Tibshirani, R., Chu, G., 2001. Significance analysis of microarrays applied to the ionizing radiation response. Proc. Natl. Acad. Sci. U.S.A. 98, 5116–5121.
Xu, L., Krzyzak, A., Oja, E., 1993. Rival penalized competitive learning for cluster analysis RBF net and curve detection. IEEE Trans. Neural Netw. 4 (4), 636–649.