Computers in Biology and Medicine 36 (2006) 1126–1142

Computers in Biology and Medicine

www.intl.elsevierhealth.com/journals/cobm

Unsupervised clustering in mRNA expression profiles

D.K. Tasoulis[a, b], V.P. Plagianakos[a, b], M.N. Vrahatis[a, b, *]

[a] *Computational Intelligence Laboratory (CILAB), Department of Mathematics, University of Patras, GR-26110 Patras, Greece*

[b] *University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-26110 Patras, Greece*

Abstract

The development of microarray technologies gives scientists the ability to examine, discover and monitor the mRNA transcript levels of thousands of genes in a single experiment. Nonetheless, the tremendous amount of data that can be obtained from microarray studies presents a challenge for data analysis. The most commonly used computational approach for analyzing microarray data is cluster analysis, since the number of genes is usually very high compared to the number of samples. In this paper, we investigate the application of the recently proposed *k*-windows clustering algorithm on gene expression microarray data. This algorithm apart from identifying the clusters present in a data set also calculates their number and thus requires no special knowledge about the data. To improve the quality of the clustering, we employ various dimension reduction techniques and propose a hybrid one. The results obtained by the application of the algorithm exhibit high classification success.

Keywords: Bioinformatics; Gene expression analysis; mRNA expression profiles; *k*-windows algorithm; Unsupervised clustering; Cluster analysis; Dimension reduction

1. Introduction

In any living cell that undergoes a biological process, different subsets of its genes are expressed. A cell's proper function is crucially affected by the gene expression at a given stage and their relative abundance. To understand biological processes one has to measure *gene expression levels* in different developmental phases, different body tissues, different clinical conditions and different organisms. This

[*] Corresponding author. Tel.: +30 2610 997374.

E-mail addresses: dtas@math.upatras.gr (D.K. Tasoulis), vpp@math.upatras.gr (V.P. Plagianakos), vrahatis@math.upatras.gr (M.N. Vrahatis).

0010-4825/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiomed.2005.09.003

kind of information can aid in the characterization of gene function, the determination of experimental treatment effects, and the understanding of other molecular biological processes [1].

Compared to the traditional approaches to genomic research, which rely on the collection and examination of data for a single gene locally, DNA microarray technologies have rendered possible to monitor the expression pattern for thousands of genes simultaneously. Unfortunately, the original gene expression data come along with noise, missing values and systematic variations due to the experimental procedure. Several methodologies can be employed to alleviate these problems, such as singular value decomposition based methods, weighted k-nearest neighbors, row averages, replication of the experiments to model the noise, and/or normalization, which is the process of identifying and removing systematic sources of variation. After gene expression levels are measured, the data are represented by the real-valued expression matrix X, where the rows of the matrix are vectors forming the expression patterns of genes, the columns of the matrix represent samples from various conditions, and each cell, x_{ij}, is the measured expression level of gene i in sample j.

Discovering the patterns hidden in gene expression microarray data is a tremendous opportunity and challenge for functional genomics and proteomics [1]. A promising approach to address this task is to utilize data mining techniques. Cluster analysis is a key step in understanding how the activity of genes varies during biological processes and is affected by disease states and cellular environments. In particular, clustering can be used either to identify sets of genes according to their expression in a set of samples [2,3], or to cluster samples into homogeneous groups that may correspond to particular macroscopic phenotypes [4]. The latter is in general more difficult because of the *curse of dimensionality* [5] (due to the limited number of samples and the high feature dimensionality).

Generally, clustering can be defined as the process of "grouping a collection of objects into subsets or clusters, such that those within one cluster are more closely related to one than objects assigned to different clusters" [6]. Clustering is applied in various fields including data mining [7], statistical data analysis [8], compression and vector quantization [9], global optimization [10,11], and image analysis among others. Clustering is, also, extensively applied in social sciences [8]. Recently, clustering techniques have been applied to gene expression data [2,12–16] and have proved useful for identifying biologically relevant groupings of genes and samples. Thereby clustering techniques have further helped to address questions such as gene function, gene regulation and gene expression differentiation under various conditions.

A fundamental issue in cluster analysis, independent of the particular clustering technique applied, is the determination of the number of clusters present in a data set. This issue remains an open problem in cluster analysis. For instance, well-known and widely used iterative techniques, like the k-means algorithm [17] and the Fuzzy c-means algorithm [18], require from the user to specify the number of clusters present in the data prior to the execution of the algorithm. Algorithms that have the ability to estimate the number of clusters present in a data set fall in the category of unsupervised clustering algorithms.

In [1], a survey of various clustering methods is performed. Furthermore, subspace clustering techniques are examined. In [19] a min–max cut hierarchical clustering method is presented that attempts to produce clusters quite close to human expert labeling. Moreover, they employ the F-statistic test and the Principal Component Analysis (PCA) technique for gene selection. For the same data set their approach exhibits classification rates close to the ones presented in this paper. In the present work we extend previous approaches by examining the performance of a method that attempts to perform dimension reduction, estimate the number of clusters, and classify the data. To this end, we investigate the application of the recently proposed clustering algorithm k-windows [20] on gene expression microarray data. The unsupervised k-windows (UKW) in addition to partitioning the data into clusters, it also approximates the number of clusters during its execution. We have compared our approach against four other well-known clustering algorithms and the results were satisfactory.

The rest of the paper is organized as follows: in the next section, we provide a brief literature review and in Section 3 feature selection techniques are discussed. In Section 4, the UKW algorithm is presented, and (for completeness purposes) we briefly describe the four other clustering algorithms tested. Next, in Section 5 we present results from the application of the clustering algorithms on gene expression microarray data. The paper ends with concluding remarks.

2. Brief literature review

Although numerous clustering algorithms exist [21], mostly hierarchical clustering methods have been applied to microarray data. Hierarchical clustering algorithms construct hierarchies of clusters in a top–down (agglomerative) or bottom-up (divisive) fashion. This kind of algorithms have proved to give high quality results. One of the most representative hierarchical approaches is the one developed by Eisen et al. [2]. In that work, the authors employed an agglomerative algorithm and adopted a method for the graphical representation of the clustered dataset. This method has been widely used by many biologists and has become the most widely used tool in gene expression data analysis [12,22,23]. Nonetheless, the high sensitivity of agglomerative methods to small variations of the inputs and the high computational requirements, usually prevents their usage in real applications, where the number of samples and their dimensionality is expected to be high (the cost is quadratic to the number of samples).

Partitioning clustering algorithms, start from an initial clustering (that may be randomly formed) and create flat partitionings by iteratively adjusting the clusters based on the distance of the data points from a representative member of the cluster. The most commonly used partitioning clustering algorithm is k-means. k-means initializes k centers and iteratively assigns each data point to the cluster whose centroid minimizes the Euclidean distance from the data point. Although, k-means type algorithms can yield satisfactory clustering results at a low cost, as their running time is proportional to kn, where n is the number of samples, they heavily depend on the initialization.

Graph theoretical clustering approaches construct a proximity graph, in which each data point corresponds to a vertex, and the edges among vertices model their proximity. Xing and Karp [16], developed a sample-based clustering algorithm named Clustering via Iterative Feature Filtering (CLIFF) which iteratively employs sample partitions as a reference to filter genes. The selection of genes through this approach relies on the outcome of an NCut algorithm, which is not robust to noise and outliers.

Another graph theoretical algorithm, Clustering Identification via Connectivity Kernels (CLICK) [13], tries to recognize highly connected components in the proximity graph as clusters. The authors demonstrated the superior performance of CLICK to the approaches of Eisen et al. [2], and the self organizing map [24] based clustering approach. However, as claimed in [1], CLICK has little guarantee of not generating highly unbalanced partitions. Furthermore, in gene expression data, two clusters of co-expressed genes, C1 and C2, may be highly intersected with each other. In such situations, C1 and C2 are not likely to be split by CLICK, but would be reported as one highly connected component.

Finally, Alter et al. [25], by examining the projection of the data to a small number of principal components obtained through a principal component analysis, attempt to capture the majority of gene variations. However, the large number of irrelevant genes does not guarantee that the discriminatory information will be highlighted to the projected data. For an overview of the related literature see [1] and [26].

3. Feature selection techniques

An important issue in any classification task is to identify those features that significantly contribute to the classification of interest, while at the same time discarding the least significant and/or erroneous ones. This procedure is also referred to as dimension reduction. The problem of high dimensionality is often tackled by user specified subspaces of interest. For example, in [4] the authors manually identified a subset consisting of 50 out of 7129 genes from 72 leukemia patients. However, user-identification of the subspaces is error-prone and time consuming, especially when no prior domain knowledge is available.

Another way to address high dimensionality is to apply a dimension reduction method to the data set. Methods such as the principal component analysis [27], optimally transform the original data space into a lower dimensional space by forming dimensions that are linear combinations of given attributes. The new space has the property that distances between points remain approximately the same as before.

PCA is a powerful multivariate data analysis method. Its main purpose is to reduce and summarize large and high dimensional data sets by removing redundancies and identifying correlation among a set of measurements or variables. It is a useful statistical technique that has found numerous applications in different scientific fields such as face recognition, image processing and compression, molecular dynamics, information retrieval, and recently gene expression analysis. PCA is used in gene expression analysis mainly to compute an alternative representation of the data using a much smaller number of variables, as well as, to detect characteristic patterns in noisy data of high dimensionality. More specifically, PCA is a way of identifying patterns in data and expressing the data in such a way as to highlight their similarities and differences. Since patterns in high dimensional data can be hard to find, PCA is a powerful tool of analysis, especially when the visualization of the data is not possible.

Although PCA may succeed in reducing the dimensionality, the new dimensions can be difficult to interpret. Moreover, to compute the new set of dimensions information from all the original dimensions is required. The selection of a subset of attributes in the context of clustering is studied in [28,29]. In the context of classification, subset selection has also been studied [27].

Another approach is to employ clustering techniques to perform dimension reduction [30]. Specifically, we applied the UKW clustering algorithm to identify features of interest. The clustering algorithm was applied over the entire data set to identify meaningful clusters of features and to select the most informative ones that will be used for classification. Feature selection was accomplished by extracting from each cluster one representative feature, based on the Euclidean distance among the feature values and the identified cluster center. The feature with the minimum distance from the cluster center was selected. It must be noted that the UKW algorithm automatically approximates the number of clusters present in the data set.

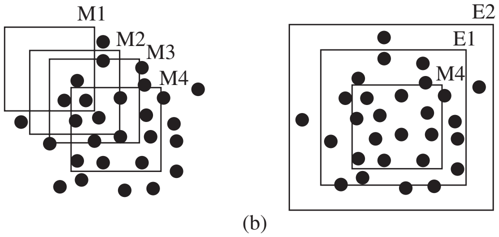

Fig. 1. (a) Sequential movements of the initial window M1 that result to the final window M4. (b) Sequential enlargements of the initial window M4 that result to the final window E2.

4. Clustering algorithms

In this section we present the k-windows clustering algorithm [20,31] along with its complexity issues and how it is extended to endogenously approximate the number of clusters present in a data set. Additionally, we briefly describe the four well-known clustering algorithms tested in this paper, namely, (a) the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm [32], (b) the Principal Direction Divisive Partitioning (PDDP) clustering algorithm [33], (c) the Fuzzy c-means (FCM) clustering algorithm [18], and (d) the Growing Neural Gas (GNG) [34].

4.1. The UKW clustering algorithm

Here we outline the basic concepts of the UKW algorithm that generalizes the k-windows clustering algorithm [20]. Suppose that we have a set of points in the $\mathbb{R}^d$ space. Intuitively, the k-windows algorithm tries to place a d-dimensional window (box) containing all patterns that belong to a single cluster; for all clusters present in the data set. At first, k points are selected (possibly in a random manner). The k initial d-ranges (windows), of size a, have as centers these points. Subsequently, the patterns that lie within each d-range are identified. Next, the mean of the patterns that lie within each d-range (i.e. the mean value of the d-dimensional points) is calculated. The new position of the d-range is such that its center coincides with the previously computed mean value. The last two steps are repeatedly executed as long as the increase in the number of patterns included in the d-range that results from this motion satisfies a stopping criterion. The stopping criterion is determined by a variability threshold θ_{v} that corresponds to the least change in the center of a d-range that is acceptable to recenter the d-range. This process is illustrated in Fig. 1(a).

Once the movement is terminated, the d-ranges are enlarged to capture as many patterns as possible from the cluster. Enlargement takes place at each dimension separately. The d-ranges are enlarged by θ_{e}/l percent at each dimension, where θ_{e} is user defined, and l stands for the number of previous successful enlargements. After the enlargement in one dimension is performed, the window is moved, as described above. Once movement terminates, the proportional increase in the number of patterns included in the window is calculated. If this proportion does not exceed the user-defined coverage threshold, θ_{c}, the enlargement and movement steps are rejected and the position and size of the d-range are reverted to their prior to enlargement values. Otherwise, the new size and position are accepted. If enlargement is accepted for dimension $d' \geqslant 2$, then for all dimensions d'', such that $d'' < d'$, the enlargement process is performed again assuming as initial position the current position of the window. This process terminates

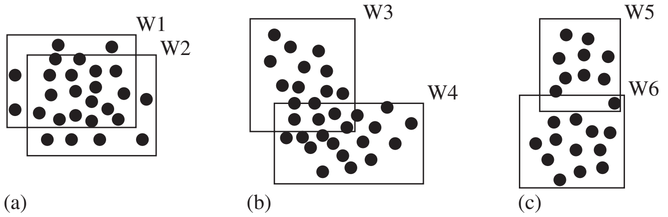

Fig. 2. The merging procedure. Windows W1 and W2 fulfill the similarity condition and thus window W1 is disregarded. Windows W3 and W4 have many points in common thus they are considered to belong to the same cluster. Finally, windows W5 and W6 capture two different clusters.

if enlargement in any dimension does not result in a proportional increase in the number of patterns included in the window beyond the threshold θ_c. An example of this process is illustrated in Fig. 1(b).

To automatically determine the number of clusters, the original *k*-windows algorithm is applied using a sufficiently large number of initial windows. The windowing technique allows for a large number of initial windows to be efficiently examined, without any significant overhead in time complexity. Once all the processes of movement and enlargement for all windows are terminated, all overlapping windows are considered for merging. The merge operation is guided by a merge threshold θ_m. Having identified two overlapping windows, the number of patterns that lie in their intersection is calculated. Next the proportion of this number to the total patterns included in each window is calculated. If the mean of these two proportions exceeds θ_m, then the windows are considered to belong to a single cluster and are merged, otherwise not. This operation is illustrated in Fig. 2. In Fig. 2(a) the extent of overlapping between windows W1 and W2 exceeds the threshold criterion, both are considered to capture the same cluster, and therefore W1 is deleted. On the other hand, in Fig. 2(b) windows W3 and W4 are considered to capture parts of the same cluster. Finally, in Fig. 2(c) windows W5 and W6, are considered to capture two different clusters.

To summarize, the UKW algorithm is a robust unsupervised clustering algorithm and its performance is predictable. The algorithm takes as input six easily tuned user-defined parameters. In this study no effort has been made to fine-tune these parameters. Instead, default values have been used in all the experiments. More specifically, the initial window size was $a=5$; the enlargement threshold, was $\theta_e=0.8$; the merging threshold, was $\theta_m=0.1$; the coverage threshold, was $\theta_c=0.2$; and the variability threshold, was $\theta_v=0.02$.

The computational complexity of the UKW algorithm depends on the complexity of determining the points that lie in a specific window. This is the well studied *orthogonal range search* problem [35]. Numerous Computational Geometry techniques have been proposed [35–38] to address this problem. All these techniques employ a preprocessing stage at which they construct a data structure that stores the patterns. This data structure allows them to answer range queries fast. For applications of very high dimensionality, data structures like the Multidimensional Binary Tree [35], and Bentley and Maurer [37] seem more suitable. On the other hand, for low dimensional data with a large number of points the approach of Alevizos [36] appears more attractive. For the multidimensional binary tree used here, the time complexity is $O(ckdn^{1/d})$, where c is the iteration number, k is the number of initial windows used, n is the number of samples, and d is the dimension of the data.

The UKW algorithm has been successfully applied in numerous applications including bioinformatics [39–41], medical diagnosis [42,43], time series prediction [44] and web personalization [45]. In [46] the

UKW is presented in detail and several modifications for different distributed environments and dynamic databases are proposed.

4.2. The DBSCAN clustering algorithm

The DBSCAN algorithm [32,47] relies on a density-based notion of clusters and is designed to discover clusters of arbitrary shape, as well as, to distinguish noise. More specifically, the algorithm is based on the idea that for each point in a cluster at least a minimum number of objects (*Mints*) should be contained in a neighborhood of given radius (ESP) around it. Thus by iteratively scanning all the points in the data set it forms clusters of points that are connected through chains of ESP-neighborhoods of at least *Mints* points each.

4.3. The PDDP clustering algorithm

The PDDP algorithm [33], is a divisive clustering algorithm. The key component in this algorithm is the computation of the principal directions of the data. Starting with an initial cluster of all the data points, the algorithm iteratively splits the clusters. The use of a distance or similarity measure is limited to deciding which cluster should be split next, but the similarity measure is not used to perform the actual splitting. In detail, all the data points are projected onto the leading eigenvector of the covariance matrix of the data. Based on the sign of that projection the algorithm splits an initial cluster into two. This fact enables the algorithm to operate on extremely high dimensional spaces. PDDP, as well as PDDP(l) [48], which is a recent generalization of PDDP, does not provide a direct estimation for the number of clusters. Proposed methods that provide such estimations through these algorithms are based on scattering of the data around their centroids. Nonetheless, they tend to overestimate the true number of clusters resulting in rigid clustering [33,48].

4.4. The fuzzy c-means clustering algorithm

The FCM algorithm [18], considers each cluster as a fuzzy set. It firstly initializes a number of c prototype vectors (centroids) p^j over the data set. Each centroid represents the center of a cluster. At a next step it computes the degree of membership of every data vector, x^i, to each cluster using the membership function:

$$\mu_j(x^i) = \left(\sum_{l=1}^{c} \left(\frac{\|x^i - p^j\|}{\|x^i - p^l\|} \right)^{1/r-1} \right)^{-1}$$

which takes values in the interval [0, 1], where $r \in (1, \infty)$ determines the fuzziness of the partition. If r tends to 1_+, then the resulting partition asymptotically approaches a crisp partition. On the other hand, if r tends to infinity, the partition becomes a maximally fuzzy partition. Next the c prototypes are updated using the following equation:

$$P^j = \frac{\sum_{i=1}^{n} [m_j(x^i)]^r x^i}{\sum_{i=1}^{n} [m_j(x^i)]^r}.$$

This procedure is iteratively executed until the measure of distortion:

$$d = \sum_{j=1}^{c} \sum_{i=1}^{n} [m_j(x^i)]^r \|x^i - p^l\|^2,$$

changes less than a user defined threshold.

4.5. Growing neural gas

GNG [34] is an incremental neural network. It can be described as a graph consisting of k nodes, each of which has an associated weight vector, w_j, defining the node's position in the data space and a set of edges between the node and its neighbors. During the clustering procedure, new nodes are introduced into the network until a maximal number of nodes is reached. GNG starts with two nodes, randomly positioned in the data space, connected by an edge. Adaptation of weights, i.e. the nodes position, is performed iteratively. For each data object the closest node (winner), s_1, and the closest neighbor of a winner, node s_2, are determined. These two nodes are connected by an edge.

An age variable is associated with each edge. At each learning step the ages of all edges emanating from the winner are increased by 1. When the edge connecting s_1 and s_2 is created its age is set to 0. By tracing the changes of the age variable inactive nodes are detected. Any nodes having no emanating edges and edges exceeding a maximal age are removed.

The neighborhood of the winner is limited to its topological neighbors. The winner and its topological neighbors are moved in the data space toward the presented object by a constant fraction of the distance, defined separately for the winner and its topological neighbors. There is no neighborhood function or ranking concept. Thus, all topological neighbors are updated in the same manner.

5. Experimental results

To investigate the performance of the UKW algorithm on gene expression microarray data we primarily used data from a previous study that examined mRNA expression profiles from 72 leukemia patients to develop an expression-based classification method for acute leukemia [4]. This data set contains a large number of patients and has been well characterized. We performed two sets of experiments. In the first set, the UKW algorithm was applied on two previously published gene subsets as well as their union. The comparative results indicate that the UKW exhibits the best performance, among the clustering algorithms tested.

The second set of experiments, we do not use class information for the gene selection. To this end, the PCA technique, as well as, the UKW algorithm were used to perform dimension reduction. Subsequently, UKW was applied on the reduced data set to group samples into clusters. The second set of experiments is closer to real life applications where no class information is a priori known, and the UKW exhibited robust performance and promising results. Moreover, a hybridization of UKW and the PCA technique is evaluated.

The hybrid scheme was able to provide results equivalent to those obtained with the supervised gene selection. Thus this scheme is applied on three other datasets in a third set of experiments.

5.1. Clustering based on supervised gene selection

In the data set each sample is measured over 7129 genes. The first 38 samples were used for the clustering process (train set), while the remaining 34 were used to evaluate the clustering result (test set). The initial 38 samples contained 27 acute myeloid leukemia (AML) samples and 11 acute lymphoblastic leukemia (ALL) samples. The test set contained 20 ALL samples and 14 AML samples. Golub et al. in [4] applied the Self Organizing Map [49] (SOM) based clustering approach on the training set, selecting 50 highly correlated genes with the ALL-AML class distinction. SOM automatically grouped the 38 samples into two classes, one containing 24 out of the 25 ALL samples, and the other containing 10 out of the 13 AML samples.

Generally, in a typical biological system, it is often not known how many genes are sufficient to characterize a macroscopic phenotype. In practice, a working mechanistic hypothesis that is testable and largely captures the biological truth, seldom involves more than a few dozens of genes. Therefore, identifying the relevant genes is critical [16]. Initially we applied the UKW algorithm over the train set using all 7129 genes as well as various randomly selected gene collections ranging from 10 to 2000. The algorithm produced clusters that often contained both AML and ALL samples. Typically, at least 80% of all the samples that were assigned to a cluster were characterized by the same leukemia type.

To improve the quality of the clustering, it proved essential to identify sets of genes that significantly contribute to the partition of interest. Clearly, there exist many such sets and it is difficult to determine the best one. To this end, we tested the clustering algorithm on two previously discovered sets of significant genes. The first set has been published in the original paper of Golub et al. [4] (we call it $GeneSet_1$), while the second set has been statistically discovered by Thomas et al. [50] ($GeneSet_2$). Each dataset contains 50 genes. Furthermore, we tested the clustering algorithms on the union of the above gene sets ($GeneSet_3$), consisting of 72 genes.

Regarding the second set of genes ($GeneSet_2$), the 50 most highly correlated genes with the ALL-AML class distinction (top 25 differentially expressed probe sets in either sample group) have been selected. More specifically, the selection approach is based on well-defined assumptions, uses rigorous and well-characterized statistical measures, and accounts for the heterogeneity and genomic complexity of the data. The modeling approach uses known sample group membership to focus on expression profiles of individual genes in a sensitive and robust manner, and can be used to test statistical hypotheses about gene expression.

The first step in the statistical analysis of microarray expression profiles is preprocessing and/or transformation of the data. This includes removal of the spiked inoculated controls. The second step is to estimate correction factors for sample-specific heterogeneity, as well as for chip-specific heterogeneity, and to use these factors to normalize the data. The final step is to perform a regression analysis to estimate the relevant model parameters for each gene transcript using robust statistical techniques in order to assess the confidence level that the corresponding gene is differentially expressed between the two groups.

Applying the UKW algorithm on those 3 gene train sets, each produced 6 clusters containing ALL or AML samples. Table 1 exhibits the results. More specifically, the algorithm using $GeneSet_1$ discovered 4 ALL clusters and 2 AML clusters (3 misclassifications), while using $GeneSet_2$ discovered 4 clusters containing only ALL samples and 2 clusters containing only AML samples (0 misclassifications). The algorithm discovered 4 ALL clusters and 2 AML clusters (1 misclassification) when applied to $GeneSet_3$. $GeneSet_2$ yielded the best results in the training set (followed by $GeneSet_3$).

Table 1
The performance of the UKW algorithm for the different train sets

Leukemia type	ALL clusters				AML clusters	
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2
Clustering result for the train set $GeneSet_1$						
ALL accuracy: 87.5 %—*AML accuracy:* 100 %						
ALL	4	4	12	4	3	0
AML	0	0	0	0	4	7
Clustering result for the train set $GeneSet_2$						
ALL accuracy: 100.0 %—*AML accuracy:* 100 %						
ALL	10	3	10	4	0	0
AML	0	0	0	0	8	3
Clustering result for the train set $GeneSet_3$						
ALL accuracy: 95.83 %—*AML accuracy:* 100 %						
ALL	8	9	5	4	0	1
AML	0	0	0	0	7	4

Table 2
The performance of the UKW algorithm for the different test sets

Leukemia type	ALL clusters				AML clusters	
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2
Clustering result for the test set $GeneSet_1$						
ALL accuracy: 60.00 %—*AML accuracy:* 92.85 %						
ALL	2	0	7	3	8	0
AML	1	0	0	0	8	5
Clustering result for the test set $GeneSet_2$						
ALL accuracy: 100 %—*AML accuracy:* 78.57 %						
ALL	8	0	9	3	0	0
AML	0	0	3	0	8	3
Clustering result for the test set $GeneSet_3$						
ALL accuracy: 90 %—*AML accuracy:* 100 %						
ALL	10	4	3	1	0	2
AML	0	0	0	0	5	9

To further evaluate the clustering results each sample from each test set was assigned to one of the clusters discovered in the train set according to its distance from the cluster center. Specifically, if an ALL (AML) sample from the test set was assigned to an ALL (AML, respectively) cluster then that sample was considered correctly classified. From the results exhibited in Table 2 it is evident that using the clustering from $GeneSet_1$ 1 AML and 8 ALL samples from the test set were misclassified, resulting in a 73.5% correct classification. The clusters discovered using $GeneSet_2$ resulted in 3 misclassified AML

Table 3
Comparative results for the test set *GeneSet*$_3$

	Misclassified samples		Number of clusters		Accuracy (%)	
	Train set	Test set	Train set	Test set	AML	ALL
DBSCAN	1	3	4	4	78.5	100
FCM	1	2	4	4	85.7	100
GNG	1	3	3	3	78.5	100
PDDP	2	4	6	6	71.4	100
UKW	1	2	6	6	100.0	90.0

samples (91.2% correct classification), while *GeneSet*$_3$ clusters yielded the best performance with only 2 misclassified ALL samples (94.1% correct classification).

In Table 3 we present comparative results from the test set *GeneSet*$_3$ only, as all algorithms exhibited improved classifications performance on this dataset. The best performance was achieved by the UKW algorithm and the FCM, followed by the DBSCAN and GNG algorithms. Notice that the FCM requires from the user to supply the number of clusters (supervised algorithm) and that the DBSCAN algorithm did not classify 7 samples of the train set and 5 samples of the test set (all of them belonging in the AML class), since it characterized them as outliers.

Although the PDDP algorithm exhibited the worst classification performance, it must be noted that it was the only algorithm capable of using all the 7129 genes to cluster the samples. Using the complete set of genes, the PDDP algorithm misclassified 2 samples from the training set and 8 samples from the test set.

5.2. *Clustering based on unsupervised gene selection*

In this section we investigate the performance of the proposed UKW algorithm on data sets selected using unsupervised methods (no class information is necessary). We only use the UKW algorithm, since the other algorithms tested above are not suitable for unsupervised clustering. Firstly, we compute a new data set using the UKW algorithm and then the same algorithm is used to group the samples (biclustering). More specifically, the UKW algorithm was applied over the entire data set to select clusters of genes. Feature selection was accomplished by extracting from each cluster one representative feature, based on the Euclidean distance among the feature values and the identified cluster center. The feature with the minimum distance from the cluster center was selected. This approach produced a new subset containing 293 genes (*GeneSet*$_4$).

The UKW algorithm was then applied on *GeneSet*$_4$ to group the samples. The results are illustrated in Table 4. From this table it is evident that high classification accuracy is possible even when class information is not known. Specifically, UKW exhibited accuracy of 93.6% and 76% for the ALL and the AML samples, respectively.

A second set of experiments is performed using the PCA technique for dimension reduction. A common problem when using PCA is that there is no clear answer to the question of how many factors should be retained for the new data set. A rule of thumb is to inspect the *scree plot*, i.e. plot all the eigenvalues in decreasing order. The plot looks like the side of a hill and “scree” refers to the debris fallen from the

Table 4
The performance of the UKW algorithm for the *GeneSet*$_4$ data set

Leukemia type	ALL clusters					AML cluster
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 1
Clustering result for the set GeneSet$_4$						
ALL accuracy: 93.61 %—*AML accuracy:* 76 %						
ALL	12	5	8	16	3	3
AML	2	0	3	0	1	19

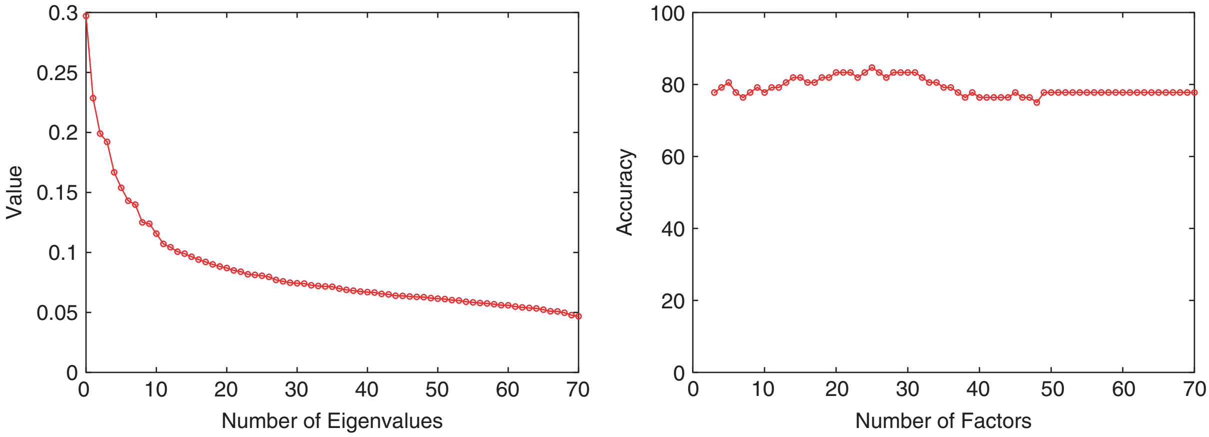

Fig. 3. Plot of the 70 first eigenvalues in decreasing order (left) and the corresponding classification accuracies (right).

top and lying at its base. The scree test suggests to stop analysis at the point the mountain (signal) ends and the debris (error) begins. However, for the considered problem the scree plot was indicative, but not decisive. The scree plot, exhibited in Fig. 3 (left), suggests that the contributions are relatively low after approximately ten components. In our experiments, we tried all the subsets using factors from 2 to 70. The classification accuracy is shown in Fig. 3 (right). The best performance was attained when 25 factors were used (84.72%).

Although, the PCA technique optimally transforms the data set, with limited loss of information, to a space of significantly lower dimension, the classification accuracy, was not as high as when supervised methods for gene selection were used (see Section 5.1). Next, we study the hybridization of the UKW algorithm and the PCA technique.

To this end, the entire data set is firstly partitioned into clusters of features using the UKW algorithm. Next, each feature cluster is independently transformed to a lower dimension space through the PCA technique. Regarding the number of factors selected from each cluster many approaches could be followed. In our experiments only two factors from each cluster were selected, resulting in *GeneSet*$_5$. Experiments conducted using scree plots exhibited identical results. Our experience is that the number of selected factors from each cluster is not critical, since the entire data set has already been clustered. Finally, the UKW algorithm is again applied to group the samples into clusters and the results are exhibited in Table 5. The UKW exhibited accuracy 97.87% and 88% for the ALL and the AML samples, respectively.

Table 5
The performance of the UKW algorithm for the *GeneSet*$_5$ data set

Leukemia type	ALL clusters				AML clusters	
	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 1	Cluster 2
Clustering result for the set GeneSet$_5$						
ALL accuracy: 97.87 %—*AML accuracy:* 88 %						
ALL	7	14	14	11	1	0
AML	0	0	3	0	13	9

Overall, the obtained experimental results indicate that using *GeneSet*$_1$ and *GeneSet*$_2$, yields very satisfactory results. The best results were obtained using the union of the genes in *GeneSet*$_1$ and *GeneSet*$_2$. The drawback of this feature selection scheme is that it relies on human expertise (*GeneSet*$_1$) and requires class information (*GeneSet*$_2$) to construct the final dataset (*GeneSet*$_3$). On the other hand, performing unsupervised gene selection using either PCA or UKW results in a lower classification accuracy. The hybridization of the two approaches yielded results comparable to those obtained through the first three gene sets. The main drawback of this approach is that it requires information from all the genes.

5.3. Evaluation of the hybrid approach

The evaluation of the hybrid approach is performed through three publicly available data sets.

- The COLON data set [12] consists of 40 tumor and 22 normal colon tissues. For each sample there exist 2000 gene expression level measurements. The data set is available at http://microarray.princeton.edu/oncology.
- The PROSTATE data set [51] contains 52 prostate tumor samples and 50 nontumor prostate samples. For each sample there exist 6033 gene expression level measurements. It is available at http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.
- The LYMPHOMA dataset [22] that contains 62 samples of the 3 lymphoid malignancies samples types. The samples are measured over 4026 gene expression levels. This dataset is available at http://genome-www.stanford.edu/.

For each dataset, the hybrid approach is being compared with the PCA technique as a dimension reduction method. While the hybrid approach automatically determines the number of reduced dimensions only the screen plot can provide such an information for the PCA technique. Although the scree plots, reported in Fig. 4, provide an indication they are not conclusive. Generally, in all three cases the contributions are relatively low after approximately twenty components.

In our experiments, we tried all available factors for each datasets. The classification accuracy of the UKW clustering result for the three datasets and all the available factors are reported in Fig. 5. For the COLON dataset the best classification accuracy obtained was 80.64% employing 16 factors. For the PROSTATE dataset the best result was 82.35% classification accuracy, using 71 factors. Finally, for the LYMPHOMA dataset the best result was 98.38% classification accuracy using only 3 factors.

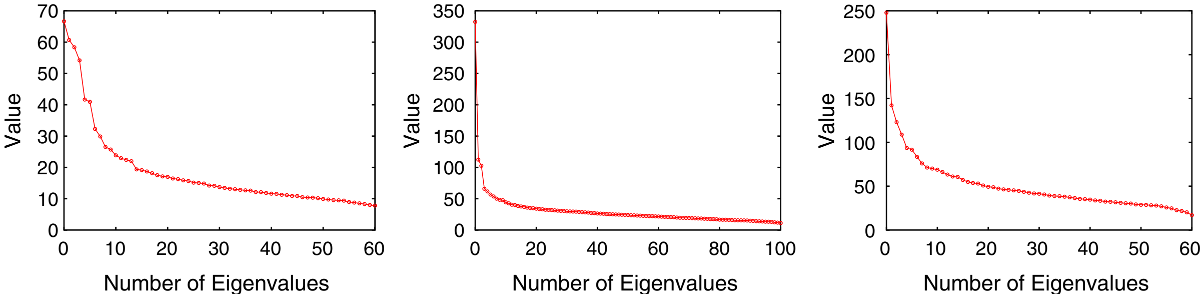

Fig. 4. Plot of the first eigenvalues in decreasing order, for the COLON (left), PROSTATE (middle) and the LYMPHOMA (right) datasets.

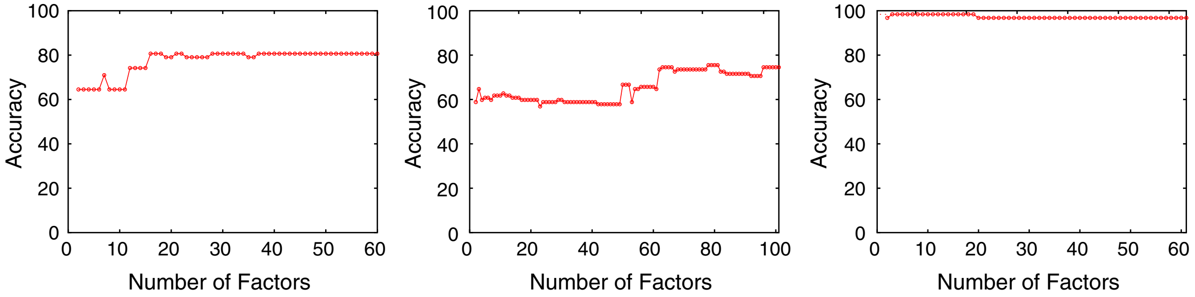

Fig. 5. Classification accuracy for all the factors, for the COLON (left), PROSTATE (middle) and the LYMPHOMA (right) datasets.

Table 6
The performance of the hybrid approach for the COLON, PROSTATE and LYMPHOMA datasets

Dataset	Number of factors used	Classification accuracy (%)
COLON	229	82.25
PROSTATE	84	83.3
LYMPHOMA	103	99.01

The results of the hybrid approach, for the three datasets, are presented in Table 6. As it is evident, the classification accuracy of the resulting partitions increases in all three cases. The high number of factors that the hybrid scheme decides to use, does not impose a problem to the algorithm since they originate in different clusters, and they are not correlated to each other. Furthermore, the additional advantage of the automatic determination of the required factors, exhibits a robust result that is not possible through the PCA technique. The classification accuracies obtained are considered very high, in comparison to other methods [52].

6. Concluding remarks

DNA microarray technologies measure gene expression levels for a very large number of genes covering the entire genome. However, the number of genes is usually very high compared to the number of data samples. In machine learning terminology these data sets are called undersampled, since they have very high dimension and small sample size. To cope with the performance and accuracy problems associated with high dimensionality and noise, the data are transformed, with limited loss of information, to a space of significantly lower dimension containing only the most relevant genes.

Cluster analysis presented here groups leukemia samples into clusters based on similar gene expression microarray data. More specifically, we have applied the unsupervised version of the recently proposed *k*-windows clustering algorithm, since it has already been proved successful in similar settings [39–41]. We have compared our approach against four well-known clustering algorithms and the results were satisfactory.

The first data set used for the experiments was provided by the center of genome research, Whitehead Institute [4]. From the 7129 genes provided, 3 different gene sets were considered. The first set was published in the original paper of Golub et al. [4], while the second gene set was proposed by Thomas et al. [50]. The third gene set was constructed from the union of the two previously mentioned gene sets. The results were evaluated using an independent test set. The clusters discovered using the sets $GeneSet_1$ and $GeneSet_2$ exhibited 73.5% and 91.2% classification success, respectively. However, the best results were obtained using $GeneSet_3$ (94.1%).

However in a practical setting it is not possible to have an a priori class information. To this end, we employed techniques for unsupervised dimension reduction. In detail, we compared the performance of the UKW clustering algorithm against the PCA dimension reduction technique and we proposed a new hybrid system that utilizes both PCA and UKW for the automatic classification of gene expression microarray data sets. The hybrid system was capable of performing dimension reduction and classification, exhibiting high accuracy and robust performance. It is important to note that no class information is used, which implies that the proposed system is best suited for real world gene data sets. We have demonstrated, the performance of the proposed approach in three other datasets. The experiments indicate the hybrid scheme is able to provide results that are comparable with those obtained through supervised approaches [52].

Acknowledgements

The authors would like to thank the four anonymous referees as well as the editors for their constructive comments and useful suggestions that helped to significantly improve this paper. We also thank the European Social Fund, Operational Program for Educational and Vocational Training II (EPEAEK II), and particularly the Program "Pythagoras" for funding this work. This work was also partially supported by the University of Patras Research Committee through a "Karatheodoris" research grant.

References

[1] D. Jiang, C. Tang, A. Zhangi, Cluster analysis for gene expression data: a survey. IEEE Trans. Knowledge Data Eng. 16 (11) (2004) 1370–1386.

[2] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. USA 95 (1998) 14863–14868.

[3] X. Wen, S. Fuhrman, G. Michaels, D. Carr, S. Smith, J. Barker, R. Somogyi, Large-scale temporal gene expression mapping of cns development, Proc. Natl. Acad. Sci. USA 95 (1998) 334–339.

[4] T.R. Golub, D.K. Slomin, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M.L. Loh, J. Downing, M. Caligiuri, C. Bloomfield, E. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.

[5] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, Princeton, 1961.

[6] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, Berlin, 2001.

[7] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, Advances in Knowledge Discovery and Data Mining, MIT Press, Cambridge, 1996.

[8] M.S. Aldenderfer, R.K. Blashfield, Cluster Analysis, Quantitative Applications in the Social Sciences, vol. 44, SAGE Publications, London, 1984.

[9] V. Ramasubramanian, K. Paliwal, Fast *k*-dimensional tree algorithms for nearest neighbor search with application to vector quantization encoding, IEEE Trans. Signal Process. 40 (3) (1992) 518–531.

[10] R.W. Becker, G.V. Lago, A global optimization algorithm, in: Proceedings of the Eighth Allerton Conference on Circuits and Systems Theory, 1970, pp. 3–12.

[11] A. Törn, A. Žilinskas, Global Optimization, Springer, Berlin, 1989.

[12] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array, Proc. Natl. Acad. Sci. USA 96 (12) (1999) 6745–6750.

[13] R. Shamir, R. Sharan, CLICK: a clustering algorithm for gene expression analysis, in: Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB 00), AAAIPress, 2000.

[14] C. Tang, A. Zhang, M. Ramanathan, Espd: a pattern detection model underlying gene expression profiles, Bioinformatics 20 (6) (2004) 829–838.

[15] S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho, G.M. Church, Systematic determination of genetic network architecture, Nature Genet. 22 (1999) 281–285.

[16] E.P. Xing, R.M. Karp, CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts, Bioinformatics Discovery Note 1 (2001) 1–9.

[17] J.A. Hartigan, M.A. Wong, A *k*-means clustering algorithm, Appl. Statist. 28 (1979) 100–108.

[18] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Dordrecht, 1981.

[19] C.H.Q. Ding, Analysis of gene expression profiles: class discovery and leaf ordering, in: Sixth Annual International Conference on Computational Biology, ACM Press, New York, 2002, pp. 127–136.

[20] M.N. Vrahatis, B. Boutsinas, P. Alevizos, G. Pavlides, The new *k*-windows algorithm for improving the *k*-means clustering algorithm, J. Complexity 18 (2002) 375–391.

[21] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.

[22] A.A. Alizadeh, et al., Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling, Nature 403 (6769) (2000) 503–511.

[23] C.M. Perou, S.S. Jeffrey, M. Van de Rijn, C.A. Rees, M.B. Eisen, D.T. Ross, A. Pergamenschikov, C.F. Williams, S.X. Zhu, D. Lashkari, J.C. Lee, D. Shalon, P.O. Brown, D. Botstein, Distinctive gene expression patterns in human mammary epithelial cells and breast cancers, Proc. Natl. Acad. Sci. USA 96 (1999) 9212–9217.

[24] P. Tamayo, D. Slonim, Q. Mesirov, J. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, T.R. Golub, Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation, Proc. Natl. Acad. Sci. USA 96 (1999) 2907–2912.

[25] O. Alter, P.O. Brown, D. Bostein, Singular value decomposition for genome-wide expression data processing and modeling, Proc. Natl. Acad. Sci. USA 97 (18) (2000) 10101–10106.

[26] Z. Szallasi, R. Somogyi, Genetic network analysis—the millennium opening version, in: Pacific Symposium of BioComputing Tutorial, 2001.

[27] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: International Conference on Machine Learning, 1994, pp. 121–129.

[28] C.C. Aggarwal, J.L. Wolf, P.S. Yu, C. Procopiuc, J.S. Park, Fast algorithms for projected clustering, in: 1999 ACM SIGMOD International Conference on Management of Data, ACM Press, New York, 1999, pp. 61–72.

[29] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in: 1998 ACM SIGMOD International Conference on Management of Data, ACM Press, New York, 1998, pp. 94–105.

[30] S.C. Madeira, A.L. Oliveira, Biclustering algorithms for biological data analysis: a survey, IEEE/ACM Trans. Comput. Biol. Bioinformatics 1 (2004) 24–45.

[31] D.K. Tasoulis, M.N. Vrahatis, Unsupervised distributed clustering, in: Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks, Innsbruck, Austria, 2004, pp. 347–351.

[32] J. Sander, M. Ester, H.-P. Kriegel, X. Xu, Density-based clustering in spatial databases: the algorithm gdbscan and its applications, Data Mining Knowledge Discovery 2 (2) (1998) 169–194.

[33] D. Boley, Principal direction divisive partitioning, Data Mining and Knowledge Discovery 2 (4) (1998) 325–344.

[34] B. Fritzke, Growing cell structures a self-organizing network for unsupervised and supervised learning, Neural Networks 7 (9) (1994) 1441–1460.

[35] F. Preparata, M. Shamos, Computational Geometry, Springer, New York, Berlin, 1985.

[36] P. Alevizos, An algorithm for orthogonal range search in $d \geqslant 3$ dimensions, in: Proceedings of the 14th European Workshop on Computational Geometry, Barcelona, 1998.

[37] J.L. Bentley, H.A. Maurer, Efficient worst-case data structures for range searching, Acta Inform. 13 (1980) 1551–1568.

[38] B. Chazelle, Filtering search: a new approach to query-answering, SIAM J. Comput. 15 (3) (1986) 703–724.

[39] V.P. Plagianakos, D.K. Tasoulis, M.N. Vrahatis, Hybrid dimension reduction approach for gene expression data classification, in: International Joint Conference on Neural Networks 2005 (IJCNN 2005), Post-Conference Workshop on Computational Intelligence Approaches for the Analysis of Bioinformatics Data, 2005.

[40] D.K. Tasoulis, V.P. Plagianakos, M.N. Vrahatis, Unsupervised cluster analysis in bioinformatics, in: Fourth European Symposium on "Biomedical Engineering", 2004.

[41] D.K. Tasoulis, V.P. Plagianakos, M.N. Vrahatis, Unsupervised clustering of bioinformatics data, in: European Symposium on Intelligent Technologies, Hybrid Systems and Their Implementation on Smart Adaptive Systems, Eunite, 2004, pp. 47–53.

[42] G.D. Magoulas, V.P. Plagianakos, D.K. Tasoulis, M.N. Vrahatis, Tumor detection in colonoscopy using the unsupervised *k*-windows clustering algorithm and neural networks, in: Fourth European Symposium on "Biomedical Engineering", 2004.

[43] D.K. Tasoulis, L. Vladutu, V.P. Plagianakos, A. Bezerianos, M.N. Vrahatis, On-line neural network training for automatic ischemia episode detection, in: L. Rutkowski, J.H. Siekmann, R. Tadeusiewicz, L.A. Zadeh (Eds.), Lecture Notes in Computer Science, vol. 2070, Springer, Berlin, 2003, pp. 1062–1068.

[44] N.G. Pavlidis, D.K. Tasoulis, M.N. Vrahatis, Financial forecasting through unsupervised clustering and evolutionary trained neural networks, in: Congress on Evolutionary Computation, Canberra, Australia, 2003.

[45] M. Rigou, S. Sirmakessis, A. Tsakalidis, A computational geometry approach to web personalization, in: IEEE International Conference on E-Commerce Technology (CEC'04), San Diego, California, 2004, pp. 377–380.

[46] D.K. Tasoulis, M.N. Vrahatis, Novel approaches to unsupervised clustering through the *k*-windows algorithm, in: Knowledge Mining, Studies in Fuzziness and Soft Computing, 2005.

[47] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases, in: Second International Conference on Data Mining KDD-96, Portland, Oregon, 1996, pp. 226–231.

[48] D. Zeimpekis, E. Gallopoulos, PDDP(l): towards a flexing principal direction divisive partitioning clustering algorithms, in: D. Boley, I. Dhillon, J. Ghosh, J. Kogan (Eds.), Proceedings of the IEEE ICDM '03 Workshop on Clustering Large Data Sets, Melbourne, Florida, 2003, pp. 26–35.

[49] T. Kohonen, Self-Organized Maps, Springer, New York, Berlin, 1997.

[50] J.G. Thomas, J.M. Olson, S.J. Tapscott, L.P. Zhao, An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles, Genome Res. 11 (2001) 1227–1236.

[51] D. Singh, et al., Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209.

[52] J. Ye, T. Li, T. Xiong, R. Janardan, Using uncorrelated discriminant analysis for tissue classification with gene expression data, IEEE ACM Trans. Comput. Biol. Bioinformatics 1 (4) (2004) 181–190.