Microarray *Leukemia* Gene Data Clustering by Means of Generalized Self-organizing Neural Networks with Evolving Tree-Like Structures

Marian B. Gorzałczany(✉), Jakub Piekoszewski, and Filip Rudziński

Department of Electrical and Computer Engineering
Kielce University of Technology
Al. 1000-lecia P.P. 7, 25-314 Kielce, Poland
{m.b.gorzalczany,j.piekoszewski,f.rudzinski}@tu.kielce.pl

Abstract. The paper presents the application of our clustering technique based on generalized self-organizing neural networks with evolving tree-like structures to complex cluster-analysis problems including, in particular, the sample-based and gene-based clusterings of microarray *Leukemia* gene data set. Our approach works in a fully unsupervised way, i.e., without the necessity to predefine the number of clusters and using unlabelled data. It is particularly important in the gene-based clustering of microarray data for which the number of gene clusters is unknown in advance. In the sample-based clustering of the *Leukemia* data set, our approach gives better results than those reported in the literature and obtained using a method that requires the cluster number to be defined in advance. In the gene-based clustering of the considered data, our approach generates clusters that are easily divisible into subclusters related to particular sample classes. It corresponds, in a way, to subspace clustering that is highly desirable in microarray data analysis.

Keywords: Microarray cancer gene data · Gene expression data clustering · Generalized self-organizing neural networks with evolving tree-like structures · Cluster analysis · Unsupervised learning

1 Introduction

Microarray technologies have been playing an increasingly important role in genomic research (see, e.g., [11]). They make possible to measure the level of expression or activity of tens of thousands of genes simultaneously in different experimental samples (in general, under different experimental conditions). The resulting data are usually represented in the form of the so-called gene expression data matrix. Its rows represent genes and its column - various specific samples. Thus, each cell of the matrix represents a numeric level of the expression of a given gene in a given sample. One of the essential steps in interpreting the meaning of such immense amount of biological information is to discover clusters of genes that manifest similar expression patterns (coexpressed and possibly

L. Rutkowski et al. (Eds.): ICAISC 2015, Part I, LNAI 9119, pp. 15–25, 2015.
DOI: 10.1007/978-3-319-19324-3_2

coregulated genes). In general, however, it is meaningful to cluster both genes and samples into homogeneous groups [8].

This paper presents both gene-based and sample-based clusterings of *Leukemia* human cancer microarray data set by means of our original approach that employs generalized self-organizing neural networks (SONNs) with evolving tree-like structure and with dynamically defined neighborhood (SONNs with DDN for short) presented in [5]. It is worth stressing that our approach works in a fully unsupervised way, i.e., using unlabelled data and without a predefined number of clusters which is particularly important in the gene-based clustering of microarray data where the number of gene clusters is unknown in advance. First, the clustering process using SONNs with DDN is outlined (its more detailed presentation can be found in [5]). Then, the operation of the proposed networks on two- and three-dimensional benchmark data sets [12] that contain data groups of various shapes and densities is shown. Finally, their application to the clustering of the afore-mentioned human cancer microarray data set, i.e., *Lukemia* [4] is presented and compared with an alternative solution.

2 Generalized SONNs with DDN for Data Clustering - An Outline [5]

In the course of learning that controls the evolution of tree-like structures of generalized SONNs with DDN, they are able to: a) automatically adjust the number of neurons in the network by removing low-active neurons from the network and adding new neurons in the areas of existing high-active neurons in order to take over some of their activities, b) disconnect the tree-like structures into subnetworks, and c) reconnect some of the subnetworks preserving the no-loop spanning-tree properties. These mechanisms enable them to detect data clusters of various shapes and densities including both volumetric as well as thin, shell, piece-wise linear, polygonal, etc. kinds of clusters. Each detected cluster is represented by a single disconnected subnetwork. Therefore, the number of automatically generated subnetworks is equal to the number of clusters. Moreover, our approach also generates a multi-point prototype for each cluster; that prototype is represented by the collection of neurons belonging to a given subnetwork. Such prototypes can be directly used in clustering/classification tasks by employing the well-known nearest multi-prototype approach [2], [1]. The application of our approach to the clustering of several synthetic and real (coming from the UCI repository [10]) benchmark data sets has been presented in [5]. Our approach is a generalization of our earlier solutions to automatic determination of the number of clusters and their prototypes in data sets [6], [7].

The point of departure for the idea of the generalized SONNs (see [5] for details) is the conventional SONN with one-dimentional neighborhood (i.e., the neuron chain) with n inputs $x_1, x_2, \ldots, x_n$ and m neurons with outputs $y_1, y_2, \ldots, y_m$, respectively. $y_j = \sum_{i=1}^{n} w_{ji} x_i$, $j = 1, 2, \ldots, m$ and w_{ji} are weights connecting the i-th input of the network with the output of the j-th neuron. Using vector notation ($\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$, $\boldsymbol{w}_j = (w_{j1}, w_{j2}, \ldots, w_{jn})^T$),

$y_j = \boldsymbol{w}_j^T \boldsymbol{x}$. The learning data set contains L input vectors $\boldsymbol{x}_l$ ($l = 1, 2, \ldots, L$). In the first stage of any Winner-Takes-Most (WTM) learning algorithm that can be applied to the considered network, the neuron $j_{\boldsymbol{x}}$ winning in the competition of neurons when learning vector $\boldsymbol{x}_l$ is presented to the network must be determined in the following way (assuming the normalization of learning vectors):

$$d(\boldsymbol{x}_l, \boldsymbol{w}_{j_x}) = \min_{j=1,2,\ldots,m} d(\boldsymbol{x}_l, \boldsymbol{w}_j), \qquad (1)$$

where $d(\boldsymbol{x}_l, \boldsymbol{w}_j)$ is a distance measure between $\boldsymbol{x}_l$ and $\boldsymbol{w}_j$; throughout this paper, the Euclidean distance measure $d_E(\boldsymbol{x}_l, \boldsymbol{w}_j) = \sqrt{\sum_{i=1}^{n} (x_{li} - w_{ji})^2}$ will be applied. The WTM learning rule is following:

$$\boldsymbol{w}_j(k+1) = \boldsymbol{w}_j(k) + \eta_j(k) N(j, j_{\boldsymbol{x}}, k)[\boldsymbol{x}(k) - \boldsymbol{w}_j(k)], \qquad (2)$$

where k is the iteration number, $\eta_j(k)$ is the learning coefficient, and $N(j, j_{\boldsymbol{x}}, k)$ is the neighborhood function:

$$N(j, j_{\boldsymbol{x}}, k) = e^{-\frac{d_{tpl}^2(j, j_x)}{2\lambda^2(k)}} \qquad (3)$$

with $\lambda(k)$ being the radius of the neighborhood and $d_{tpl}(j, j_{\boldsymbol{x}})$ representing the topological distance between the neurons no. $j_{\boldsymbol{x}}$ and no. j. The neighborhood of a given neuron in the tree-like topology of our generalized SONNs is defined along the arcs (being the pieces of the conventional SONN with one-dimensional neighborhood) emanating from that neuron as shown in Fig. 1 (see [5] for details). Therefore, $d_{tpl}(j, j_{\boldsymbol{x}}) = 1$ for all j-th neurons being direct neighbors of the $j_{\boldsymbol{x}}$-th one as illustrated in Fig. 1. In turn, $d_{tpl}(j, j_{\boldsymbol{x}}) = 2$ for all j-th neurons being second along all paths starting at the $j_{\boldsymbol{x}}$-th one (see Fig. 1), etc.

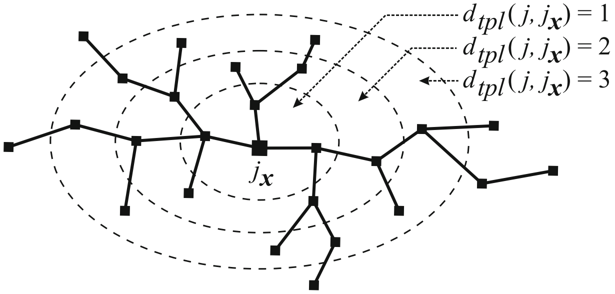
$d_{tpl}(j, j_{\boldsymbol{x}}) = 1$
$d_{tpl}(j, j_{\boldsymbol{x}}) = 2$
$d_{tpl}(j, j_{\boldsymbol{x}}) = 3$
$j_{\boldsymbol{x}}$

Fig. 1. Examples of neighborhood of the j_x-th neuron [5]

In order to implement three mechanisms, listed as a), b), and c) in the first paragraph of this section, five operations are activated (under some conditions) after each learning epoch.

Operation 1 (first component of mechanism a)) - the removal of single, low active neurons: The neuron no. j_r is removed from the network (preserving the network continuity - see [5] for details) if its activity - measured by the number of its wins win_{j_r} - is below an assumed level win_{min}, i.e., $win_{j_r} < win_{min}$. win_{min} is experimentally selected parameter (usually, $win_{min} \in \{2, 3, 4\}$).

Operation 2 (mechanism b)) - the disconnection of the network (subnetwork) into two subnetworks: The disconnection of two neighboring neurons j_1 and j_2 takes place if the following condition is fulfilled: $d_E(\boldsymbol{w}_{j_1}, \boldsymbol{w}_{j_2}) > d_{coef} d_{E,avr}$ where $d_{E,avr} = \frac{1}{P}\sum_{p=1}^{P} d_{E,p}$ is the average distance between two neighboring neurons for all pairs p, $p = 1, 2, \ldots, P$, of such neurons, and d_{coef} is experimentally selected parameter (a distance coefficient) governing the disconnection operation (usually, $d_{coef} \in [3, 4]$).

Operation 3 (second component of mechanism a)) - the removal of small-size subnetworks: A subnetwork of m_s neurons is removed from the system if $m_s < m_{s,min}$, where $m_{s,min}$ is experimentally selected parameter (usually, $m_{s,min} \in \{3, 4\}$).

Operation 4 (third component of mechanism a)) - the insertion of additional neurons into the neighborhood of high-active neurons in order to take over some of their activities. *Case 4a*: A new neuron, labelled as (*new*), is inserted between two neighboring and high-active neurons j_1 and j_2 (i.e., their numbers of wins win_{j_1} and win_{j_2} are above an assumed level win_{max}: $win_{j_1}, win_{j_2} > win_{max}$). win_{max} is experimentally selected parameter (usually $win_{max} \in \{2, 3, 4\}$ and $win_{max} \geq win_{min}$, where win_{min} is defined in Operation 1). The weight vector $\boldsymbol{w}_{(new)}$ of the new neuron is calculated as follows: $\boldsymbol{w}_{(new)} = \frac{\boldsymbol{w}_{j_1} + \boldsymbol{w}_{j_2}}{2}$. *Case 4b*: A new neuron (*new*) is inserted in the neighborhood of high-active neuron j_1 surrounded by less-active neighbors (i.e., $win_{j_1} > win_{max}$ and $win_j < win_{max}$ for j such that $d_{tpl}(j, j_1) = 1$). The weight vector $\boldsymbol{w}_{(new)} = [w_{(new)1}, w_{(new)2}, \ldots, w_{(new)n}]^T$ is calculated as follows: $w_{(new)i} = w_{j_1 i}(1 + \xi_i)$, $i = 1, 2, \ldots, n$, where ξ_i is a random number from the interval $[-0.01, 0.01]$ (see [5] for details).

Operation 5 (mechanism c)) - the reconnection of two selected subnetworks: Two subnetworks S_1 and S_2 are reconnected by introducing topological connection between neurons j_1 and j_2 ($j_1 \in S_1$, $j_2 \in S_2$) after fulfilling condition $d_E(\boldsymbol{w}_{j_1}, \boldsymbol{w}_{j_2}) < d_{coef} \frac{d_{E,avr_{S_1}} + d_{E,avr_{S_2}}}{2}$. $d_E(\boldsymbol{w}_{j_1}, \boldsymbol{w}_{j_2})$ and d_{coef} are the same as in Operation 2. $d_{E,avr_{S_1}}$ and $d_{E,avr_{S_2}}$ are calculated for subnetworks S_1 and S_2, respectively, in the same way as $d_{E,avr}$ is calculated in Operation 2 for the considered network.

According to Kohonen's comments [9], the selection of learning parameters is mainly based on experimental results taking into account that the learning coefficient $\eta(k)$ and the neighborhood radius $\lambda(k)$ should be some monotonically decreasing functions of time ($\lambda(k)$ can also be constant in time). Based on that, in the experiments presented below, the learning parameters are defined as follows: $\eta_j(k) = \eta(k)$ of (2) is linearly decreasing over the learning horizon (which includes 10.000 epochs) from $7 \cdot 10^{-4}$ to 10^{-6}, $\lambda(k) = \lambda$ of (3) is equal to 2, the initial number of neurons in the network is equal to 2, $win_{min} = 2$, $win_{max} = 4$, $m_{s,min} = 2$, and $d_{coef} = 4$.

3 Clustering of Two- and Three-Dimensional Benchmark Data Sets

The so-called Fundamental Clustering Problem Suite (FCPS) [12] is a collection of benchmark data sets that, for different reasons, pose difficult problems to clustering algorithms. We selected two benchmark sets from FCPS, one two-dimensional (*Lsun* data set) and one three-dimensional (*Atom* data set) to illustrate the performance of our approach. According to [12], main clustering problems in *Lsun* are different variances in clusters and in *Atom* - linearly non-separable data of different densities and variances.

Figs. 2 and 3 present the performance of our clustering technique applied to particular data sets. The figures are arranged in the same way, i.e., parts a) of them represent the data, parts b), c), d), e), and f) show the evolution of the tree-like structures of the generalized SONNs at different stages of the learning process, and parts g) and h) - the plots of the number of neurons (g) and the number of subnetworks (clusters) (h) vs. epoch number. It can be seen that our approach, in an automatic way, increases the number of neurons in particular networks (starting from the initial numbers of two neurons) and detects the correct numbers of data clusters in both sets by disconnecting the tree-like structures of the generalized SONNs into appropriate number of subnetworks.

4 Clustering of *Leukemia* Cancer Microarray Data Set

The performance of our approach will now be validated in the clustering of a data set coming from microarray experiments. The benchmark human cancer microarray data set, i.e., *Leukemia* [4] is considered. It is typical for microarray gene expression data sets that they contain thousands of original genes (in our case, 7.129) and a small number of samples (in our case, 72 including two classes called ALL with 47 samples and AML with 25 samples). Additionally, many of the original genes are noisy and redundant. In order to filter out such genes, various preprocessing methods are applied (see, e.g., [3] for details) yielding reduced subset of genes (3.571 for *Leukemia* data set) that are used in experiments. As already mentioned in the introduction (see also [8]), in gene expression data analysis it is meaningful to consider both the sample-based and gene-based clusterings. In the first case, the samples are the objects and the genes are the features, whereas in the second case it is quite opposite. Due to a very small number of data samples, the parameter win_{max} that (together with win_{min}) controls the overall number of neurons in the network is reduced to win_{min}, i.e., $win_{max} = win_{min} = 2$. For the same reason, the distance coefficient is slightly reduced ($d_{coef} = 3$). The remaining parameters are unchanged.

Figs. 4, 5, 6, and 7 present the performance of our clustering algorithm applied to the considered data set. Figs. 4 and 5 show the plots of the number of neurons and the number of subnetworks (clusters) vs. epoch number for the sample-based and gene-based clusterings, respectively. As far as the sample-based clustering is concerned, the number of clusters in data set and the cluster assignments

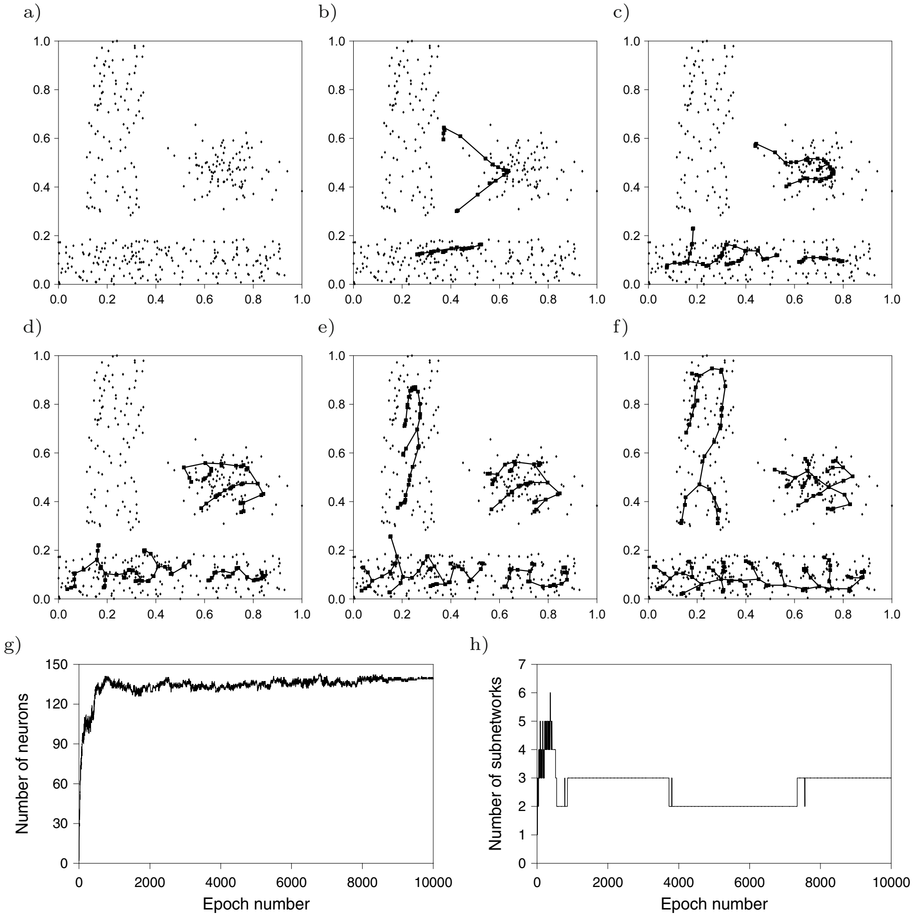

Fig. 2. *Lsun* data set (a) and the evolution of the generalized SONN in it in learning epochs: b) no. 5, c) no. 50, d) no. 100, e) no. 500, and f) no. 10 000 (end of learning), as well as plots of the number of neurons (g) and the number of subnetworks (clusters) (h) vs. epoch number

of particular data samples are known. Therefore, a direct verification of the obtained results is possible. However, it should be stressed that our approach does not use the knowledge on the cluster assignments and the cluster number during its operation. That knowledge is used after the completion of the learning to evaluate the obtained results. Fig. 4b shows that our approach detects the correct (equal to 2) number of sample clusters in the considered data set. The percentage of correct decisions, equal to 98.6%, regarding the cluster assignments of particular data samples is higher than in the case of an alternative approach presented in [3] (93.14%) which additionally requires the cluster number to be defined in advance.

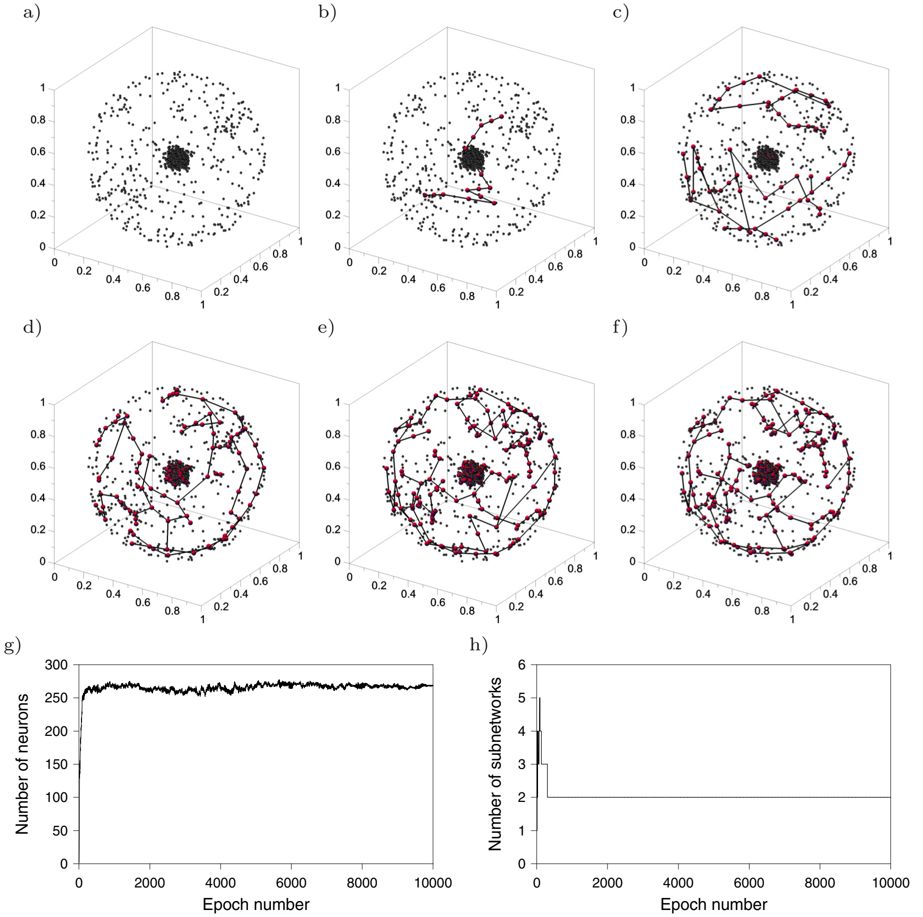

Fig. 3. *Atom* data set (a) and the evolution of the generalized SONN in it in learning epochs: b) no. 5, c) no. 50, d) no. 100, e) no. 500, and f) no. 10 000 (end of learning), as well as plots of the number of neurons (g) and the number of subnetworks (clusters) (h) vs. epoch number

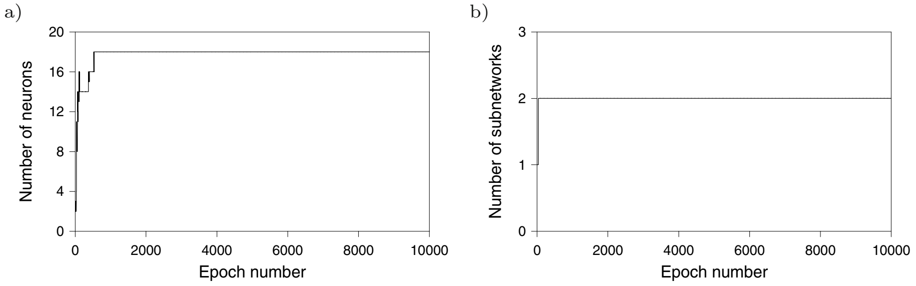

Fig. 4. Plots of the number of neurons (a) and the number of subnetworks (clusters) (b) vs. epoch number for the sample-based clustering of the *Leukemia* data set

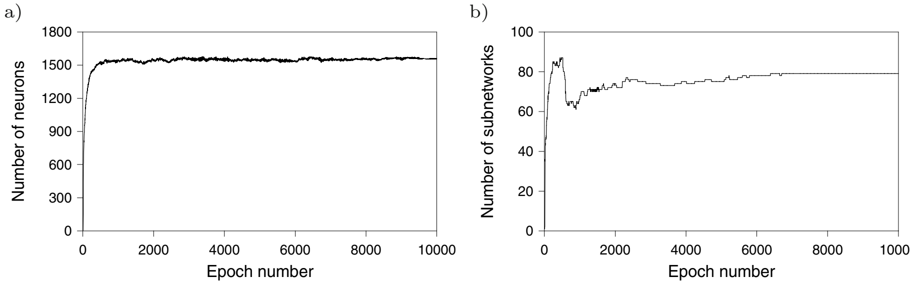

Fig. 5. Plots of the number of neurons (a) and the number of subnetworks (clusters) (b) vs. epoch number for the gene-based clustering of the *Leukemia* data set

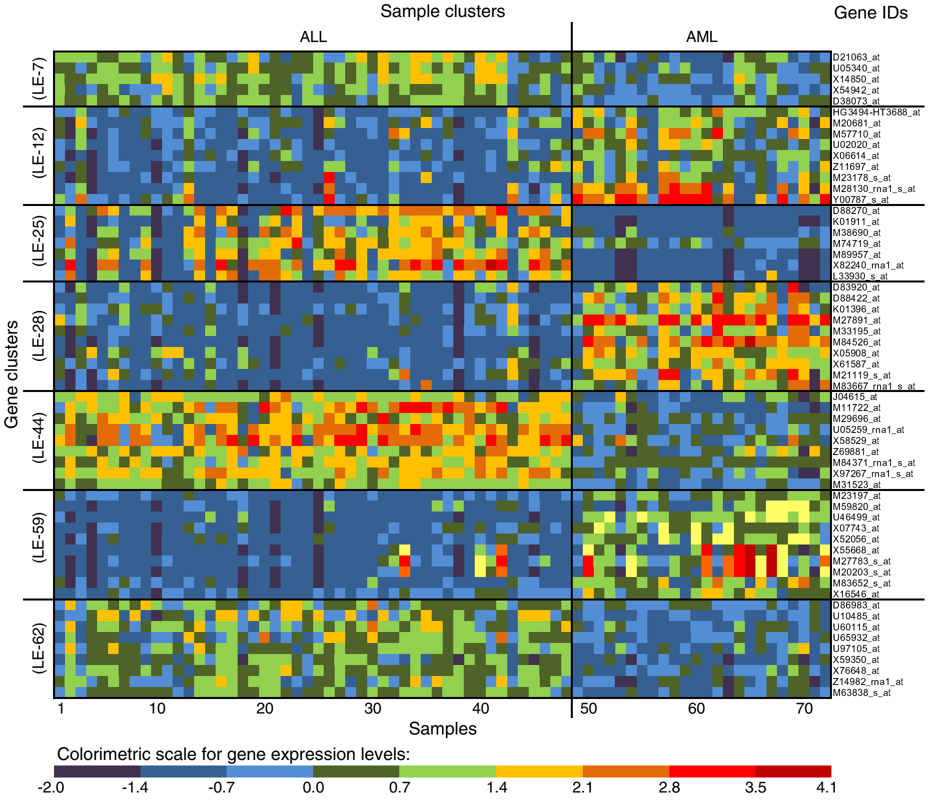

Fig. 6. Exemplary gene clusters in the *Leukemia* data set

As far as the gene-based clustering (or, to be more specific, the clustering of gene expression levels) is concerned, our approach detects 79 gene clusters (see Fig. 5b). Fig. 6 presents the pseudocolor image of some of them. Each of those clusters can be easily divided into two subclusters related to ALL and AML samples. Therefore, the results generated by our approach correspond, in a way,

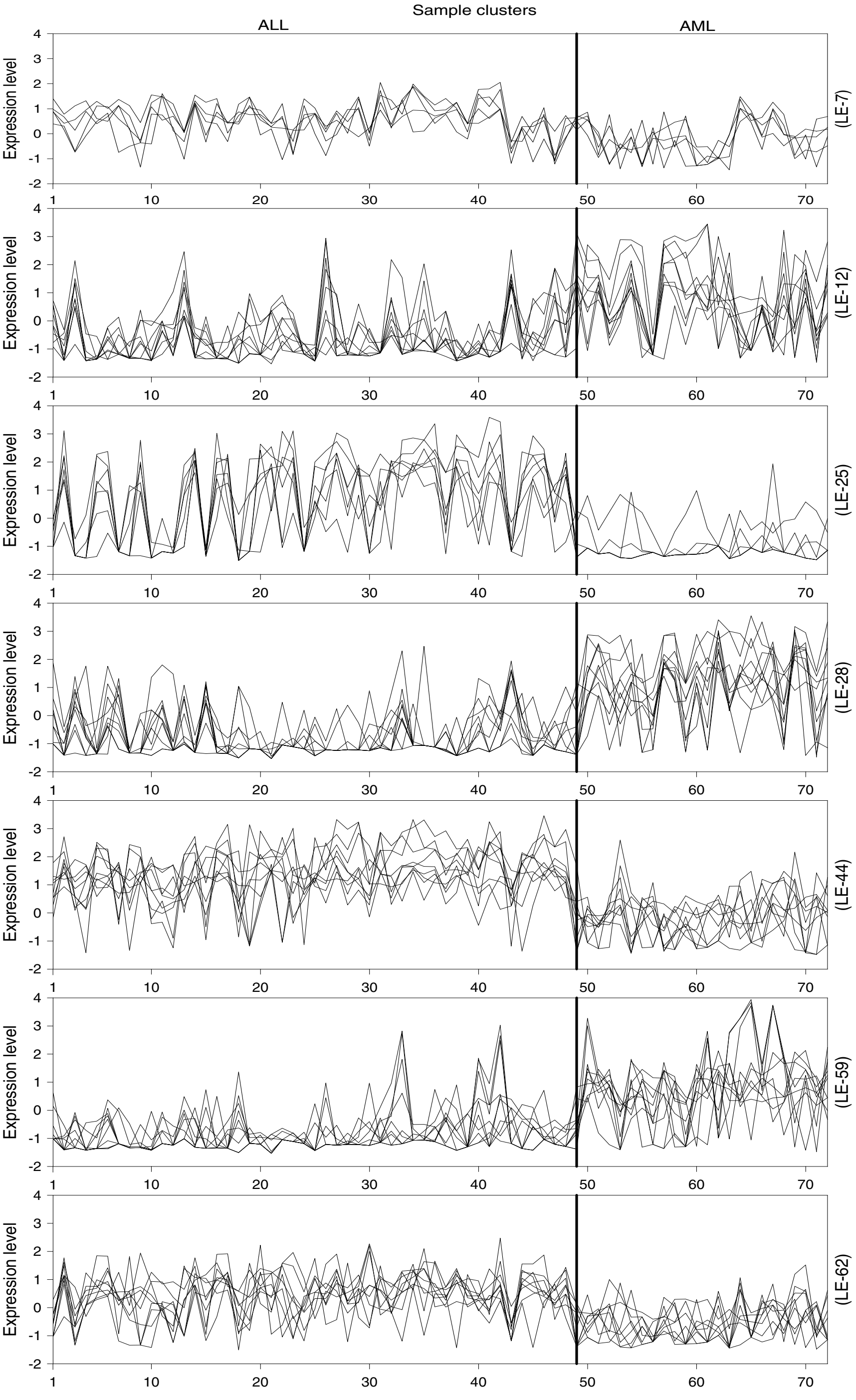

Fig. 7. Plots of the expression levels of all genes in each gene cluster of Fig. 6

to the so-called subspace clustering which is highly desirable in microarray data analysis (see discussion in [8]). The subspace clustering captures clusters created by a subset of genes across a subset of data samples (in our case, ALL and AML samples separately). Fig. 7 shows the plots of the expression levels of all genes in the gene clusters of Fig. 6 confirming the compactness of particular clusters (as well as ALL and AML related subclusters). The pseudocolor image of particular clusters of Fig. 6 (supported by Fig. 7) can be used in a deeper genetics-based discussion of the obtained results which is - due to the limited space - not possible here. The interpretation of the obtained gene clusters is possible on the basis of statistical analysis performed with the use of specialized and dedicated software. In our experiments, we use two publicly available functional profiling tools, i.e., the DAVID (Database for Annotation, Visualization and Integrated Discovery) software, available on the server of Laboratory of Immunopathogenesis and Bioinformatics, National Cancer Institute at Frederick, USA (http://david.abcc.ncifcrf.gov) and the 'g:Profiler' software, available on the server of Institute of Computer Science, University of Tartu, Estonia (http://biit.cs.ut.ee/gprofiler). We can only mention here that, for instance, in the case of 'LE-7' gene cluster (see Fig. 6), both tools indicate that all the genes collected in the cluster are responsible for one biological process named 'cell cycle'. In turn, in the case of 'LE-28' gene cluster, 7 out of 10 genes are responsible for biological process named 'defense response', etc.

5 Conclusions

The paper presents the application of our clustering technique based on the generalized SONNs with evolving tree-like structures to complex cluster-analysis problems including, in particular, the sample-based and gene-based clusterings of microarray *Leukemia* gene data set. Our approach works in a fully unsupervised way, i.e., without the necessity to predefine the number of clusters and using unlabelled data. It is particularly important in the gene-based clustering of microarray data for which the number of gene clusters is unknown in advance. In a given data set, our approach, in automatic way, detects the number of clusters (equal to the number of disconnected subnetworks) and generates multi-prototypes for them (represented by neurons in particular subnetworks). It is performed by the implementation of automatic adjustment of the number of neurons in the network as well as the disconnection and reconnection mechanisms of the tree-like structures of the network during the dynamic learning process. It is worth stressing that the same set of experimentally selected parameters that control the operation of our clustering technique (see the last paragraph of Section 2 of the paper) gives very good clustering results for completely different types of data sets such as FCPS benchmarks and microarray data. It shows, in a way, the low sensitivity of our approach in regard to those parameters. It is also worth emphasizing that in the sample-based clustering of the *Leukemia* data set our approach gives much higher percentage of correct decisions than the alternative technique of [3] which additionally requires the cluster number to be defined in advance. Moreover, our approach exhibits also interesting features as far as the gene-based clustering of the *Leukemia* data set is concerned. Namely, it generates clusters that are easily divisible into subclusters related to particular sample classes; it, in a way, corresponds to subspace clustering which is highly desirable in microarray data analysis [8].

References

1. Bezdek, J.C., Keller, J., Krisnapuram, R., Pal, N.R.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Springer Science & Business Media, New York (2005)
2. Bezdek, J.C., Reichherzer, T.R., Lim, G.S., Attikiouzel, Y.: Multiple-prototype classifier design. IEEE Trans. Systems, Man and Cybernetics, Part C 28(1), 67–79 (1998)
3. Cho, H., Dhillon, I.S.: Coclustering of human cancer microarrays using minimum sum-squared residue coclustering. IEEE/ACM Trans. Computational Biology and Bioinformatics 5(3), 385–400 (2008)
4. Golub, T.R., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537 (1999)
5. Gorzałczany, M.B., Piekoszewski, J., Rudziński, F.: Generalized tree-like self-organizing neural networks with dynamically defined neighborhood for cluster analysis. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2014, Part II. LNCS (LNAI), vol. 8468, pp. 713–725. Springer, Heidelberg (2014)
6. Gorzałczany, M.B., Rudziński, F.: Cluster analysis via dynamic self-organizing neural networks. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 593–602. Springer, Heidelberg (2006)
7. Gorzałczany, M.B., Rudziński, F.: WWW-newsgroup-document clustering by means of dynamic self-organizing neural networks. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 40–51. Springer, Heidelberg (2008)
8. Jiang, D., Tang, C., Zhang, A.: Cluster analysis for gene expression data: a survey. IEEE Trans. Knowledge and Data Engineering 16(11), 1370–1386 (2004)
9. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Berlin (2001)
10. Machine Learning Database Repository, University of California at Irvine, ftp.ics.uci.edu
11. Schena, M.: Microarray Analysis. John Wiley & Sons (2003)
12. Ultsch, A.: Clustering with SOM: U*C. In: Proc. Workshop on Self-Organizing Maps, Paris, France, pp. 75–82 (2005)