Gene expression data clustering using tree-like SOMs with evolving splitting-merging structures

Marian B. Gorzałczany, Filip Rudziński, and Jakub Piekoszewski
Department of Electrical and Computer Engineering
Kielce University of Technology
Al. 1000-lecia P.P. 7, 25-314 Kielce, Poland
Email: {m.b.gorzalczany, f.rudzinski, j.piekoszewski}@tu.kielce.pl.

***Abstract*—The paper presents an application of our clustering technique using generalized tree-like SOMs with evolving splitting-merging structures to complex clustering tasks, including, in particular, the sample-based and gene-based clustering of the *Lymphoma* human cancer microarray data set. It is worth emphasizing that our approach works in a fully unsupervised way, i.e., using unlabelled data and without the necessity to predefine the number of clusters. 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 *Lymphoma* data set, our approach gives better results than those reported in the literature (some of alternative methods require, additionally, the cluster number to be defined in advance). In the gene-based clustering of the considered microarray data, out approach generates clusters that are easily divisible into subclusters related to particular sample classes. In some way, it corresponds to subspace clustering that is highly desirable in microarray data analysis.**

I. Introduction

Gene expression, in general, is the process by which genetic information is used to synthesize functional gene products. The traditional approach to genomic research was focused on the local examination of data on single genes. Presently, microarray technologies (see, e.g., [1]) make it possible to monitor and measure the level of expression of tens of thousands of genes simultaneously in different experimental samples, or in general, under different experimental conditions. The microarray data are usually represented by a matrix (referred to as the gene expression matrix) with rows representing genes and columns corresponding to various specific experimental conditions (usually different samples but also different time points or different organisms can be considered). Hence, each entry of the matrix contains a numerical representation of the expression of a particular gene under a given experimental condition (e.g., in a given sample). An interpretation of the meaning of such an immense amount of biological information poses a serious challenge nowadays. One of the essential steps in addressing that problem is to discover clusters of genes manifesting similar expression patterns (i.e., coexpressed and possibly coregulated genes), keeping in mind that it is meaningful to cluster both genes and experimental conditions (e.g., samples) [2].

We first briefly present a general concept and implementation of our data clustering (or cluster analysis) technique based on tree-like SOMs with evolving splitting-merging structures (see also [3]–[5]). It is worth emphasizing that our approach works in a fully unsupervised way, i.e., using unlabelled data and without a predefined number of clusters. It is particularly important in the gene-based clustering of microarray data where the number of gene clusters is unknown in advance. Then, the operation of the proposed approach is illustrated on selected two- and three-dimensional benchmark data sets [6] containing data groups of various shapes and densities. Finally, the application of our approach to both gene-based and sample-based clustering of *Lymphoma* human cancer microarray data set is presented and compared with alternative solutions.

II. A general concept and implementation of a clustering technique based on tree-like SOMs with evolving splitting-merging structures [3]–[5]

Original SOMs [7], in general, are used to visually display topological structures of high dimensional data in lower, usually two- or three-dimensional space rather than for clustering, i.e., partitioning of these data into groups [8]. However, the proposed generalized SOMs with evolving tree-like structures and structure splitting and merging mechanisms are equipped with both data-dimensionality reduction and data-segmentation capabilities. The evolution of the tree-like structures of the considered networks takes place during the learning process and is controlled by three mechanisms: a) automatic adjustment of 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) disconnection of the tree-like structures into subnetworks, and c) reconnection of some of the subnetworks preserving the no-loop spanning-tree properties. Such structure-evolution mechanisms enable the networks to detect data clusters of virtually any shape and density including volumetric as well as thin, shell, piece-wise linear, polygonal, etc. kinds of clusters. Each detected cluster is represented by a single disconnected subnetwork. Hence, the number of automatically generated subnetworks is equal to the number of clusters. Moreover, our approach also generates a multi-point prototype (represented by the collection of neurons belonging to a given subnetwork) for each cluster. Such prototypes can be directly used in clustering/classification

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

tasks by employing the well-known nearest multi-prototype algorithm [9]. The proposed solution is a generalization of our earlier approaches to automatic determination of the cluster numbers and cluster prototypes in data sets [10]–[13].

In order to implement the afore outlined concept of the data clustering, we first consider the conventional SOM with one-dimensional neighborhood (SOM with 1DN), i.e., the neuron chain. Let's assume that the network has n inputs (features, attributes) $x_1, x_2, \ldots, x_n$ and consists of m neurons; their outputs are $y_1, y_2, \ldots, y_m$, where $y_j = \sum_{i=1}^{n} w_{ji} x_i$, $j = 1, 2, \ldots, m$ 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 consists of 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 used in the learning process of the considered network, the neuron $j_{\boldsymbol{x}}$, which wins in competition of neurons when the learning vector $\boldsymbol{x}_l$ is presented to the network must be determined. Assuming that the normalization of learning vectors is performed, the winning neuron $j_{\boldsymbol{x}}$ is selected in the following way:

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

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 formulated as follows:

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

where k is the iteration number, $\eta_j(k)$ is the learning coefficient, and $N(j, j_{\boldsymbol{x}}, k)$ is the neighborhood function of the $j_{\boldsymbol{x}}$-th winning neuron. Most often the Gaussian-type neighborhood functions are used, i.e.:

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

where $\lambda(k)$ is the neighborhood radius and $d_{tpl}(j, j_{\boldsymbol{x}})$ - the topological distance between the $j_{\boldsymbol{x}}$-th and j-th neurons. In case of the conventional SOM with 1DN, $d_{tpl}(j, j_{\boldsymbol{x}}) = |j - j_{\boldsymbol{x}}|$. However, when our mechanisms (presented below) for splitting and merging of the network structure are implemented, the conventional SOM with 1DN evolves toward a tree-like structure. As a result of that, the neighborhood of a given neuron in such a tree-like topology of our generalized SOMs is defined along all the arcs emanating from that neuron as shown in Fig. 1. Those arcs are the pieces of the conventional SOM with 1DN. 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.

In order to implement three mechanisms, listed as a), b), and c) in the first paragraph of this section, four operations are activated after each learning epoch (epoch means one pass of all learning data), provided that the required conditions are fulfilled.

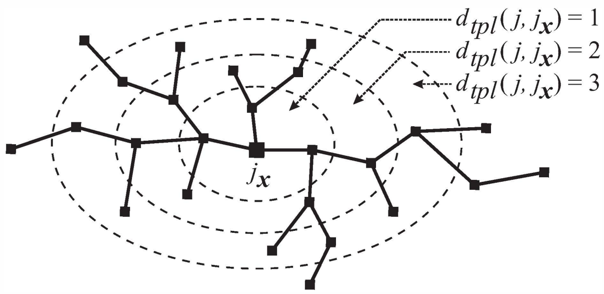
$d_{tpl}(j, j_{\boldsymbol{x}}) = 1$
$d_{tpl}(j, j_{\boldsymbol{x}}) = 2$
$d_{tpl}(j, j_{\boldsymbol{x}}) = 3$
$j_{\boldsymbol{x}}$

Fig. 1. Illustration of neighborhood of the $j_{\boldsymbol{x}}$-th neuron [4], [5]

Operation 1 (the removal of single low-active neurons): The neuron no. j_r is removed from the network (preserving the network continuity - see [3] 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 (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: $d(\boldsymbol{w}_{j_1}, \boldsymbol{w}_{j_2}) > d_{coef} d_{avr}$ where $d_{avr} = \frac{1}{P} \sum_{p=1}^{P} d_p$ is the average distance between two neighboring neurons for all pairs p, $p = 1, 2, \ldots, P$, of such neurons (d, d_{avr}, and d_p are the Euclidean distance measures). d_{coef} is experimentally selected parameter (a distance coefficient) governing the disconnection operation (usually, $d_{coef} \in [3, 4]$). Possible very short (single- or two-neuron) subnetworks are removed from the system since they cannot be reconnected by Operation 4 (see below).

Operation 3 (the insertion of additional neurons into the neighborhood of high-active neurons in order to take over some of their activities). Case 3a: 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 $win_{max} \in \{2, \ldots, 5\}$ 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 3b: 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 $win_j \leqslant win_{max}$ 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 [3] for details).

Operation 4 (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 $d(\boldsymbol{w}_{j_1}, \boldsymbol{w}_{j_2}) < d_{coef} \frac{d_{avr_{S_1}} + d_{avr_{S_2}}}{2}$. $d(\boldsymbol{w}_{j_1}, \boldsymbol{w}_{j_2})$ and d_{coef} are the same as in Operation 2. $d_{avr_{S_1}}$ and $d_{avr_{S_2}}$ are calculated for subnetworks S_1 and S_2, respectively, in the same way as d_{avr} is calculated in Operation 2 for the considered network.

According to Kohonen's comments [7], the selection of learning parameters is performed mainly in an experimental way assuming 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 our experiments $\eta_j(k) = \eta(k)$ of (2) is linearly decreasing over the learning horizon (with 10000 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, $win_{min} = 2$, $win_{max} = 4$, and $d_{coef} = 4$. The experiments on some benchmark data sets (see the following section) demonstrate that the same set of experimentally selected parameters governing the number of neurons in the network and its structure splitting and merging mechanisms gives excellent results in quite different (in terms of data dimensionality and cluster complexity) applications. They show, in a way, a low sensitivity of our approach to the selection of those parameters.

III. A BENCHMARK-DATA-BASED ILLUSTRATION AND EVALUATION OF OUR CLUSTERING TECHNIQUE

In order to illustrate the operation and to evaluate the performance of our approach, the clustering of two benchmark data sets from the so-called Fundamental Clustering Problem Suite (FCPS) [6] and one benchmark data set from the UCI repository of machine learning databases (http://archive.ics.uci.edu/ml) will be carried out. The FCPS is a collection of benchmark sets that, for different reasons, pose difficult problems to clustering algorithms. We selected two benchmark sets from FCPS, one two-dimensional (*WingNut* data set) and one three-dimensional (*Hepta* data set) to illustrate the performance of our approach. According to [6], main clustering problems in *WingNut* are largest densities at cluster borders and in *Hepta* - different densities in clusters. As far as UCI repository is concerned, the well-known *Wine* data set with 178 records, 13 numerical attributes, and 3 classes/clusters was selected. Since the class assignment of particular data records and the number of classes is known here, it allows us to directly verify the results obtained. However, it must be emphasized that the knowledge of the class assignments is by no means used by our approach. It works in a fully-unsupervised way, i.e., it operates on unlabelled data and without any predefinition of the cluster number.

Figs. 3 and 4 present the performance of our clustering technique applied to both data sets from FCPS. The figures are arranged correspondingly, i.e., part a) represents the data, parts b), c), d), e), and f) show the evolution of the tree-like structures of the generalized SOMs 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 automatically adjusts 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 SOM into appropriate number of subnetworks. Moreover, based on the obtained cluster multi-prototypes and using the aforementioned nearest multi-prototype algorithm [9], we can find that all data records in both sets are correctly assigned to corresponding clusters.

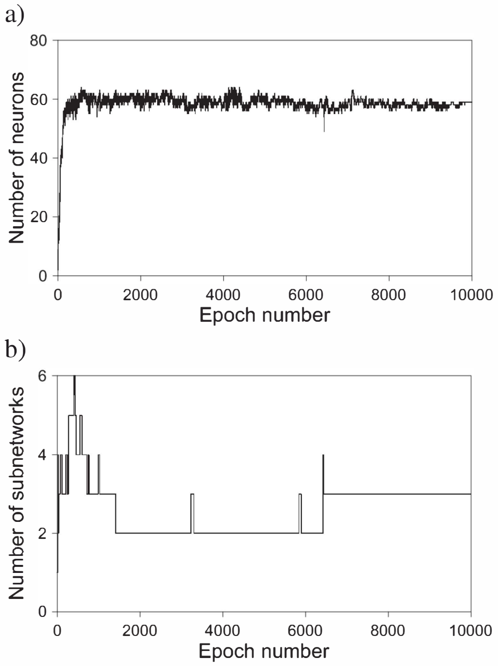

Fig. 2. Plots of the number of neurons (a) and the number of subnetworks (clusters) (b) vs. epoch number (*Wine* data set)

TABLE I
CLUSTERING RESULTS FOR *Wine* DATA SET

Class label	Number of samples	Number of decisions for subnetwork labelled:			Number of correct decisions	Number of wrong decisions	Percentage of correct decisions
		1	2	3			
1	59	57	2	0	57	2	96.6%
2	71	2	65	4	65	6	91.6%
3	48	0	0	48	48	0	100%
ALL	**178**	**59**	**67**	**52**	**170**	**8**	**95.5%**

Figs. 2 and Table I present the performance of our approach applied to *Wine* data set. Firstly, Fig. 2b shows that our approach detects the correct number of clusters in the considered data set. Secondly, since the number of classes and class assignments are known in the original set, a direct verification of the obtained results is also possible (see Table I). The percentage of correct decisions, equal to 95.51%, regarding the class assignments is very high (especially, since it has been achieved by the system working in a fully-unsupervised way).

IV. SAMPLE-BASED AND GENE-BASED CLUSTERING OF *Lymphoma* CANCER MICROARRAY DATA SET

In this section, the performance of our approach will be evaluated in the clustering of microarray gene expression data describing the *Lymphoma* human cancer (the data are available from the server of the Shenzhen University, China: http://csse.szu.edu.cn/staff/zhuzx/datasets.html). Microarray gene expression data sets usually contain thousands of original genes (in

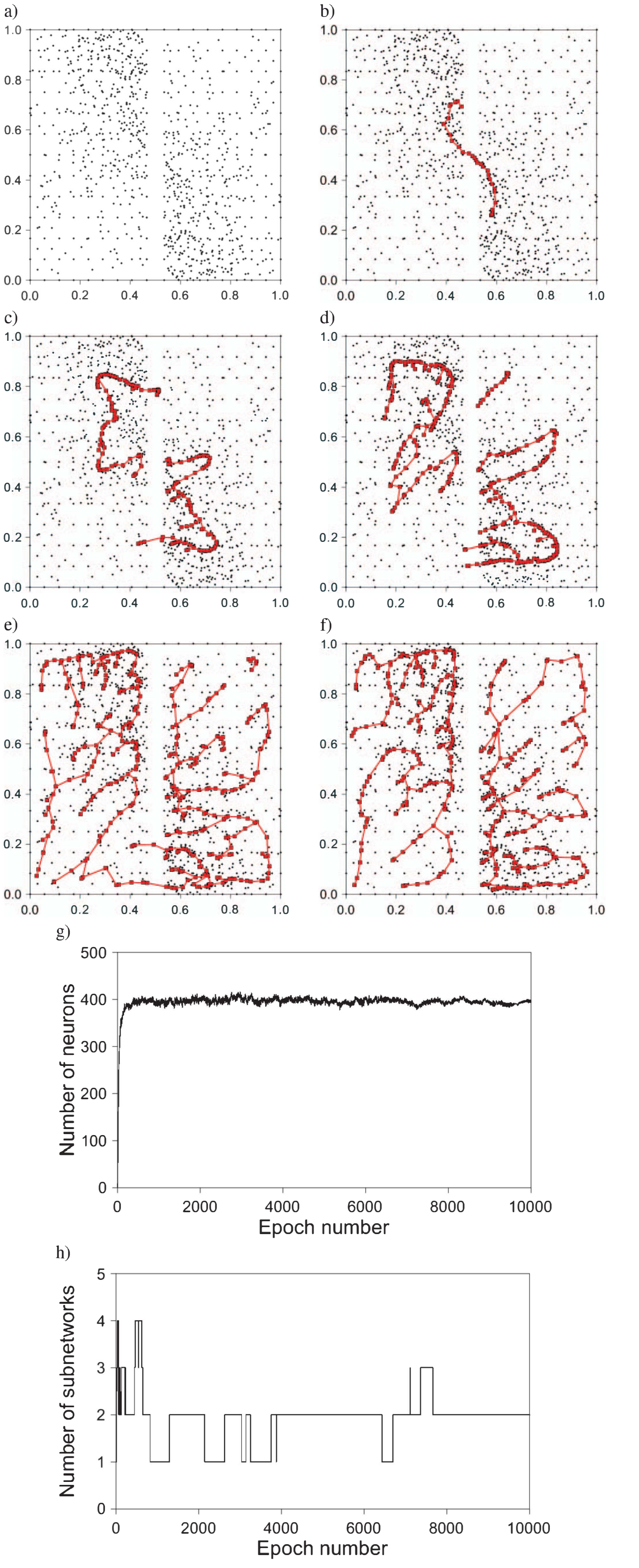

Fig. 3. *WingNut* data set (a) and the evolution of the generalized tree-like SOM in it in learning epochs: b) no. 5, c) no. 50, d) no. 100, e) no. 500, and f) no. 10 000 (end of learning), as well as plots of the number of neurons (g) and the number of subnetworks (clusters) (h) vs. epoch number

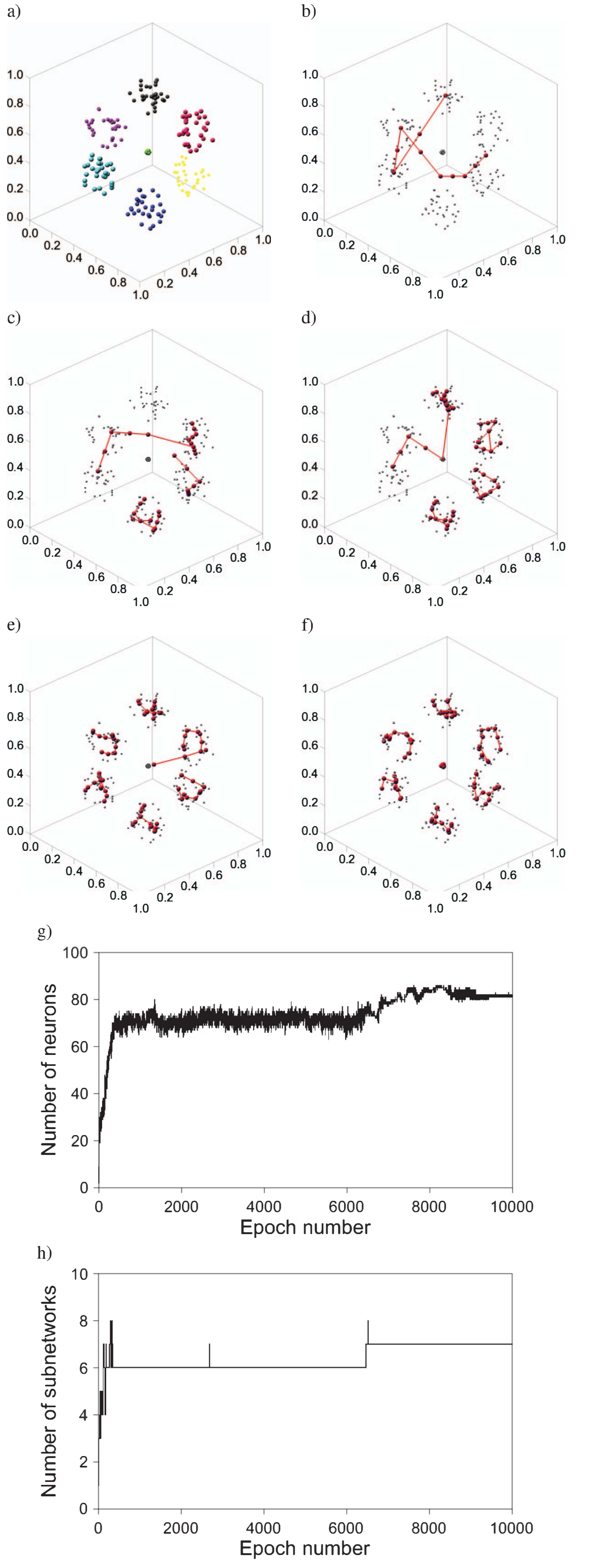

Fig. 4. *Hepta* data set (a) and the evolution of the generalized tree-like SOM in it in learning epochs: b) no. 5, c) no. 50, d) no. 100, e) no. 500, and f) no. 10 000 (end of learning), as well as plots of the number of neurons (g) and the number of subnetworks (clusters) (h) vs. epoch number

our case, 4.026) and a small number of samples (in our case, 62). The samples represent three types of lymphoma cancer, referred to as *diffuse large B-cell lymphoma* (DLBCL) with 40 samples, *follicular lymphoma* (FL) with 9 samples, and *chronic lymphocytic leukemia* (CLL) with 13 samples [14]. As already mentioned in the introduction, it is meaningful in gene expression data analysis to consider both the sample-based and gene-based clustering. In the first case, the samples are the objects and the genes are the features, whereas in the second case it is the opposite. Since a very small number of data samples is available, 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 d_{coef} is slightly reduced ($d_{coef} = 3$). The remaining parameters are unchanged.

Figs. 5, 6, 7, and 8 present the performance of our clustering algorithm applied to the considered data set. Figs. 5 and 6 show the plots of the number of neurons and the number of subnetworks (clusters) vs. epoch number for the sample-based and gene-based clustering respectively. In the case of the sample-based clustering, the number of clusters in the data set and the cluster assignments of particular data samples are known. Therefore, a direct verification of the obtained results is possible (see Table II). It should be emphasized, however, that our approach - similarly as in Section III - does not use that knowledge during its operation; it is used after the completion of the learning process to evaluate the obtained results. Fig. 5b shows that our approach detects the correct (equal to 3) number of sample clusters in the considered data set. The percentage of correct decisions, equal to 93.6% (see Table II for details), regarding the cluster assignments of particular data samples is higher than in case of several alternative approaches (see Table III). Moreover, the first three approaches of Table III additionally require the cluster number to be defined in advance.

In the case of the gene-based clustering, our approach detects 117 gene clusters (see Fig. 6b). Fig. 7 presents the pseudocolor image of some of them. Each of those clusters can be easily divided into three subclusters related to DLBCL, FL, and CLL samples. Therefore, the results generated by our approach correspond, in a way, to the so-called subspace clustering which is highly desirable in microarray data analysis (see discussion in [2]). The subspace clustering captures clusters created by a subset of genes across a subset of data samples (in our case, DLBCL, FL, and CLL samples separately). Fig. 8 shows the plots of the expression levels of all genes in the gene clusters of Fig. 7 confirming the compactness of particular clusters (as well as DLBCL, FL, and CLL related subclusters). The pseudocolor image of particular clusters of Fig. 7 (supported by Fig. 8) can be used in a deeper genetics-based discussion of the obtained results, which due to a limited space is not possible here (we can only briefly address one cluster of Fig. 7 which contains the highest number of genes with known IDs, i.e., the LY-96 gene cluster).

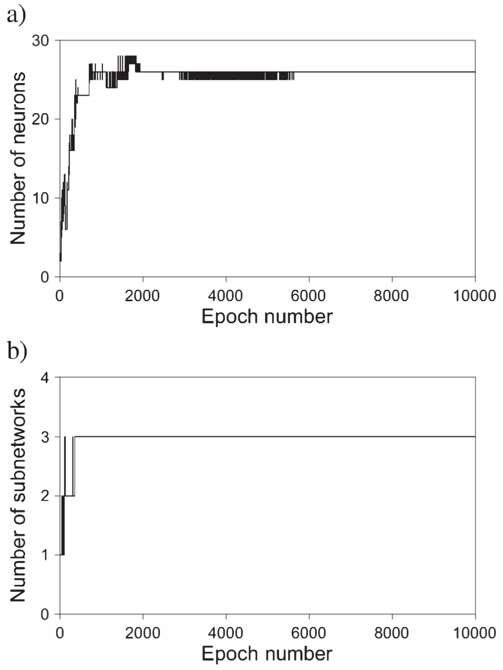

Fig. 5. Plots of the number of neurons (a) and the number of subnetworks (clusters) (b) vs. epoch number for the sample-based clustering of the *Lymphoma* data set

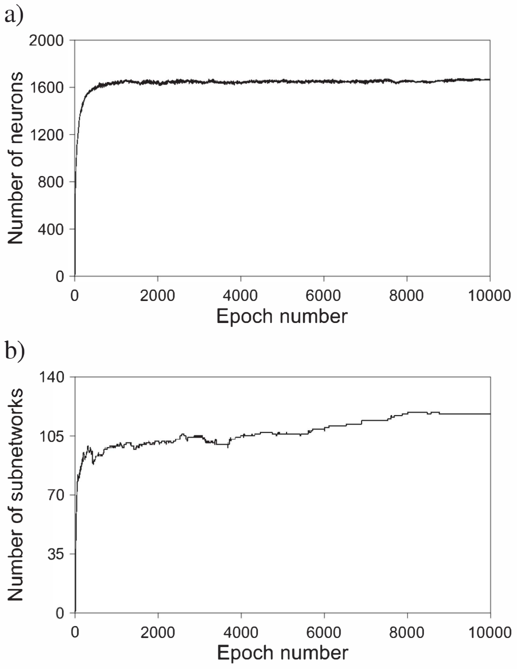

Fig. 6. Plots of the number of neurons (a) and the number of subnetworks (clusters) (b) vs. epoch number for the gene-based clustering of the *Lymphoma* data set

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 a publicly available functional profiling tool, i.e., the DAVID (Database for Annotation, Visualization and Integrated Discovery) software, available from the server of Laboratory

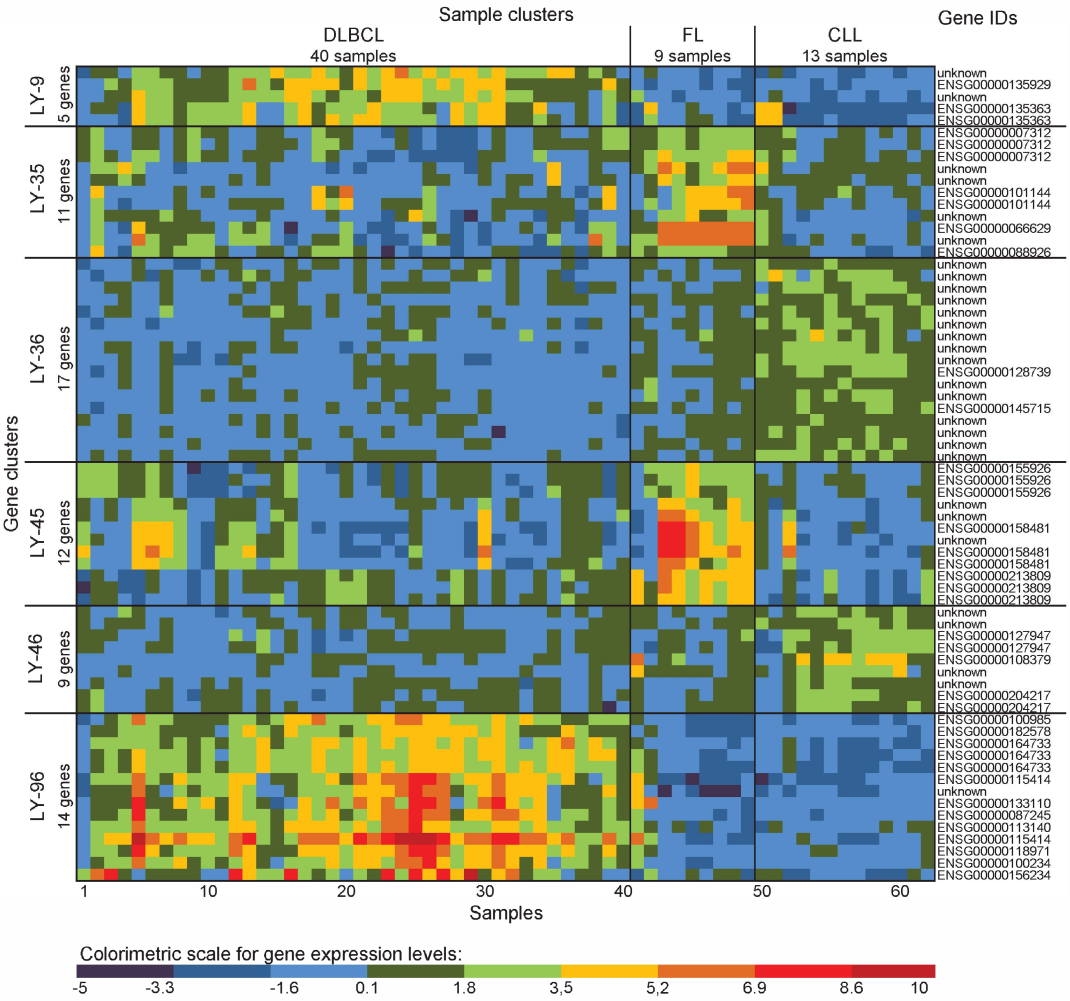

Fig. 7. Exemplary gene clusters in the *Lymphoma* data set

of Immunopathogenesis and Bioinformatics, National Cancer Institute at Frederick, USA (http://david.abcc.ncifcrf.gov). In the *Lymphoma* set, only 2.382 genes (59.2% of the overall number of genes) have unique gene IDs. IDs of the remaining genes are unknown (see the labels ”unknown” in Fig. 7). In order to perform an analysis of the gene-based clustering using the DAVID tool, it is necessary to convert the initial gene IDs to the so-called *Ensembl* format. The conversion was made by means of *GeneCards: Human Gene Database* available at the Weizmann Institute in Israel (http://genecards.org). As far as the aforementioned LY-96 gene cluster is concerned, the DAVID tool shows that the number of genes belonging to that cluster and performing the same functions is very high. For instance, 9 genes (i.e., 90% of the overall number of genes from LY-96 included in statistical analysis) perform the following functions: *disulfide bond*, *signal*, and *signal peptide*. In turn, 6 genes (60% of the number of genes) perform functions such as *extracellular matrix* and *proteinaceous extracellular matrix*, etc. Table IV presents a detailed list of gene functions characterized by highest statistical significance in LY-96 cluster, i.e., the functions for which the *p*-value significance level of the Fisher exact test is not greater than 10E-5. In conclusion, the analysis of the LY-96 gene cluster confirms the high effectiveness of our clustering technique in creating clusters with significant numbers of coexpressed genes. Similar analysis can be carried out for other gene clusters, as well as for other clustering algorithms for the purpose of comparison.

TABLE II
RESULTS OF SAMPLE-BASED CLUSTERING OF THE *Lymphoma* DATA SET

Class label	Number of samples	Number of decisions for subnetwork labelled:			Number of correct decisions	Number of wrong decisions	Percentage of correct decisions
		DLBCL	FL	CLL			
DLBCL	42	40	2	0	40	2	95.2%
FL	9	0	7	2	7	2	77.8%
CLL	11	0	0	11	11	0	100%
ALL	**62**	**40**	**9**	**13**	**58**	**4**	**93.6%**

V. Conclusions

The main goal of this paper is the application of the clustering technique proposed by us and based on generalized tree-like SOMs with evolving splitting-merging structures to complex clustering tasks including, in particular, the sample-based and gene-based clustering of the *Lymphoma* human cancer microarray data set. It is worth emphasizing that our approach works in a fully unsupervised way, i.e., using unlabelled data and without the necessity to predefine the number of clusters. It is particularly important in the gene-based clustering of microarray data for which the number of gene clusters is unknown in advance. During the learning process, the structure disconnection and reconnection mechanisms of the tree-like SOMs are activated and the automatic adjustment of the number of neurons in SOMs takes place. As a result, our approach automatically detects the number of clusters (equal to the number of disconnected subnetworks) in a given data set and generates multi-prototypes for particular clusters. The *Lymphoma* data clustering process is preceded in the paper by a brief presentation of our approach and its testing on the benchmark data sets selected from the FCPS collection and UCI repository. It is worth stressing that almost the same set of experimentally selected parameters that control the operation of our clustering technique gives very good clustering results for completely different types of data sets such as the FCPS and UCI benchmarks and microarray data. It is also worth emphasizing that in the sample-based clustering of the *Lymphoma* data set our approach gives much higher percentage of correct decisions than alternative techniques (some of them additionally require the cluster number to be defined in advance). As far as the gene-based clustering of the *Lymphoma* data set is concerned, our approach generates clusters that are easily divisible into subclusters related to particular sample classes; in some way, it corresponds to subspace clustering which is highly desirable in microarray data analysis [2].

TABLE III
Comparative analysis of sample-based clustering of the *Lymphoma* data set

No.	Clustering technique*)	Number of correct decisions	Number of wrong decisions	Percentage of correct decisions
1	k-means	42	20	67.7%
2	EM	38	24	61.3%
3	FFTA	47	15	75.8%
4	our approach of [10]	57	5	91.9%
5	**our present approach**	**58**	**4**	**93.6%**

*) k-means [15], EM - Expectation Maximization [16], FFTA - Farthest First Traversal Algorithm [17]

TABLE IV
The most statistically significant functions of genes from the LY-96 gene cluster

No.	Gene function	Informative genes			p-value
		Count	Gene IDs	Share	
1	*extracellular matrix* (category SP_PIR_KEYWORDS)	6	100234, 115414, 087245, 100985, 133110, 113140	60%	3.6E-8
2	*proteinaceous extracellular matrix*	6	100234, 115414, 087245, 100985, 133110, 113140	60%	1.1E-6
3	*extracellular matrix* (category GOTERM_CC_FAT)	6	100234, 115414, 087245, 100985, 133110, 113140	60%	1.6E-6
4	*disulfide bond* (category UP_SEQ_FEATURE)	9	100234, 164733, 156234, 182578, 115414, 087245, 100985, 133110, 113140	90%	1.7E-6
5	*domain:Fibronectin type-II 1*	3	115414, 087245, 100985	30%	2.0E-6
6	*domain:Fibronectin type-II 2*	3	115414, 087245, 100985	30%	2.0E-6
7	*disulfide bond* (category SP_PIR_KEYWORDS)	9	100234, 164733, 156234, 182578, 115414, 087245, 100985, 133110, 113140	90%	2.2E-6
8	*signal*	9	100234, 164733, 156234, 182578, 115414, 087245, 100985, 133110, 113140	90%	5.0E-6
9	*signal peptide*	9	100234, 164733, 156234, 182578, 115414, 087245, 100985, 133110, 113140	90%	5.3E-6

References

[1] M. Schena, *Microarray Analysis*. John Wiley & Sons, 2003.

[2] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: a survey," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1370–1386, Nov 2004.

[3] M. B. Gorzałczany, J. Piekoszewski, and F. Rudziński, "Generalized tree-like self-organizing neural networks with dynamically defined neighborhood for cluster analysis," in *Artificial Intelligence and Soft Computing - ICAISC 2014*, ser. Lecture Notes in Computer Science, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Żurada, Eds. Berlin: Springer-Verlag, 2014, vol. 8468, pp. 725–737.

[4] M. B. Gorzałczany, J. Piekoszewski, and F. Rudziński, "Microarray leukemia gene data clustering by means of generalized self-organizing neural networks with evolving tree-like structures," in *Artificial Intelligence and Soft Computing - ICAISC 2015*, ser. Lecture Notes in Computer Science, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Żurada, Eds. Springer-Verlag, 2015, vol. 9119, pp. 15–25.

[5] M. B. Gorzałczany, F. Rudziński, and J. Piekoszewski, "Generalized SOMs with splitting-merging tree-like structures for WWW-document clustering," in *2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15)*, J. M. Alonso, H. Bustince, and M. Reformat, Eds., vol. 89. Gijón, Spain: Atlantis Press, Jun. 30 – Jul. 3 2015, pp. 186–193.

[6] A. Ultsch, "Clustering with SOM: U*C," in *Proceedings of the Workshop on Self-●rganizing Maps*, Paris, France, 2005, pp. 75–82.

[7] T. Kohonen, *Self-●rganizing Maps*, 3rd ed. Berlin: Springer-Verlag, 2001.

[8] N. R. Pal, J. C. Bezdek, and E. C.-K. Tsao, "Generalized clustering networks and Kohonen's self-organizing scheme," *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 549–557, 1993.

[9] J. C. Bezdek, T. R. Reichherzer, G. S. Lim, and Y. Attikiouzel, "Multiple-prototype classifier design," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 28, no. 1, pp. 67–79, 1998.

[10] M. B. Gorzałczany and F. Rudziński, "Cluster analysis via dynamic self-organizing neural networks," in *Artificial Intelligence and Soft Computing - ICAISC 2006*, ser. Lecture Notes in Computer Science, L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. M. Żurada, Eds. Berlin: Springer-Verlag, 2006, vol. 4029, pp. 593–602.

[11] M. B. Gorzałczany and F. Rudziński, "WWW-newsgroup-document clustering by means of dynamic self-organizing neural networks," in *Artificial Intelligence and Soft Computing - ICAISC 2008*, ser. Lecture Notes in Computer Science, L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. M. Żurada, Eds. Berlin: Springer-Verlag, 2008, vol.

5097, pp. 40–51.

[12] M. B. Gorzałczany and F. Rudziński, “Application of genetic algorithms and Kohonen networks to cluster analysis,” in *Artificial Intelligence and Soft Computing - ICAISC 2004*, ser. Lecture Notes in Computer Science, L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. H. Siekmann, Eds. Berlin: Springer-Verlag, 2004, vol. 3070, pp. 556–561.

[13] M. B. Gorzałczany and F. Rudziński, “Modified Kohonen networks for complex cluster-analysis problems,” in *Artificial Intelligence and Soft Computing - ICAISC 2004*, ser. Lecture Notes in Computer Science, L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, and J. H. Siekmann, Eds. Berlin: Springer-Verlag, 2004, vol. 3070, pp. 562–567.

[14] A. Alizadeh *et al.*, “Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling,” *Nature*, vol. 403, no. 6769, pp. 503–511, Feb 2000.

[15] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967, pp. 281–297.

[16] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, “Flow clustering using machine learning techniques,” in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science, C. Barakat and I. Pratt, Eds. Berlin: Springer-Verlag, 2004, vol. 3015, pp. 205–214.

[17] M. Panda and M. R. Patra, “Detecting network intrusions - a clustering approach,” *International Journal of Secure Digital Information Age*, vol. 1, no. 2, pp. 1–6, 2009.

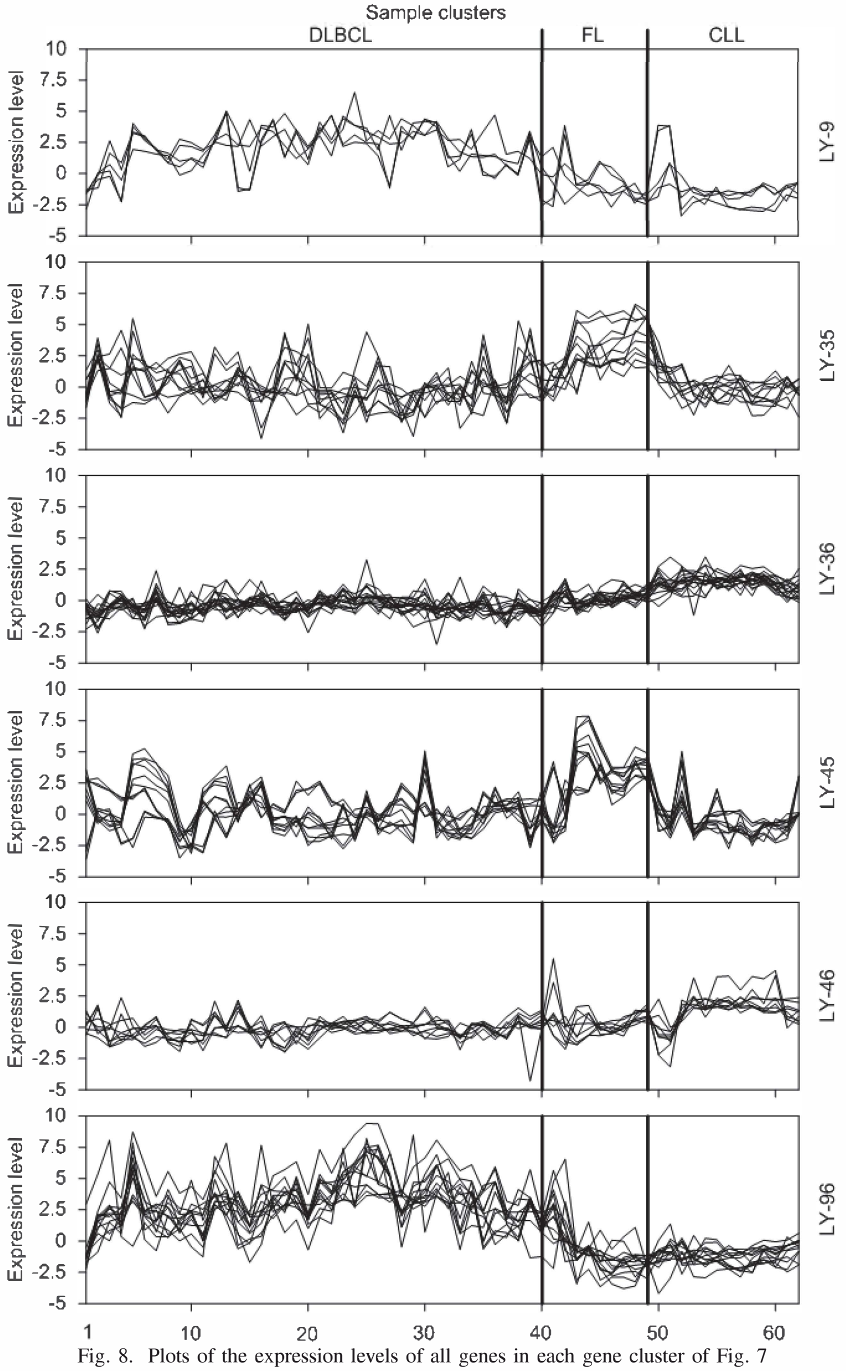

Fig. 8. Plots of the expression levels of all genes in each gene cluster of Fig. 7