View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by PORTO Publications Open Repository TOrino

A Graph-Based Representation of Gene Expression Profiles in DNA Microarrays

A. Benso, S. Di Carlo, G. Politano, L. Sterpone

***Abstract*— This paper proposes a new and very flexible data model, called Gene Expression Graph (GEG), for genes expression analysis and classification. Three features differentiate GEGs from other available microarray data representation structures: (i) the memory occupation of a GEG is independent of the number of samples used to built it; (ii) a GEG more clearly expresses relationships among expressed and non expressed genes in both healthy and diseased tissues experiments; (iii) GEGs allow to easily implement very efficient classifiers. The paper also presents a simple classifier for sample-based classification to show the flexibility and user-friendliness of the proposed data structure.**

I. Introduction

Microarray technology has evolved since the 1980s. A DNA microarray is a collection of microscopic DNA spots, usually representing single genes, regularly arranged on a solid support, such as a glass microscope slide, and covalently attached via a chemical matrix. Tens of thousands of DNA probes can be attached to a single slide, and the genes they represent can all be analyzed in a single experiment. Microarrays provide simultaneous expression measurements for thousands of genes and facilitate the analysis of the complex relations among them. This new technology is being used to address several goals in bioinformatics [1][2].

Among the different applications of microarrays, a challenging research problem is how to use genes expression data to classify diseases on a molecular level. It involves designing expression-based classifiers to discriminate differences in cells state, such as one type of cancer or another. An expression-based microarray phenotype classifier takes a vector of gene expression levels as input and outputs a class label to predict the class (phenotype) the input vector belongs to [3]. Classifiers design involves assessing expression levels from different microarrays experiments, determining spots/genes whose expression is relevant, and then applying a rule to design the classifier from the sampled microarray data.

The main problem in this type of classification is the huge disparity between the number of potential gene expressions (thousands) and the number of samples (usually less than a hundred). This extreme disparity impacts the major aspects of the classifier design: the classification rule, the error estimation, and the feature selection.

A. Benso, S. Di Carlo, G. Politano, and L. Sterpone are with the Department of Computer and Control Engineering of Politecnico di Torino, I-10129, Torino, Italy. Email: {alfredo.benso, stefano.dicarlo, gianfranco.politano, luca.sterpone}@polito.it

Many machine-learning techniques have been applied to classify microarray data. These techniques include artificial neural networks [4], [5], [6] and [7], Bayesian approaches [8] and [9], support vector machines [10], [11] and [12], decision trees [13] and [14], and *k* nearest neighbors [15].

Evolutionary techniques have also been used to analyze gene expression data. Genetic algorithms and genetic programming are mainly used in gene selection [23][24], optimal gene sets finding [25], disease prediction [26], and classification [27] [28] [29] [30]. Approaches that combine multiple classifiers have also received much attention in the past decade, and this is now a standard approach to improving classification performance in machine-learning. [31][32][33][34][35][36]

While the proposed solutions mainly focus on the definition of very efficient classification algorithms, one issue that has not been widely addressed is the definition of flexible data structures that can be used to represent classes of phenotypes and features relationships.

This paper proposes a new and very flexible data model for gene expression analysis and classification called *Gene Expression Graph* (GEG). Three features differentiate GEGs from other available microarray data representation structures, and in particular from Gene Expression Matrices [16]: (i) the memory occupation of a GEG is independent of the number of samples used to built it; (ii) a GEG more clearly expresses relationships among expressed and non expressed genes in both healthy and diseased tissues experiments; (iii) GEGs allow to easily locate potential *informative genes*, i.e. genes whose expression levels strongly correlate with a particular phenotype.

We believe that this new data model is able to intrinsically express very useful information about microarray experiments that can support the development of new and very efficient feature extraction and classification algorithms. Although the goal of this paper is to introduce the new data model, to demonstrate the usability of the proposed representation, the paper also presents a simple classifier for sample-based classification and its applications on a set of microarray experiments for three well known diseases: Diffuse Large B-Cell Lymphoma, Lymphocytic Leukemia Watch&Wait and Lymphocytic Leukemia.

The paper is organized as follows: Section II describes how to build a Gene Expression Graph starting from a set of experiments, and Section III proposes a simple example of classifier based on GEGs. Section IV presents some experimental results and Sections V concludes the paper suggesting future activities.

978-1-4244-1779-7/08/$25.00 ©2008 IEEE

II. Building Gene Expression Graphs

A microarray experiment typically assesses a large number of DNA sequences (genes, cDNA clones, or expressed sequence tags ESTs) under multiple conditions. These conditions may be a collection of different tissue samples (e.g., normal versus cancerous tissues).

The result of a microarray experiment is a gene expression dataset usually represented in the form of a real-valued expression matrix, called *Gene Expression Matri*x (GEM) [16][17]. A Gene Expression Matrix *M* defined for a set of *m* samples, each involving *n* genes is defined as:

$$M = \{w_{i,j} \mid 1 \le i \le n, 1 \le j \le m\} \quad (1)$$

where:

- Each row $\vec{g}_i$ $(1 \le i \le n)$ is associated to a gene. It identifies the expression pattern of gene *i* over *m* samples;
- Each column $\vec{s}_j$ $(1 \le j \le m)$ is associated to a sample. It represents the genes expression profile of the sample;
- Each element $w_{i,j} \in M$ measures the expression of gene *i* in sample *j*.

The original GEM obtained from the scanning process of a set of microarrays usually contains noise, missing values, and systematic variations arising from the experimental procedure. This row data is therefore usually pre-processed before performing any type of analysis. Examples of pre-processing techniques (out of the scope of this paper) can be found in [18][19][20].

A Gene Expression Graph modeling a microarray experiment can be easily built starting from one or more GEMs. A GEG elaborates the information contained in the GEMs in order to clearly highlight those genes considered "*expressed*".

Since gene expression levels ($w_{i,j}$) in a GEM are represented as real numbers in a continuous interval of valid values, it is clear that, in order to discriminate between expressed and non-expressed genes, it is necessary to define an *expression threshold T* able to remove the biological and experimental noise. This is one of the most critical parameters affecting the quality of the resulting GEG.

A Gene Expression Graph built over a GEM *M* is a non-oriented weighted graph $GEG = (V, E)$ where:

- *V* is the set of vertices. The vertex $v_i \in V$ is associated with gene *i* of M;
- $E = \{(u,v) \mid u,v \in V\}$ is the set of edges connecting vertices (genes). Two vertices *u* and *v* are connected by an edge iff the corresponding genes are both expressed in the same sample $\vec{s}_j \in M$.

If *n* genes are expressed in the same sample $\vec{s}_j \in M$, each corresponding vertex is connected to the other *n-1* in the graph. Therefore, genes expressed in the same sample constitute a *clique* in the graph.

Each edge $(u,v) \in E$ is finally weighted with a weight $W_{u,w}$ that counts the number of times the genes associated with vertexes *u* and *v* are simultaneously expressed in the same sample over the *m* samples included in *M.* In a graph representing a single sample (microarray), each edge will be weighted as 1. Adding additional experiments will modify the graph by introducing additional edges and/or by modifying the weight of existing ones. Implicitly, this representation takes also into account non-expressed genes. Missing vertexes correspond to non-expressed genes relationships.

Algorithm 1 summarizes, using a pseudo-code formalism, the steps required to build a GEG starting from a Gene Expression Matrix.

Since differential analysis between healthy and diseased tissues is widely used in gene expression analysis, to represent a complete microarray experiment we would need two graphs, one for the healthy tissue (green dye in c-DNA microarrays) and one for diseased one (red dye in c-DNA microarrays). Since both experiments share the same set of genes, we can compact the two resulting GEGs into a single graph $GEG = (V, E_d, E_h)$ with two sets of edges:

- *Diseased edges* (E_d): representing the genes expression relationships in the diseased tissues;
- *Healthy edges* (E_h): representing the genes expression relationships in the healthy tissue.

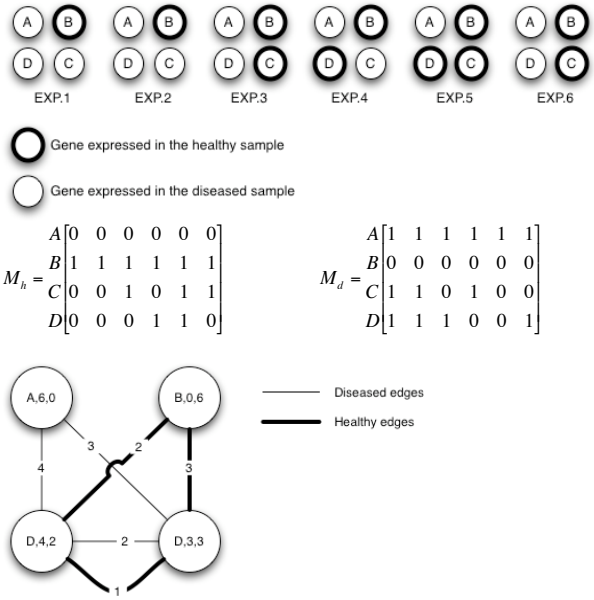
Finally, each vertex *v* of a GEG is also labeled with a set of additional information that may in turn be useful for future elaborations:

- The Name and UnigeneID [21] of the corresponding gene;
- The Total Expression Intensity (TEI) for both the healthy and the diseased tissues over the different samples. The TEI is computed as the sum of the expression intensities of the same gene in the different samples;
- The Expression Counts (EC) of the gene, i.e., the number of times the gene is expressed in the healthy and the diseased samples.

Algorithm 1: GEG Construction

1. *– GEM filtering using the threshold T*
2. **for** each $w_{i,j} \in M$ **do**
3. **if** $w_{i,j} > T$ **then**
4. $w_{i,j} \leftarrow 1$
5. **else**
6. $w_{i,j} \leftarrow 0$
7. **end if**
8. **end for**
9. *– GEM construction begins*
10. $V \leftarrow \varnothing$
11. $E \leftarrow \varnothing$
12. **for** $i \leftarrow 1 \ldots m$ **do**
13. **for** $j \leftarrow 1 \ldots n-1$ **do**
14. **if** $w_{j,i} == 1$ **then**
15. **if** $v_j \notin V$ **then** *add* v_j to V
16. **for** $k \leftarrow j+1 \ldots n$ **do**
17. **if** $w_{k,i} == 1$ **then**
18. **if** $v_k \notin V$ **then** *add* v_k to V
19. **if** $(v_j, v_k) \notin E$ **then**
20. *add* (v_j, v_k) to E
21. $W_{v_j,v_k} \leftarrow 0$
22. **end if**
23. $W_{v_j,v_k} \leftarrow W_{v_j,v_k} + 1$
24. **end if**
25. **end for**
26. **end if**
27. **end for**
28. **end for**

Fig. 1 shows an example of GEG construction from a set of six samples already filtered with the threshold *T* in order to identify expressed and not expressed genes. M_h and M_d are the two GEMs corresponding to the healthy and diseased tissue experiments used for creating the final graph. The label of each vertex in the graph reports the gene name, and the two expressions counts.



$$M_h = \begin{matrix} A \\ B \\ C \\ D \end{matrix}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \qquad M_d = \begin{matrix} A \\ B \\ C \\ D \end{matrix}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 1. GEG Construction Example

If new samples become available from new experiments referring to the same pathology, the related information can be easily added to the corresponding GEG at anytime. Since GEGs store information about both expressed and non-expressed genes, by simply changing the expression thresholds *T* it is possible to drastically reduce the number of significant genes to be used for clustering or signature extraction, and also to precisely target the set of genes to be investigated in future experiments. Finally, the memory occupation of a GEG is independent of the number of samples in the initial dataset.

A. Gene Expression Graphs Representation

To efficiently represent a GEG, we use a modified ADjacency Matrix (ADM). A standard ADM for a non-oriented graph $G = (V, E)$ is a $n \times n$ symmetrical matrix $ADM = \{c_{i,j} \mid 1 \le i \le n, 1 \le j \le n\}$ (with n equal to the number of vertices of G) where each cell $c_{i,j}$ stores the weight of the edge connecting vertex *i* to the vertex *j*.

The GEGADM uses the two halves of the matrix to represent the weights of the healthy edges (upper right half) and the diseased edges, respectively. In this way we are able to keep all the necessary information regarding the experiment on the same type of pathology in one very compact data structure. Fig. 2 shows the GEGADM for the example of Fig. 1.

The more experiments are available, the more information the matrix will store. The memory occupancy of this structure is independent of the number of experiments.

$$GEGADM = \begin{bmatrix} 0 & 0 & 3 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 2 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

Fig. 2. GEG Adjacency Matrix

B. Informative Genes

Besides using GEGs as data structures for the definition of efficient classifiers, they also allow deriving useful information about gene expression characteristics that can be used to understand the meaningfulness of a gene and its correlation with the *informative genes* describing a given pathology. If built from a dataset representing a well-defined phenotype, a GEG will start displaying clusters of edges with very high weights that show a very strong relationship in the expression of the genes corresponding to the vertices connected in the cluster. These clusters can be used to select the informative genes that represent the considered phenotype.

III. Classifier

To show the flexibility of the proposed data model we designed a simple classifier able to provide a *Proximity Measure* between a GEG representing a given phenotype (GEG_{pat}), and a GEG generated from a single microarray sample (GEG_{exp}).

The classification rule is in fact implemented as a weighted comparison between the two graphs. GEG_{exp}, is characterized by having all edges weighted with 1; moreover, it is by construction, a clique. GEG_{pat}, extracted from the dataset as described in the beginning of the previous section, contains edges weighted from the expression information of the considered phenotype.

We basically have four possible matching situations (Fig. 3):

1) *Perfect match*: GEG_{exp} and GEG_{pat} perfectly match. For each edge in GEG_{exp} there is a corresponding edge in GEG_{pat} with weight greater then zero, and viceversa. Consequently, in this case, the set of expressed genes in GEG_{exp} and in GEG_{pat} exactly match;

2) *Partial Match*: GEG_{exp} and GEG_{pat} partially match. There are three possible sub cases:

   a) The set of vertices in GEG_{exp} and the set of vertices in GEG_{pat} share some element, i.e., some of the genes expressed in GEG_{exp} (not all) are also expressed in GEG_{pat}, and viceversa;

   b) GEG_{exp} is a subset of GEG_{pat}. All genes expressed in GEG_{exp} are also expressed in GEG_{pat}, but not viceversa;

   c) GEG_{pat} is a subset of GEG_{exp}. All genes expressed in GEG_{pat} are also expressed in GEG_{exp}, but not viceversa.

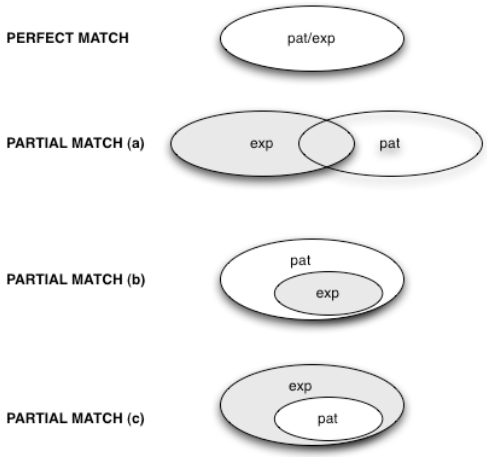


Fig. 3. GEG_{pat} and GEG_{exp} possible matches

The proximity measure defined in our classifier is computed multiplying two values: (i) a *Matching Score* (MS) providing a measure of how many edges in GEG_{exp} cover informative edges (edges with a high weight) in GEG_{pat}; and (ii) a *Confidence Score* (CS), that can be considered as an error estimation. It measures the quality of the match, taking into account how much of GEG_{exp} actually matches GEG_{pat}, and how much of it is left out because it is expressing genes that are not even present in the phenotype it is compared with.

CS is ranged between 0 and 1. Depending on the four matching situations of Fig. 3, we should expect the behavior reported in Table 1.

TABLE I
MS and CS Behavior

	CS
Perfect match	1
Partial Match-a	0÷1
Partial Match-b	1
Partial Match-c	0÷1

To compute the MS and CS we have first to introduce additional measures:

- The *Sample Signature Weight* (SSW) as the sum of all the weights of the arcs of GEG_{exp}:

$$SSW = \sum_{(i,j) \in E_{GEG_{exp}}} W_{i,j} \quad (2)$$

- The *Perfect Match Distance* (PMD) as the distance of the matching portion of GEG_{pat} from the optimal situation of matching with a complete clique with all the edges with maximum weight (equal to the number of samples #DS in the dataset):

$$PMD = \sum_{(i,j)\in E_{GEG_{\exp}}} \#DS - W_{i,j_{GEG_{pat}}} \quad (3)$$

The better is the match; the higher is PMD, which in case of a perfect match would tend to ∞;

- The *Sample Matching Weight* (SMW) as the sum of the weights of all the edges in GEG_{exp} that have a matching arc in GEG_{pat} (or, since the weights of GEG_{exp} are all equal to 1, SMW corresponds to the number of matching arcs):

$$SMW = \sum W_{ij_{GEG_{pat}}} \mid \exists (i,j) \text{ in } E_{GEG_{\exp}} \quad (4)$$

- The *GEG Matching Weight* (GMW) as the sum of the weights of all the edges in GEG_{pat} that have a corresponding edge in GEG_{exp}.

$$GMW = \sum W_{ij_{GEG_{\exp}}} \mid \exists (i,j) \text{ in } E_{GEG_{pat}} \quad (5)$$

The *Matching Score* (MS) can now be defined as:

$$MS = GMW / PMD \quad (6)$$

The *Confidence Score* (CS) of the disease sample compared with the dataset graph is defined as:

$$CS = SMW / SSW \quad (7)$$

Finally, the *Proximity Measure* (PM) is computed as:

$$PM = CS \cdot MS \quad (8)$$

IV. Experimental Results

Before discussing the experimental results, it is important to analyze the data sources we used to generate our test GEGs. Each used dataset comes from cDNA Stanford's Microarray database [22]. The problem with this data is that in many cases it refers to old experiments done on first generations of microarrays affected by probe sensing problems, reduced gene-set, and lack of UnigeneID [21] for many spots. Moreover, since old microarrays often used to duplicate spots in order to have more reliable results, in our GEG generation procedure we considered as "expressed" a gene expressed in at least one of its copies on the microarray. Also, we had to discard all information concerning spots that did not have a valid UniGeneID.

Even if the model allows to use and combine together data coming from different types of microarrays embedding different genes, in this set of experiments we used samples that have the same microarray's technology and genes set.

A. Data source and Dataset

The Stanford's collection catalogs a huge number of experiments using the cDNA chip technology. cDNA chips use two colors to distinguish tissues: green for the healthy tissue (wavelength of 635nm) and red for the diseased one (wavelength of 532 nm).

Besides the microarray image, each experiment is associated with the corresponding microarray's image and a text file in CSV (Comma Separated Values) format in which each line describes a spot. From the set of CSV files we derived a Gene Expression Matrix for each considered dataset.

We created three datasets: B-Cell, Lymphocytic Leukemia Watch&Wait and Lymphocytic Leukemia. The first dataset used to create the graph is a group of 53 microarrays related to Diffuse Large B-Cell Lymphoma (that is a non-Hodgkin Lymphoma disease). From parsing the corresponding CSV files we obtained a GEG of 6826 correctly named (using UniGeneID) vertices. The second dataset is a group of 22 microarrays focusing on Lymphocytic Leukemia Watch&Wait. From this set we extracted valid information for 7628 genes. Finally, the third dataset targeting Lymphocytic Leukemia is a group of 12 experiments from which we were able to extract valid information for 6826 genes.

To select the expressed nodes, we analyzed the expression levels distribution of the spots on all datasets and we performed various experiments using different values of T. As a result of this analysis we decided to adopt a threshold equal T=3000 (the expression ranges between 0 and 25,000) that seems (in our case) able to keep enough information about expressed genes.

B. Classifier

To verify the usability of the proposed model for sample-based classification algorithms, we applied the classification procedure described in Section II.B using, as samples, 6 different sets of microarrays data downloaded from Stanford Microarray's Database. Each set contains 11 distinct experiments. We used the three datasets described in Section IV.A as classes in which to classify the samples.

Each sample set is targeting a different phenotype:

1. Lymphocytic Leukemia W&W (watch and wait);
2. Lymphoma Normal Subset – non specific lymphoma subset;
3. Lymphocytic Leukemia - non specific subset;
4. Diffuse Large B-cell Lymphoma – Subset of B-cell sample not used during the graph creation and used here as cross-validation of the B-cell dataset;
5. Tumor: Brain – solid tumor;
6. Tumor: Ovarian – solid tumor.

We divided the 8 sample sets into four main groups. The first group contains pathologies #5 and #6, which are both referred to as solid tumors and therefore highly different from the lymphoma. We expect a strong difference from all datasets. The second group includes pathology #2, characterized by a slight affinity with the dataset. The third

group, including phenotypes #1 and #3, is very important because each sample represents a subset of very similar tissues. The idea is to observe if their unspecificity is evident in a sufficient distance from the specific B-Cell signature of the dataset. Finally, the fourth group contains the sample #4, a non-folded subset of the B-cell disease. This is the same pathology than the one used to generate the Dataset and it is used as cross-reference check for the classifier.

Fig. 4-5-6-7 report the Proximity Distances results computed by the classifier for pathologies 1 to 4. For the last two, the classifier correctly returned a null match with the three datasets.

Fig. 4 shows how the classifier correctly puts the Lymphocytic Leukemia W&W in the W&W dataset. The same happens for the Diffuse Large B-Cell Lymphoma classified in the B-Cell dataset of Fig. 7.

Fig. 5 shows how the classifier is able to classify the sample to all three classes. This is important since we do not have a dataset for that particular type of Lymphoma. Nevertheless, the classifier correctly recognizes the disease as a Lymphoma.

Fig. 6 is also very important because as expected the classifier recognizes that the sample (Lymphocytic) is not a Diffuse Large B-cell Lymphoma; it correctly classifies the sample in both the remaining datasets.

V. Conclusions and Future Works

In this paper we presented a new data structure designed for the analysis of gene expression data in microarrays experiments. The proposed model is essentially based on a graph representing meaningful expression relationships between spots. Gene Expression Graphs have many advantages over other known standards, as Gene Expression Matrices, in particular in terms of memory occupation and identification of the most informative genes and genes relationships. The full potential of this new data model is still under investigation, but it is believed to be able to provide a very useful ground for the development of new gene expression analysis algorithms.

To demonstrate the flexibility of the approach we also implemented a very simple classifier. The results demonstrate how the topological information extracted from the GEGs allows a very easy classification.

A lot of work is under way on GEGs. One of the first problems we encountered is the choice of the optimal threshold to use when building the graph. The results obtained so far showed that the efficiency of the classification algorithm is too sensitive to the choice of the threshold. We are therefore modifying the GEG generation algorithm, avoiding to base it on absolute expression values, but considering instead the differential expression between healthy and diseased samples. This new approach is guaranteeing a significant increase in the robustness of the data structure and a consequent reduction of the sensitivity of the classification algorithms to the expression threshold.

Also, we are working on the development of more detailed and complete supervised and unsupervised analysis algorithms able to fully exploit the information stored in a GEG.

References

[1] G. Gibson, "*Microarray Analysis"*, PLoS Biology, Vol.1, No. 1, Oct. 2003, pp. 28-29

[2] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armananzas, G. Santafe, A. Perez and V. Robles, "*Machine learning in bioinformatics*", Briefings in Bioinformatics, Vol. 7, No. 1, Feb. 2006, pp. 86-112

[3] E. R. Dougherty, "*The fundamental role of pattern recognition for gene-expression/microarray data in bioinformatics*", Pattern Recognition, Vol. 38, No. 12, Dec. 2005, pp. 2226-2228

[4] F. Azuaje, "*A computational neural approach to support the discovery of gene function and classes of cancer*", IEEE Trans. Biomed. Eng. Vol. 48, 2001, pp. 332–339

[5] J. Khan, J. Wei, M. Ringner, L. Saal, M. Ladanyi and F. Westermann et al., "*Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks*", Nat. Med. Vol. 7, 2001, pp. 673–679

[6] A. Albrecht, S. Vinterbo and L. Ohno-Machado, "*An epicurean learning approach to gene-expression data classification*", Artif Intell Med Vol. 28, 2003, pp. 75–87

[7] C. Huang and W. Liao, "*Application of probabilistic neural networks to the class prediction of leukemia and embryonal tumor of central nervous system*", Neural Process Lett, Vol. 19, 2004

[8] V. Roth and T. Lange, "*Bayesian class discovery in microarray datasets*", IEEE Trans Biomed Eng, Vol. 51, 2004, pp. 707–718

[9] X. Zhou, K. Liu and S. Wong, "*Cancer classification and prediction using logistic regression with Bayesian gene selection*", J Biomed Inform, Vol. 37, 2004

[10] F. Pan, B. Wang, X. Hu and W. Perrizo, "*Comprehensive vertical sample-based KNN/LSVM classification for gene expression analysis*", J Biomed Inform, Vol. 37, 2004, pp. 240–248

[11] C. Ding and I. Dubchak, "*Multi-class protein fold recognition using support vector machines and neural networks*", Bioinformatics, Vol. 17, 2001, pp. 349–358

[12] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang and M. Angelo et al., "*Multiclass cancer diagnosis using tumor gene expression signature*s", Proc Natl Acad Sci 98, 2001, pp. 15149–15154

[13] N. Camp and M. Slattery, "*Classification tree analysis: a statistical tool to investigate risk factor interactions with an example for colon cancer*", Cancer Causes Contr Vol. 13, 2002, pp. 813–823

[14] H. Zhang, C. Yu and B. Singer, "*Cell and tumor classification using gene expression data: construction of forests*", Proc Natl Acad Sci Vol. 100, 2003, pp. 4168–4172

[15] L. Li, C. Weinberg, T. Darden and L. Pedersen, "*Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method*", Bioinformatics, Vol. 17, 2001, pp. 1131–1142

[16] Wen, X., Fuhrman, S., Michaels, G.S., Carr, D.B., Smith, S., Barker, J.L., and Somogyi, R., 1997, "*Large-scale temporal gene expression mapping of CNS development*", Proc. Natl. Acad. Sci., in press.

[17] D. Jiang, C. Tang, and A, Zhang, "*Cluster Analysis for Gene Expression Data: A Survey*", IEEE Transaction On Knowledge and Data Engineering, Vol. 16, No. 11, Nov. 2004.

[18] Troyanskaya O., Cantor M., Sherlock G. Brown P., Hastie T., Tibshirani R., Botstein D. and Altman R., "*Missing value estimation methods for dna microarrays*", Bioinformatics, in press

[19] Hill A., Brown E., Whitley M., Tucker-Kellog G., Hunter C., Slonim D., "Evaluation of normalization procedures for oligonucletide array data based on spiked cRNA controls" Genome Miology, Vo. l2, No. 12, 2001

[20] Schuchhardt J., Beule D., Malik A., Wolski E., Eickhoff H., Lehrach H., and Herzel H., "Normalization strategies fo cDNA microarrays", Nucleic Acids Research, Vol. 28, No. 10, 2000

[21] UniGene, http://www.ncbi.nlm.nih.gov/sites/entrez?db=unigene

[22] cDNA Stanford's Microarray database http://genome-www.stanford.edu/

[23] 15. Li L, Weinberg CR, Darden TA, et al. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 2001;17(12):1131–42.
[24] 16. Durbin R, Eddy SR, Krogh A, et al. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, 1998.
[25] 17. Gary B. Fogel, David W. Corne. Evolutionary Computation in Bioinformatics. Morgan Kaufmann, 2002.
[26] 18. Frasconi P, Shamir R (eds). Artificial Intelligence and Heuristic Methods in Bioinformatics, Volume 183, NATO Science Series: Computer and Systems Sciences Edited. NATO, 2003.
[27] 19. Higgins D, Taylor W (eds). Bioinformatics. Sequence, Structure, and Databanks. Oxford University Press, 2000.
[28] 20. Husmeier D, Dybowski R, Roberts S (eds). Probabilistic Modeling in Bioinformatics and Medical Informatics. Springer Verlag, 2005.
[29] 21. Jagota A. Data Analysis and Classification for Bioinformatics. Bioinformatics by the Bay Press, 2000.
[30] 22. Jiang T, Xu X, Zhang MQ (eds). Current Topics in Computational Molecular Biology. The MIT Press, 2002.
[31] 24. Scholkopf B, Tsuda K, Vert J.-P (eds). Kernel Methods in Computational Biology, . The MIT Press, 2004.
[32] 25. Seiffert U, Jain LC, Schweizer P (eds). Bioinformatics Using Computational Intelligence Paradigms. Springer Verlag, 2005.
[33] 26. Wang JTL, Zaki MJ, Toivonen HTT, et al. (eds). Data Mining in Bioinformatics. Springer-Verlag, 2004.
[34] 27. Wu CH, McLarty JW. Neural Networks and Genome Identification. Elsevier, 2000.
[35] 28. Larran˜ aga P, Menasalvas E, Pen˜ a JM, et al. Special issue in data mining in genomics and proteomics. Artificial Intelligence in Medicine 2003;31:III–IV.
[36] 29. Li J, Wong L, Yang Q. Special issue on data mining for bioinformatics. IEEE Intelligent Systems 2005;20(6).

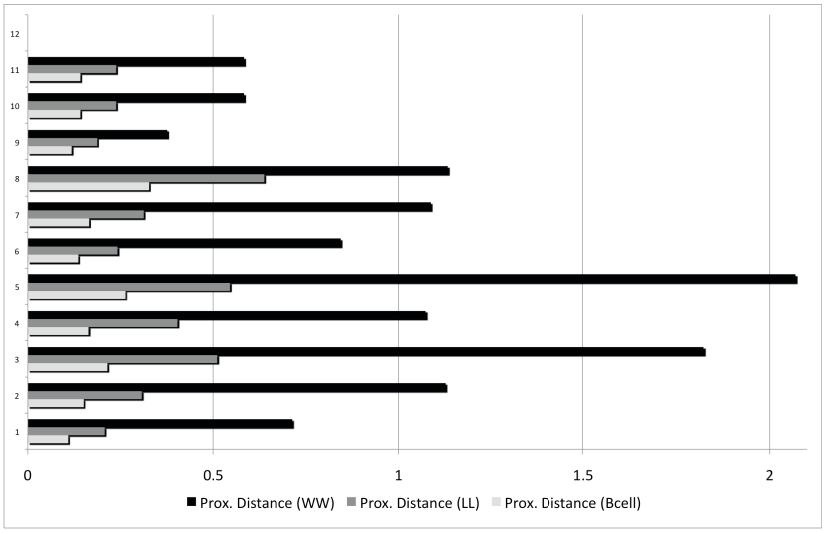


Fig. 4. Classifier results: Chronic Lymphocytic Leukemia (CLL) Watch & Wait

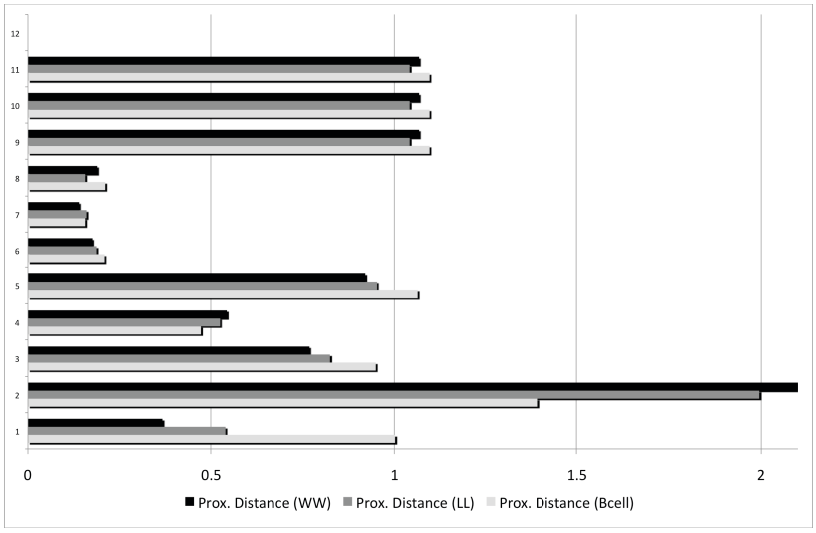


Fig. 5. Classifier results: Lymphoma Hematopoietic

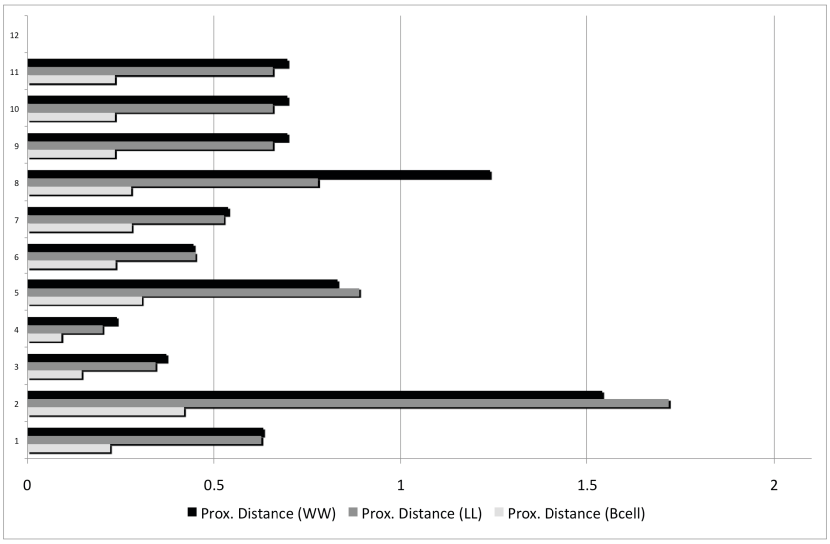


Fig. 6. Classifier results: Lymphocytic Leukemia (LL) – no specific subset

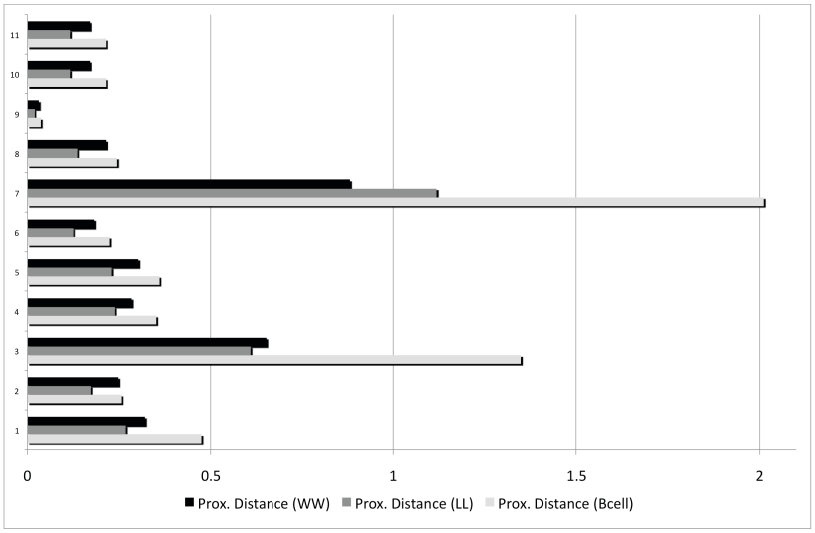


Fig. 7. Classifier results: Diffuse Large B-cell Lymphoma