Differential gene expression graphs: A data structure for classification in DNA microarrays

Differential Gene Expression Graphs: a data structure for classification in DNA Microarrays

Alfredo Benso, Stefano Di Carlo, Gianfranco Politano, and Luca Sterpone

***Abstract*— This paper proposes an innovative data structure to be used as a backbone in designing microarray phenotype sample classifiers. The data structure is based on graphs and it is built from a differential analysis of the expression levels of healthy and diseased tissue samples in a microarray dataset. The proposed data structure is built in such a way that, by construction, it shows a number of properties that are perfectly suited to address several problems like feature extraction, clustering, and classification.**

I. Introduction

DNA microarrays are small solid supports, usually membranes or glass slides, on which sequences of DNA are fixed in an orderly arrangement. Tens of thousands of DNA probes can be attached to a single slide. DNA microarrays are used to analyze and measure the activity of genes. Researchers can use microarrays and other methods to measure changes in gene expression and thereby learn how cells respond to a disease or to some other challenge [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. Statistical classification is a procedure in which individual items are placed into groups based on quantitative information on one or more characteristics inherent in the items (referred to as traits, variables, characters, etc.) and based on a training set of previously labeled items. In the case of microarrays it involves assessing gene expression levels from different experiments, determining spots/genes whose expression is relevant (*feature extraction* and *clustering*), and then applying a rule to design the classifier from the sampled microarray data (*classification*). 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].

The main problem in this type of classification is the huge disparity between the number of potential gene expressions (thousands) w.r.t. the number of samples (usually less than a hundred). This disparity impacts the major aspects of the classifier design: the classification rule, the error estimation, and the feature selection. Many machine-learning techniques have been applied to classify microarray data. These techniques include artificial neural networks [4], [5], [6], [7], Bayesian approaches [8], [9], support vector machines [10], [11], [12], decision trees [13], [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 [16], [17], optimal gene sets finding [18], disease prediction [19], and classification [20], [21], [22], [23]. Approaches that combine multiple classiers have also received much attention in the past decade, and this is now a standard approach to improving classification performance in machine-learning [24], [25], [26], [27], [28].

All these techniques mainly focus on the definition or application of statistical methods and algorithms on the huge amount of often messy data provided with microarrays. With this paper we want to shift the attention towards the data itself, by proposing a new data structure for representing classes of phenotypes and feature relationships that, *by construction*, easily allows feature selection, clustering and classification.

The proposed data model called *Differential Gene Expression Graph* (DGEG) is not an alternative to traditional data structures used to represent microarray experiments such as Gene Expression Matrices (GEM) [29]. It complements these models by providing a methodology to organize information in a more analysis-friendly format. DGEGs clearly express relationships among expressed and silenced genes in both healthy and diseased tissues experiments. Moreover, they support the identification of potential *informative genes*, i.e., genes that strongly correlate with the identification of the phenotype represented by the considered dataset.

To demonstrate the effectiveness and flexibility of the proposed data representation, the paper presents a new DGEG-based classifier and its application to a set of microarray experiments for three well known diseases: Diffuse Large B-Cell Lymphoma, Lymphocytic Leukemia Watch&Wait and Lymphocytic Leukemia. Instead of using traditional statistical approaches, the classifier is based on the topological analysis of the DGEG. Experimental results show that it is able to provide very reliable results, correctly classifying 100% of the considered samples.

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

Manuscript received June 16, 2008. This work was not supported by any organization.

A. Benso, S. Di Carlo, G. Politano, and L. Sterpone are with the Department of Control and Computer Engineering of Politecnico di Torino, Corso Duca degli Abruzzi 24 -10129 Torino Italy. Emails: {alfredo.benso, stefano.dicarlo, gianfranco.politano, luca.sterpone}@polito.it

II. Building differential gene expression graphs

A microarray experiment typically assesses a large number of DNA sequences (e.g., genes, cDNA clones, or expressed sequence tags ESTs) under multiple conditions, e.g., a collection of different tissue samples. 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 Matrix (GEM) [29], [30].

A *Gene Expression Matrix* M defined for a set of m samples, each involving n genes is defined as:

$$M : i \in \{1, 2, \ldots, n\} \times j \in \{1, 2, \ldots, m\} \rightarrow e_{i,j} \in \Re \quad (1)$$

where:

- Each row $\vec{g_i}$ $(1 \leq i \leq n)$ is associated with a gene g_i. It identifies the expression pattern of g_i over m samples;
- Each column $\vec{s_j}$ $(1 \leq j \leq m)$ is associated with a sample. It represents the gene expression profile of the sample;
- Each element $e_{i,j}$ of M measures the expression of g_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 raw data is therefore usually pre-processed before performing any type of analysis. Examples of pre-processing techniques can be found in [31], [32], [33].

In [34] we introduced the concept of Gene Expression Graph (GEG) built from a GEM M. A GEG is a non-oriented weighted graph $GEG = (V, E)$ where each vertex $v_i \in V$ corresponds to a gene g_i of M. Two nodes u and v are connected by an edge in the graph iff the corresponding genes g_u and g_v are both expressed in the same sample. Each edge $(u, v) \in E$ is weighted with the number of times g_u and g_v are simultaneously expressed in the same sample over the m samples included in the training set M.

One of the main difficulties with this model is the correct identification of expressed and silenced (not expressed) genes. This task is based on the identification of a threshold representing a Boolean cutoff to decide whether a gene is expressed or not. The high sensitivity of the GEG to this threshold makes any attempt at building GEG-based classifiers not robust enough.

To overcome this problem we propose a new graph model named *Differential Gene Expression Graph* (DGEG). DGEGs consider the *differential expression* between healthy and diseased samples. They not only overcome GEGs limitations, but also allow the cancellation of most of the "noise" often present in microarray data. This noise is usually due to differences in the gene expression profiles of the healthy and diseased samples not related to the disease.

The use of DGEGs implies the availability of a training set including, for each sample and for each considered gene, the expression level of both an healthy and a diseased tissue. This information can be organized in an extended Gene Expression Matrix (eGEM) introducing a third dimension to store healthy and diseased tissues information. A eGEM eM can be thus defined as:

$$\begin{aligned} eM : & k \in \{healthy, diseased\} \times i \in \{1, 2, \ldots, n\} \times \\ & j \in \{1, 2, \ldots, m\} \rightarrow e_{k,i,j} \in \Re \end{aligned} \quad (2)$$

This requirement makes DGEGs better suited for c-DNA microarrays, which always embed both healthy and diseased tissue samples. Application to other types of microarrays is under investigation.

The *Differential Expression* ($DE_{i,j}$) of gene g_i in sample j is the difference between the expression of g_i in the diseased tissue ($e_{diseased,i,j} \in eM$) and the one in the healthy one ($e_{healty,i,j} \in eM$):

$$DE_{i,j} = e_{diseased,i,j} - e_{healty,i,j} \quad (3)$$

If $|DE_{i,j}| > T_{diff}$, where T_{diff} is a *differential threshold*, g_i is considered *differentially expressed* in sample j; otherwise, the gene is considered not related to the disease, since it does not show a significant difference between the two samples. The differential analysis guarantees a significant increase in the robustness of the procedure to identify expressed genes (see Section IV-B) and allows building more precise classification algorithms (see section III).

A Differential Gene Expression Graph built over an eGEM eM can be thus defined as a non-oriented weighted graph $DGEG = (V, E, T_{diff})$ where:

- V is the set of vertexes. $v_i \in V$ is associated with gene g_i of eM. It exists in the DGEG only if g_i is differentially expressed in at least one sample of eM;
- $E = \{(u, v) \mid u, v \in V\}$ is the set of edges connecting the vertexes (genes). Edges show relationships among expressed and silenced genes. Two vertexes u and v are connected by an edge iff the corresponding genes are differentially expressed in at least one sample of eM;
- T_{diff} is the differential threshold used to build the graph.

If n genes are differentially expressed in the same sample each corresponding vertex will be connected to the other $n - 1$ in the graph, creating a clique. This property is very important since it could be exploited for the development of new feature extraction algorithms (not in the scope of this paper).

The weight $w_{u,v}$ of each edge $(u, v) \in E$ corresponds to the number of times the two corresponding genes are simultaneously differentially expressed in the same sample over the m samples included in eM. 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.

Finally, each node $v_i \in V$ of a DGEG related to a gene g_i of the training set is also labeled with a set of additional information, useful for classification purposes:

- The Name and UnigeneID [35] of g_i;
- The Cumulative Expression Counts (CEC_i) of g_i. CEC_i is computed as follows: starting with $CEC_i =$

0, for each sample j in eM the value of $DE_{i,j}$ is analyzed. If $DE_{i,j}$ is positive (i.e., g_i is expressed in the diseased sample but silenced in the healthy one) CEC_i is incremented by one; if $DE_{i,j}$ is negative (i.e., g_i is silenced in the diseased sample but expressed in the healthy one) CEC_i is decremented; if $DE_{i,j} = 0$ (i.e., g_i is expressed/silenced in both samples) CEC_i is not modified. In this way a node with a positive CEC corresponds to a gene that most of the time is expressed in the diseased sample and silenced in the healthy one, while a negative CEC indicates a gene that is most of the time silenced by the disease.

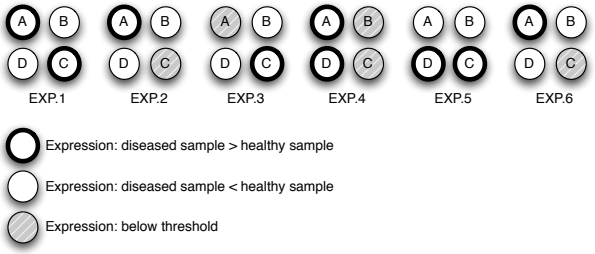
Fig. 1 and Fig. 2 show an example of DGEG construction from a set of six samples. Each sample is composed of 4 genes. The top part of Fig. 1 reports the six considered microarrays and the corresponding eGEM. The bottom part summarizes the Differential Expressions, shading in gray those genes that have a difference lower than the chosen threshold (100). From these values it is possible to build the DGEG of Fig. 2, where each vertex corresponds to a gene that is differentially expressed in at least one experiment. To give an example of how to compute the CEC for each vertex and the weight of each arc, let us look in more details at vertexes A and B. If we look at the Differential Expression table, we see that the DE of gene A is positive in 4 experiments (Exp. 1, 2, 4, and 6), negative in one (Exp. 5), and below the threshold in one (Exp. 3). The Cumulative Expression Count of node A in the DGEG is therefore $CEC_A = 4 - 1 = 3$. Gene B, instead, is differentially expressed with a negative sign in five experiments, and below the threshold in one experiment: its CEC is therefore -5. To compute the weight of the edge (A, B) it is enough to count the number of experiments in which both genes are differentially expressed (this time without taking into account the sign). They are experiments 1, 2, 5, and 6; the weight $w_{A,B}$ is therefore 4.

If new samples become available from new experiments referring to the same pathology, the related information can be easily added to the corresponding DGEG without any additional memory requirement; DGEGS memory occupation is in fact determined only by the number of considered genes, and is independent from the number of Microarray experiments in the training set.

III. DGEG BASED CLASSIFICATION

DGEGs are an excellent data structure for building efficient classifiers. The classifier presented in this paper provides what we call a *Proximity Score* (PS) between a DGEG representing a given pathology ($DGEG_{pat}$), and a DGEG representing a single microarray experiment/sample ($DGEG_{exp}$). The proposed score tries to measure how much $DGEG_{exp}$ is similar to $DGEG_{pat}$ in terms of expressed/silenced genes (analyzing the CEC of the nodes), and relationship between gene expressions (considering the weight of each edge).

The classification rule is therefore implemented as a weighted comparison between the two graphs and is computed according to Eq. 4 where SMS identifies a *Sample Matching Score* and MMS identifies a *Maximum Matching Score*.



Extended Gene Expression Matrix

Gene	A		B		C		D	
Tissue	Diseased	Healthy	Diseased	Healthy	Diseased	Healthy	Diseased	Healthy
Exp1	**5000**	10	20	**11100**	*15000*	*80*	90	**13000**
Exp2	**8000**	20	20	**1200**	*1000*	*1050*	100	**12000**
Exp3	*10000*	*10099*	30	**30000**	**11000**	30	40	**1900**
Exp4	**1200**	20	*15*	*10*	*10*	*100*	**8000**	50
Exp5	**5000**	100	20	**4500**	**10500**	30	**12500**	70
Exp6	**7000**	15	70	**5500**	*10100*	*10050*	40	**12500**

Differential Expression (DE) Threshold = 100
Gray values are considered 0

Gene	A	B	C	D
	DE	DE	DE	DE
Exp1	4990	-11080	14920	-12910
Exp2	7980	-1180	-50	-11900
Exp3	-99	-29970	10970	-1860
Exp4	1180	5	-90	7950
Exp5	-4900	-4480	10470	12430
Exp6	6985	-5430	50	-12460
EC Tot	**3**	**-5**	**-2**	**-2**

Fig. 1. DGEG Construction Example: initial training set

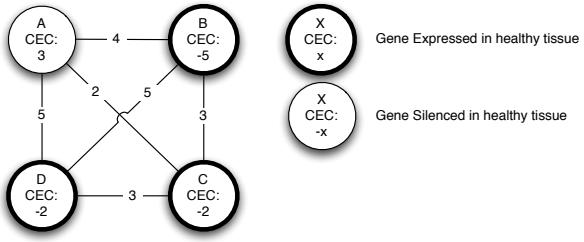


Fig. 2. DGEG Construction Example

$$PS = SMS/MMS \tag{4}$$

The Sample Matching Score analyzes the similarity of $DGEG_{pat}$ and $DGEG_{exp}$ considering only those vertexes (genes) available in both graphs. It is computed as:

$$SMS = \sum_{\forall (i,j) \in E_{exp} \bigwedge E_{pat}} \left(Z_i \cdot w_{i,j} \cdot \frac{|Z_i|}{|Z_i| + |Z_j|} \right) + \left(Z_j \cdot w_{i,j} \cdot \frac{|Z_i|}{|Z_j| + |Z_j|} \right) \tag{5}$$

where (i, j) are edges appearing both in $DGEG_{exp}$ and $DGEG_{pat}$ and Z_x is the Z-Score of vertex x computed as:

$$Z_x = CEC_{x_{pat}} \cdot CECx_{exp} \tag{6}$$

By construction, each vertex x in $DGEG_{exp}$ has a Cumulative Expression Count ($CEC_{x_{exp}}$) equal to -1 (if the gene is expressed in the healthy sample but silenced in the diseased one), 0 (if the gene expression is the same in both samples), or $+1$ (if the gene is expressed in the diseased sample and silenced in the healthy one).

The purpose of the Z-Score is to quantify to what extent the expression of gene x in $DGEG_{exp}$ is *"similar"* to

the expression of the same gene in the training dataset represented by $DGEG_{pat}$. The more genes have a positive Z-Score, the higher will be the similarity, and therefore the SMS, of the sample with respect to the considered $DGEG_{pat}$. The Z-Score may assume the following values:

- > 0 if both $DGEG_{exp}$ and $DGEG_{pat}$ have the gene differentially silenced or expressed;
- < 0 if $DGEG_{exp}$ has the gene differentially silenced and $DGEG_{pat}$ has it differentially expressed, or viceversa.

The two terms of Eq. 5 are the Z-Scores of each gene multiplied by a portion of the weight of the considered arc. This portion is computed as the percentage of the Z-Score of the gene w.r.t. the total Z-Score of the pair.

The Maximum Matching Score is the maximum SMS that would be obtained with all genes in $DGEG_{exp}$ perfectly matching all genes in $DGEG_{pat}$ with Z-Score of each gene always positive. It is computed as:

$$MMS = \sum_{\forall (i,j) \in E_{pat}} \left(w_{i,j} \cdot \frac{CEC_i^2 + CEC_j^2}{|CEC_i| + |CEC_j|} \right) \quad (7)$$

IV. Experimental results

The experimental results presented in this paper focus on assessing the effectiveness of the DGEG based classifier proposed in Section III. Three different training sets of microarray experiments have been used to generate three DGEGs for three different phenotypes.

A. Data source and Training Dataset

The training datasets used for the experimental design come from the cDNA Stanford Microarray database [36]. This source contains a large amount of data that in many cases refers to old experiments done on first generations of microarrays affected by probe sensing problems, reduced gene-set, and lack of UnigeneID for many spots. All genes without valid UnigeneID have been discarded. Moreover, since old microarrays duplicated spots in order to have more reliable results, during the DGEG generation we considered differentially expressed those genes expressed in at least one of their copies on the microarray.

Even if the model allows the use and combination of data coming from different types of microarrays embedding different genes, in our experimental design we considered samples with the same microarray technology and gene sets.

We considered three different data sets:

1) Diffuse Large B-Cell Lymphoma (Bcell);
2) Lymphocytic Leukemia Watch&Wait (CLLww);
3) Lymphocytic Leukemia (CLL).

The Bcell data set is a group of 53 microarrays related to the Diffuse Large B-Cell Lymphoma (a non-Hodgkin Lymphoma disease), it produces a DGEG of 6031 correctly named (using UniGeneID) nodes (12% reduction w.r.t. [34]). The CLLww data set is a group of 22 microarrays focusing on Lymphocytic Leukemia Watch&Wait. From this set we extracted valid information for 6324 genes (17% reduction w.r.t. [34]). Finally, the third dataset (CLL) targeting Lymphocytic Leukemia is a group of 12 experiments from which we were able to extract valid information for 5370 genes (22% reduction w.r.t. [34]).

The threshold T_{diff} used to extract the differentially expressed genes has been set to 300. The procedure to define the threshold is quite easy and requires plotting all the differential expressions of the genes in the dataset, and then setting a threshold that allows excluding the desired level of noise.

B. Classifier

To verify the usability of the proposed model for sample-based classification algorithms, we applied the classification procedure described in Section III using, as samples, six different sets of microarrays data downloaded from Stanford Microarrays Database. None of these samples were part of the training sets used to generate the DGEGs. Each set contains from 8 to 11 distinct samples. We used the three datasets described in section IV-A as classes in which to classify the samples.

Each sample set targets a different phenotype:

1) Chronic Lymphocytic Leukemia - Untreated Watch&Wait;
2) Lymphoma Classification - Hematopoietic cell lines;
3) Lymphoma Classification - Normal Lymphoid subset;
4) Lymphoma Classification - CLL;
5) Solid tumor Ovarian;
6) 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.

The implemented algorithm correctly classified 100% of the samples. This means that we never had a false positive/negative classification. We want to point out that the classifier is not based on abstract statistical methods, but it works by analyzing and somehow measuring the similarities between the two graphs. It is also very interesting to note that, using DGEGs and the Proximity Measure, the classifier is not forced to provide a result. For example, the samples of "Solid tumor Ovarian" have all been scored with a negative or null PM, meaning the classifier correctly decided that the given samples were not part of any class (see Fig. 3).

Fig. 3 shows the result of the classification for the six pathologies, where for each sample set we report the average Proximity Measure against the three datasets.

The classification results can be analyzed as follows:

- Pathology #1 is correctly matched with the CLLww dataset and also has a high score when compared with the CLL dataset. This result is acceptable since the two datasets represent very similar diseases;
- Pathology #2 is correctly classified as distant from all three datasets;
- Pathology #3 has a very low score for all datasets. It is correctly classified as distant from all datasets, but, being anyway a Lymphoma related disease, it also shows a slight similarity with all three datasets;

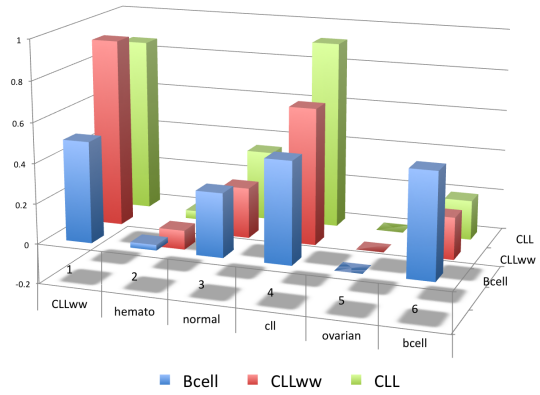


Fig. 3. Classifier results for all samples

- Pathology #4 is correctly classified as a Lymphocytic Leukemia disease; a detailed view of the classification of all 8 samples is in Fig 4;
- Pathology #5, which is a solid tumor, is correctly classified as highly different from the three datasets;
- Pathology #6 is correctly classified as a BCell Leukemia.

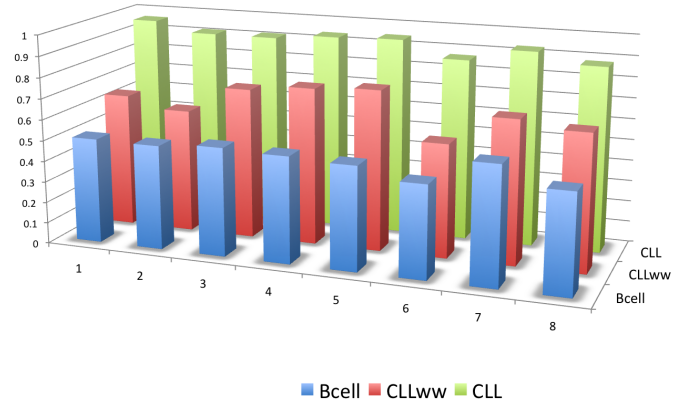


Fig. 4. Classification of the 8 samples of CLL. All samples are correctly classified

The most important result in our opinion is the ability of this classifier to correctly process and identify samples that show very similar pathologies. This result seems to suggest that the DGEG is able, by construction, to give more weight to genes and gene relationships that unequivocally identify a particular pathology. Obviously this ability also depends on the quality of the training set, but this is a common problem to all supervised classification methods.

Finally, we need to introduce a few considerations about the influence of the differential threshold used to build the graph (T_{diff}) on the quality of the classification. Table I shows the Proximity Measure produced by the classifier for seven samples of *"Lymphoma Classification - CLL"* considering DGEGs built with four different threshold. The possible range of expression in the considered training dataset and samples is between 0 and 60,000, but with more than 95% of the expression values falling between 0 and 2,500. The result of the table clearly shows the reduced influence of the threshold on the classification process. In all the experiments the classifier was able to weight with an higher score the CLL dataset (the correct classification for the used samples), regardless the used threshold. Moreover, in all cases the proximity score shows a value higher then 65% providing a good confidence in the result. The only exception concerns sample 1. For this sample, with a threshold of 10000, even if the classifier is able to distinguish between the three classes, the proximity score is very low. Actually this result is not a fault of the classifier. The considered sample has many spots not correctly readable including most of the genes selected as relevant by the DGEG (see Fig. 5).

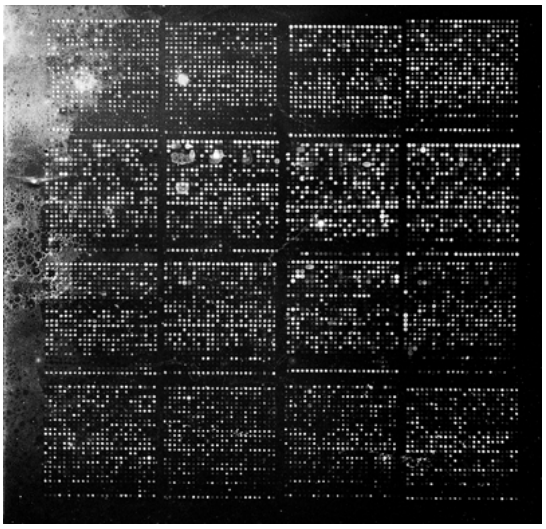


Fig. 5. Bad microarray sample

V. Conclusions and Future Works

In this paper we presented the Differential Gene Expression Graph, a new data structure designed for the analysis of gene expression data in microarrays experiments. To demonstrate the flexibility of the data model we implemented a classifier based on the analysis of topological information extracted from the DGEGs. The full potential of this new 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. In particular our work is now mainly focused on:

- Improving the classification algorithm; this can be done by defining heuristics to remove the "noise", i.e., those genes or gene relationships that are not correlated with the identification of the disease. Noise reduction will create more robust and reliable DGEGs to be used as classifiers;
- Defining a feature extraction algorithm. This task is expected to provide very good results since, by construction, groups of genes which create strong differences between healthy and diseased samples are cliques in the graph.
- Extending the DGEG approach to single-channel Microarrays. In this case the problem is to identify a pairing between healthy and diseased samples that in

Sample#	Treshold	B-Cell	CLLw&w	CLL	Sample#	Treshold	B-Cell	CLLw&w	CLL
0	30	0.421512	0.637974	0.945681	4	30	0.518139	0.795215	0.939897
	100	0.472778	0.651784	0.945307		100	0.507771	0.823667	0.951198
	5000	0.287631	0.490266	0.887320		5000	0.200811	0.367865	0.739417
	10000	0.070065	0.216501	0.650009		10000	0.136883	0.321477	0.624720
1	30	0.464888	0.605682	0.935293	5	30	0.303774	0.455125	0.721196
	100	0.506362	0.629990	0.927602		100	0.381741	0.518469	0.805219
	5000	0.216905	0.335258	0.676495		5000	0.333166	0.437727	0.819134
	10000	0.010301	0.098026	0.188105		10000	0.105602	0.366950	0.679881
2	30	0.428275	0.671583	0.756701	6	30	0.524386	0.635336	0.903806
	100	0.490301	0.737127	0.867379		100	0.523178	0.667139	0.903260
	5000	0.246763	0.447746	0.806067		5000	0.447694	0.591003	0.937333
	10000	0.116466	0.458306	0.848801		10000	0.262381	0.511725	0.864490
3	30	0.414909	0.675115	0.727648	7	30	0.406475	0.637323	0.722645
	100	0.459270	0.735869	0.855962		100	0.455502	0.671919	0.811719
	5000	0.311963	0.558636	0.936386		5000	0.242996	0.441234	0.827963
	10000	0.157106	0.379690	0.769671		10000	0.202751	0.311736	0.608221

TABLE I

INFLUENCE OF T_{diff} IN THE CLASSIFICATION OF THE LYMPHOMA - CLL

our method is necessary to compute the differential expression of each gene. To accomplish that, we are investigating the possibility of creating a reference "healthy expression profile", constructed from a dataset of Microarray experiments on healthy samples.

REFERENCES

[1] G. Gibson, "Microarray analysis," *PLoS Biology*, vol. 1, no. 1, pp. 28–29, Oct. 2003.

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

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

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

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

[6] A. Albrecht, S. Vinterbo, and L. Ohno-Machado, "An epicurean learning apprif6roach to gene-expression data classification," *Artif Intell Med*, vol. 28, pp. 75–87, 2003.

[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, pp. 707–718, 2004.

[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, pp. 240–248, 2004.

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

[12] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, and M. Angelo, "Multiclass cancer diagnosis using tumor gene expression signatures," in *Proc Natl Acad Sci 98*, 2001, pp. 15 149–15 154.

[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, pp. 813–823, 2002.

[14] H. Zhang, C. Yu, and B. Singer, "Cell and tumor classification using gene expression data: construction of forests," in *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, pp. 1131–1142, 2001.

[16] L. Li, C. Weinberg, and T. Darden, "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, no. 12, pp. 1131–1142, 2001.

[17] R. Durbin, S. Eddy, and A. Krogh, "Biological sequence analysis: Probabilistic models of proteins and nucleic acids," *Cambridge University Press*, vol. 16, 1998.

[18] B. Gary, D. Fogel, and W. Corne, *Evolutionary Computation in Bioinformatics*. Morgan Kaufmann, 2002.

[19] P. Frasconi and R. Shamir, "Artificial intelligence and heuristic methods in bioinformatics," *NATO Science Series: Computer and Systems Sciences*, vol. 183, 2003.

[20] D. Higgins and W. Taylor, "Bioinformatics. sequence, structure, and databanks," *Oxford University Press*, 2000.

[21] D. Husmeier, R. Dybowski, and S. Roberts, "Probabilistic modeling in bioinformatics and medical informatics," *Springer Verlag*, 2005.

[22] A. Jagota, "Data analysis and classification for bioinformatics," *Bioinformatics by the Bay Press*, 2000.

[23] T. Jiang, X. Xu, and M. Zhang, "Current topics in computational molecular biology," *The MIT Press*, 2002.

[24] B. Scholkopf, K. Tsuda, and J. Vert, "Kernel methods in computational biology," *The MIT Press*, 2004.

[25] U. Seiffert, L. Jain, and P. Schweizer, "Bioinformatics using computational intelligence paradigms," *Springer Verlag*, 2005.

[26] J. Wang, M. Zaki, and H. Toivonen, "Data mining in bioinformatics," *Springer-Verlag*, 2004.

[27] C. Wu and J. McLarty, "Neural networks and genome identification," *Elsevier*, 2000.

[28] P. Larranaga and E. Menasalvas, "Special issue in data mining in genomics and proteomics," *Artificial Intelligence in Medicine*, vol. 31, no. III-IV, 2003.

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

[30] 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, 2004.

[31] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman, "Missing value estimation methods for dna microarrays," *Bioinformatics*, pp. 520–525, 2004.

[32] A. Hill, E. Brown, M. Whitley, G. Tucker-Kellog, C. Hunter, and S. D., "Evaluation of normalization procedures for oligonucletide array data based on spiked crna controls," *Genome Miology*, vol. 12, no. 12, 2001.

[33] J. Schuchhardt, D. Beule, A. Malik, E. Wolski, H. Eickhoff, H. Lehrach, and H. Herzel, "Normalization strategies fo cdna microarrays," *Nucleic Acids Research*, vol. 28, no. 10, 2000.

[34] A. Benso, S. Di Carlo, S. Politano, and L. Sterpone, "A graph-based representation of gene expression profiles in dna microarrays," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Sept. 2008.

[35] Unigene. [Online]. Available: http://www.ncbi.nlm.nih.gov/sites/entrez?db=unigene

[36] cdna stanford's microarray database. [Online]. Available: http://genome-www.stanford.edu/