Intelligent Data Analysis 20 (2016) S53–S67
DOI 10.3233/IDA-160845
IOS Press

Improving pattern classification of DNA microarray data by using PCA and logistic regression

Ricardo Ocampo-Vega[a], Gildardo Sanchez-Ante[a,*], Marco A. de Luna[a], Roberto Vega[a], Luis E. Falcón-Morales[a] and Humberto Sossa[b]
[a]*Data Visualization and Pattern Recognition Lab, Tecnológico de Monterrey, Campus Guadalajara, Zapopan, México*
[b]*Instituto Politécnico Nacional-CIC, México, Distrito Federal, México*

Abstract. DNA microarrays is a technology that can be used to diagnose cancer and other diseases. To automate the analysis of such data, pattern recognition and machine learning algorithms can be applied. However, the *curse of dimensionality* is unavoidable: very few samples to train, and many attributes in each sample. As the predictive accuracy of supervised classifiers decays with irrelevant and redundant features, the necessity of a dimensionality reduction process is essential. The main idea is to retain only the genes that are the most influential in the classification of the disease. In this paper, a new methodology based on Principal Component Analysis and Logistics Regression is proposed. Our method enables the selection of particular genes that are relevant for classification. Experiments were run using eight different classifiers on two benchmark datasets: Leukemia and Lymphoma. The results show that our method not only reduces the number of required attributes, but also increase the classification accuracy in more than 10% in all the cases we tested.

Keywords: DNA microarray, feature reduction, principal component analysis, logistic regression

1. Introduction

Cancer is an important health issue worldwide. Up to date, more than 200 types of cancer have been identified and according to the National Cancer Institute (NCI) [29], there were 8,689,771 cases of cancer reported just in the United States from 1973 to 2014. From those, 7,813,979 are malignant. For a patient to receive the appropriate treatment, the clinician must identify as accurately as possible the cancer type. Although biopsy is still a standard diagnostic method, other techniques from molecular biology are becoming more common. One of such techniques is the DNA Microarrays.

A DNA microarray is a collection of microscopic DNA spots distributed on a solid surface. There could be several thousands of spots on one single microarray. Each spot contains multiple copies of identical strands of DNA. Each spot has a unique DNA sequence, representing one gene. The material on the microarray is put in contact with cells or tissue from the subject that is going to be analyzed. Doing so may activate the expression of certain genes, which in turn will produce different fluorescent

*Corresponding author: Gildardo Sanchez-Ante, Data Visualization and Pattern Recognition Lab, Tecnológico de Monterrey, Campus Guadalajara, Av. Gral Ramon Corona 2514, Zapopan, Jal, 45201, México. E-mail: gildardo@itesm.mx.

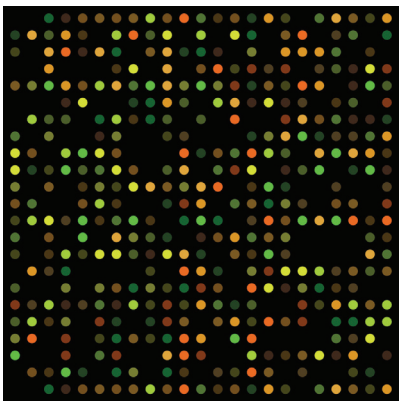

Fig. 1. An example of a microarray that uses two different color dyes (Wikimedia Commons, Guillaume Paumier).

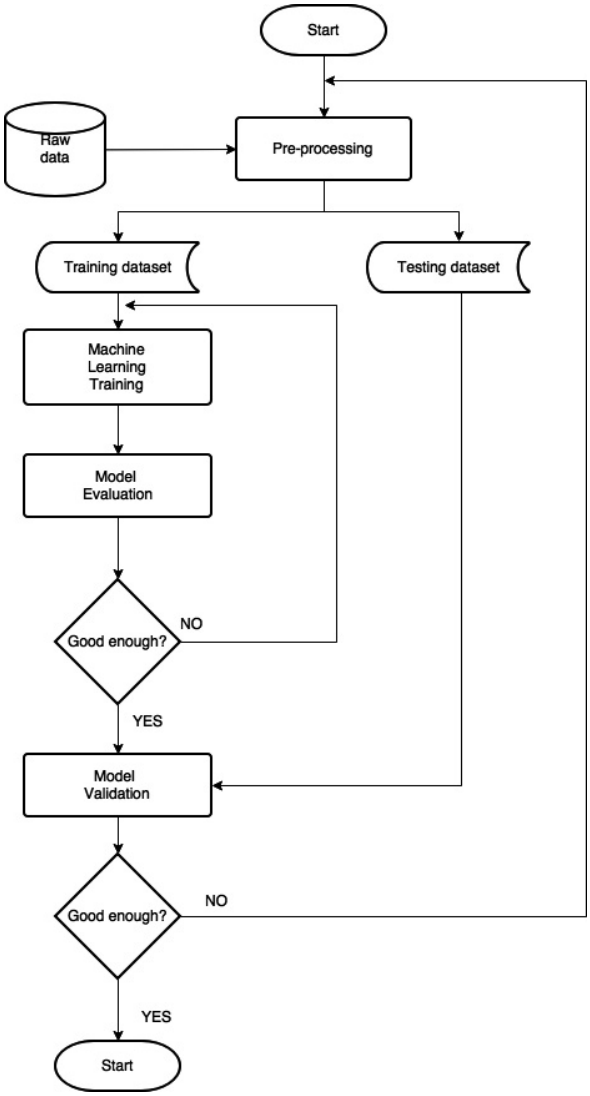

Fig. 2. A simple flowchart of the steps required for supervised learning.

colors (this is a process called hybridization). After hybridization, fluorescence measurements are made with a microscope that illuminates each DNA spot and measures fluorescence for each dye separately; these measurements are used to determine the ratio, and in turn the relative abundance, of the sequence of each specific gene in the two mRNA or DNA samples [6]. Microarray images typically contain several thousands of small spots, each of which represents a different gene in the experiment. Figure 1 shows a simplified representation of one such microscope slides.

One of the main advantages of using DNA microarrays over traditional diagnostic methods is that given that each microarray contains several thousands of spots, it is equivalent to run several thousands of experiments at the same time. This provides a huge amount of molecular information. Such information not only allows the classification of tumor samples into known categories, but also helps to discover new classes, as well as new diagnostic and therapeutic markers [5]. Different approaches have been

considered for the analysis of microarray data, including some of the most well known machine learning algorithms [10]. We will cover some of them later on in the previous work section. In most of the cases the problem has been considered from a *supervised learning* perspective. This means that we are given a training set of items, represented through a vector of *features*, and from there a model is obtained to perform classification tasks [34]. Figure 2 shows a diagram of this process.

Generally speaking, the process to train a classifier to identify patterns may include the selection of appropriate features. This is particularly important in the case of DNA Microarray data. Just to give an example, let us consider the Leukemia cancer dataset [21], which consists of 72 samples, characterized by 7129 genes, or features. The number of samples is two orders of magnitude smaller than the number of features. Almost all of the datasets available up to now are like that. The Small Round Blue Cell Tumors (SRBCT) dataset includes 88 samples each with 2308 genes [32]. The Lymphoma dataset is composed by 62 samples, with 4026 genes [1]. This fact represents a challenge for many of the classifiers. It implies serious limitations on the statistical significance of the results of basically any pattern recognition method.

This has been called the *curse of dimensionality*. This concept is used to describe problems where there is a huge number of attributes and just a few samples [3]. The main issue is that when the dimensionality increases, the volume of the search space grows exponentially, making that the available data becomes sparse. One of the possible approaches to deal with the curse of dimensionality is called *feature selection* or *feature reduction*. In general terms, *feature selection* consists in finding ways to reduce the dimensionality n of the feature space F, to reduce the risk of over-fitting as well as to allow efficient computation in the classifier. This is a main topic in general machine learning research since some time ago [34]. The approaches try, by different means, to identify and retain those attributes that better represent the original information contained in every sample of a dataset. In other words, the idea is to retain a subset F^* of F such that $\|F^*\| \ll \|F\|$ and that the elements of F^* still represent F reasonably well. Considering that, we will see that it is common that many of the different approaches to develop classifiers for DNA Microarray data consider also a feature selection step, combined with the selection of a particular classifier.

In this paper, a new methodology that combines some well known statistical methods to support feature reduction is introduced. By doing so, we aim not only to decrease the number of attributes. We want to retain information about which particular genes were chosen. To test how general our proposed method is, we ran experiments using two DNA microarray datasets and eight different classifiers. We then compare our results with others reported in the literature. The results show an increase in the classification accuracy of at least 10%, with a comparable number of attributes. The main difference of our method, compared with others is that we can individually identify which genes were selected. Besides, among the classifiers, we report the use of a Lattice Neural Network with Dendritic Processing (LNNDP).

We first introduced LNNDP to DNA microarray classification tasks in [48]. When using that particular classifier, there is no constraint on the number of classes. More over, for the classification step, no human intervention is required for the adjustment of parameters. The results show that LNNDP is competitive in performance compared with other approaches. The remainder of the paper is organized as follows: Section 2 describes previous work on DNA microarray analysis. Section 3 presents our methodology. Section 4 describes experiments and results and finally, Section 5 presents the conclusions and future work.

2. Previous work

The interest to develop automated methods for classification of biological samples obtained from DNA microarrays is increasing rapidly. It has been considered that in order to find a good solution to the classification of DNA Microarray data, two related directions can be explored. One is to build or to apply powerful classifiers. The other is to reduce the amount of features (genes) that are used to induce a model in the classifier. Both are important and in the following paragraphs we will go through some of the most relevant advances in both directions.

2.1. Classifiers

In general, the use of machine learning, statistics and other artificial intelligence techniques is considered as a good opportunity to solve problems in the area of DNA microarray data analysis. Researchers have reported the use of Support Vector Machines (SVM), like in [12], as well as in [19], or in [45]. Artificial Neural Networks (ANN) have also been explored by authors such as: [28,38,39]. Methods bases on evolution such as Genetic Algorithms (GA) are also reported in [14], and in [16], to cite a couple. Other related methods, such as Ant Colony Optimization (ACO) are reported by [9]. Artificial Bee Colony (ABC) has been applied by [20], as well as some hybrid approaches, like the ones described in [8,27].

Truth is that each one of the classifiers has advantages and disadvantages. Some authors have conducted comparative studies, like the ones reported in [35,43,56]. Authors such as [63] suggest that SVM behave well in general for this problems, although it is important to consider that such method was originally developed for bi-class problems, and applying SVMs to multi-class classification implies a number of additional steps, that might not be easy to accomplish, as described in [54].

The work in [19] reports the application of SVMs for the classification of ovarian cancer data. The authors not only apply the SVM to the raw data, but also consider a feature selection method described in [21]. From their experiments, they conclude that although SVMs are able to deal with such kinds of problems, they shown a precision similar to other approaches.

In a different approach, Guyon et al. [22] also report the application of SVMs, but they use them as a tool to perform feature ranking. The experiments were run on the Leukemia and Colon datasets. One of the most interesting findings of their work is that the selection of features might be even more important than the classifier used. By using a Recursive Feature Elimination, they retain the subsets of genes that had the highest informative density. The comparison of classifiers such as Linear SVM, Linear Discriminant and the one introduced in [21] did not show significantly different performance. But the performance of all was affected by the genes used.

In the paper by Cho et al. [10], the authors tested 42 different combinations of feature selection methods and classifiers. The feature selection methods they considered were: Euclidean distance, Pearson's and Spearman's correlation coefficients, cosine coefficient, information gain, mutual information and signal to noise ratio. As for the classification methods, they used multi-layer perceptron (MLP), k-nearest neighbor (KNN), support vector machine and structure adaptive self-organizing map (SOM). The experiments were run on datasets for Leukemia, Colon Cancer and Lymphoma. In their experiments they found that information gain and Pearson's correlation coefficient were the best feature selection methods, while MLP and KNN were the best classifiers, reaching up to 97% accuracy on the classification. In [12] the authors apply an SVM, and four feature reduction methods: principal components analysis (PCA), class-separability measure, Fisher ratio, and t-test. The datasets used were: SRBCT, Lymphoma and Leukemia. In this case, they determined how many genes each classifier required to reach a 100%

precision, being the MLP the one needing the most (96 for SRBCT), and SVM with polynomial kernel the smallest (6 for the same dataset). As for the feature selection part, they found that t-test was the best in their experiments.

In [11] the authors compared the performance of three algorithms: artificial neural networks, decision trees (DT) and logistic regression (LR) and two composite models of DT-ANN and DT-LR. The DT models exhibited the lowest predictive power and the poorest extrapolation when applied to the test samples. The ANN models displayed the best predictive power and showed the best extrapolation. SVMs were originally developed for bi-class problems.

Some authors have worked to extend the applicability of SVMs to multi-class problems [44,47]. In [63], the authors deal with multi-class imbalance classification problems. The method divides multi-class problems into multiple binary-class problems. After this, one of two correction techniques is used to tackle the class imbalance problem. Finally, a novel voting rule is applied. The experimental results demonstrated that the proposed method outperforms any traditional classification approaches because it produces more balanced and robust classification results. A recent approach for classification, called Extreme Learning Machine (ELM) has been also tried [50,65]. It seems that for the set of problems considered, the ELM algorithm shows higher or similar correct classifications for most of the classes compared with other algorithms.

One of the most common classifiers is the Naïve Bayes (NB). Although it is frequently outperformed by other more recent methods, it is very efficient [26]. The method is based on Bayes theorem, with strong independence assumptions. The classifier learns from training data the conditional probability of each attribute A_i given the class label C. Then, when used for classification, the Bayes rule is applied to compute the probability of C given a particular instance of attributes $A_1, \ldots, A_n$. A Naïve Bayesian model is easy to build, and compared with other methods that requiere the iterative determination of parameters, Naïve Bayes does not need that. That makes this model particularly useful for very large datasets, or in problems where the time to start classifying is constrained.

A related approach is the Bayesian Networks (BN). Bayesian networks are part of what is called probabilistic graphical models. These models use directed acyclic graphs to represent knowledge about uncertain environments. In that way the network represents a probability distribution. Given the data on the attributes, the model computes the probability of that pattern belonging to each one of the classes, and then simply gets the maximum value to decide for the class. More details can be found in [18].

In the work by [30], the authors compared the efficiency of several feature selection methods. Some of those methods are very well known in the field: Significance analysis of microarrays (SAM), analysis of variance (ANOVA), Area under the receiver operating characteristic (ROC) curve and various others. The tests were performed over 9 different bi-class microarray datasets. Some of the results they report is that there was little consistency in the genes selected by the different methods. Only 8 to 21% of the genes where in common. In general, their findings are that the classification is substantially influenced by many variables, including the feature selection method, the number of genes, noise and other variables.

In [15] the authors propose a Minimum Redundancy-Maximum Relevance (MRMR) feature selection framework. Under this approach the idea is to choose features that are maximally dissimilar to each other. This is a minimum redundancy criteria that can be combined with the traditional maximum relevance, such as the maximal mutual information. The authors test the method with 6 different datasets and employ Naïve Bayes, linear discriminant analysis, logistic regression and SVM classifiers. The results suggest that by applying this process, the accuracy of the classifier is better that when using all the features. In a somewhat related approach, the authors in [41] develop an MRMR method using what they call the normalized mutual information, to evaluate the redundancy and relevance. The method was applied to three datasets, obtaining compact feature representations with high accuracy.

2.2. Feature selection

As it was mentioned before, the DNA microarray data has some interesting properties: there are just a few samples to characterize a disease, and those samples are composed by a large number of features or attributes. Attempting to analyze directly such information is not a good idea. Thus, methods to reduce the amount of features considered are usually applied before using the information to train a classifier. The methods for feature selection can be classified in: filter, wrapper and hybrid approaches [59]. Filter methods use some measure to rank the attributes based on univariate functions, and then, the best ranked attributes are selected [40]. Wrapper methods are usually multivariate and they involve also a learning algorithm to evaluate the sets of attributes [33,51]. Hybrid methods are combinations of the former two.

Usually, the approach to try first is the filter, since it is simple to run and requires $O(n)$ time. However, the main disadvantage is that it creates redundancy and evaluates attributes based only on their individual scores [52]. Measures that have been used with DNA microarray data include: Pearson's and Spearman's correlation coefficients, Euclidean distance, information gain [10], and t-test [57], among others [60,61].

Works like [60] also tested wrappers. In this case, by considering a correlation-based feature selection (CFS) in conjunction with J48 and naïve Bayes. In this particular case, the authors mention that both approaches arrived at similar results. In [49], the authors present an excellent comparison of classifiers and feature selection methods. According to their findings, the choice of feature selection methods and the number of genes substantially influence classification success. In [17], a Minimum Redundancy-Maximum Relevance (MRMR) filter is used in combination with a Genetic Algorithm. Shah et al. [53] introduce a formulation that includes the task of feature selection as well as classification. The problem is formulated as to find the optimal classifier.

The work in [4] presents changes to logistic regression method and apply it in three cancer classification problems with microarray data. In the paper by Zhou et al. [66] the authors propose a Bayesian approach to gene selection and classification using the logistic regression model. They evaluated the proposed method against several microarray data sets. Other authors have applied logistics regression in DNA microarray problems, like the ones reported in [2,36,37,42,46,64].

Techniques from computational intelligence have also been analyzed. For instance, in [62] the authors report the use of ant colony optimization (ACO) to select relevant genes. In [58] the authors develop a hybrid genetic algorithm-neural network (GANN) model that performs feature selection over raw microarray data.

3. Methodology

The methodology proposed in this paper is aimed at reducing the cardinality of the vectors that will be used to feed the machine learning algorithm. In that sense, a traditional approach is to apply Principal Component Analysis (PCA). PCA compresses the information contained in a number p of original variables into a smaller set of q factors [23]. Each factor is a linear combination of all the p original variables. Therefore, PCA does not actually reduce the number of attributes, it only creates a different representation of the same data. In our problem it means that if we use the n principal components find through PCA analysis, although we would be working with only n features, at the end, all k original genes are involved, so, no real gene reduction in the process.

Thus, our methodology uses PCA as an intermediate step, but incorporates other operations such that the actual number of attributes (genes) are reduced. When this process is performed, the user can identify the specific genes needed for the classification. This is an important aspect of our contribution.

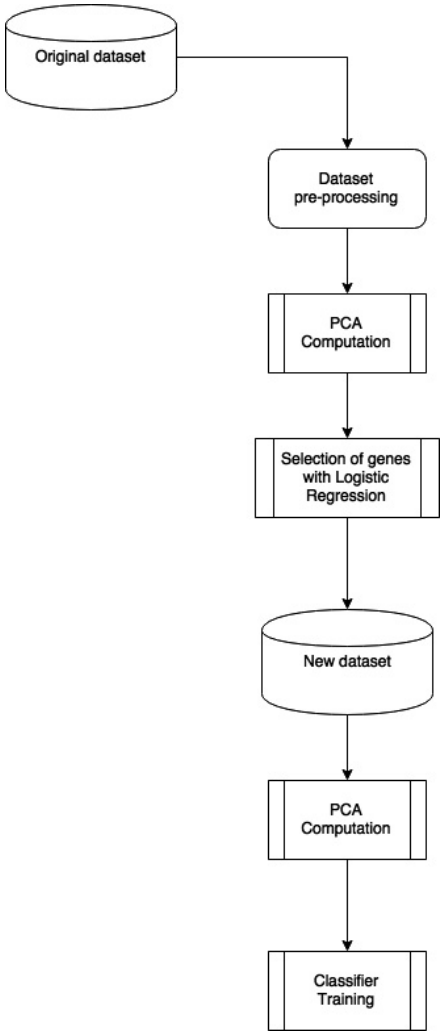

Fig. 3. The methodology proposed in this work. The intermediate computation of PCA will allow the identification of the most significant elements to be retained.

The methodology can be summarized in the following main steps: 1) Preparation of dataset, 2) Computation of PCA on the original dataset, 3) Selection of attributes (genes) using logistic regression, 4) Creation of a new dataset containing only the attributes chosen in step 3, 4) Computation of PCA on the new dataset and finally in 5) Train the classifier using those components. Figure 3 shows the process. In the following subsections we provide more details on each of the steps.

3.1. Datasets

We used two publicly available, bi-class datasets in order to test our proposed methodology: The Leukemia dataset [21] and the Lymphoma dataset [13]. Each one contains 7,129 genes, or attributes. The Leukemia dataset consists of 72 samples: 25 samples of *acute myeloid leukemia* (AML) and 47 samples of *acute lymphoblastic leukemia* (ALL). The dataset is divided into two subsets: training set with 38 samples, and test set with 34 samples.

The Lymphoma dataset contains a total of 77 samples. We randomly divided it in a training set with 22 samples (11 of each class), and a test set with the remaining 55 samples.

3.2. Principal component analysis

Principal Component Analysis, or simply PCA is a statistical technique that allows the identification of the principal directions in which the data varies. The basic idea behind PCA is that, unless there is perfect correlation between two or more of the variables, p principal components are required to account for the p-dimensional variable space. PCA replaces the p original variables by a smaller number, q, of derived variables, the principal components, which are linear combinations of the original variables. Often, it is possible to retain most of the variability in the original variables with q much smaller than p. PCA projects p-dimensional data into a q-dimensional sub-space ($q \leqslant p$) in a way that minimizes the sum of squared distances from the points to their projections. In computational terms, the principal components are found by calculating the eigenvectors and eigenvalues of the data covariance matrix. This process is equivalent to finding the axis system in which the co-variance matrix is diagonal. The eigenvector with the largest eigenvalue is the direction of greatest variation, the one with the second largest eigenvalue is the (orthogonal) direction with the next highest variation and so on. More details on PCA and its computation can be found, for example, in [31].

3.3. Logistic regression

Logistic regression is a multivariate statistical technique used to model the relationship between a binary dependent variable and one or more independent variables(continuous or discrete variables). The difference with linear regression is that in that case, the response variable is continuous, while in logistic regression it is a categorical variable (0 or 1). The Logistic Regression Model, called binary logit regression, is described by:

$$\ln\left(\frac{\pi_j}{1-\pi_j}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

where β_0 is the intercept from the linear regression equation, X_i represents the i-th independent variable $i = 1, 2, \ldots, n$, β_i represents its corresponding coefficient, $\ln(\frac{\pi_j}{1-\pi_j})$ is the logit transformation used with the dependent variable of the j-th sample $j = 1, 2, \ldots, m$, and π_j is the success probability.

The way in which logistic regression works is by estimating the probability of an event occurring. It focuses on identifying the independent variables that explain the observed variation in the dependent variable. The probability of an event occurring is described by:

$$\pi_j = \frac{\exp^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n}}{1+\exp^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n}}$$

A more detailed explanation of logistic regression is available, for instance, in [23].

3.4. Proposed method

Let us consider that we have a training set composed of p pairs (indexed by i running from 1 up to p), which can be represented by $\mathcal{T} = \{(\vec{x}_i, y_i)\}_{i=1}^{p}$. The $\vec{x}_i$ are the attributes (expression levels of the genes in our case), and the y_i are the labels for the classes.

The process is described in Algorithm 1. This algorithm describes a procedure called SELECT(), that receives the attributes of the dataset, represented by $\{\vec{x}_i\}_{i=1}^{p}$, and a two parameters: ν, an integer number that represents the percentage of the total variance that we would like to retain. This value indirectly defines the number of components to be considered. The second parameter is a threshold μ, given by a real number.

Line 1 in Algorithm 1 applies the PCA on the original dataset. We retain only the q first components that account for the variance given by ν. The components are stored in the structure Q. Each principal component stored in Q has the form: $\vec{q}_i = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n$, where $\vec{q}_i$ is the ith principal component obtained using PCA, x_n is the nth attribute and α_n is its corresponding weight. In Line 2, a logistic regression is applied on those components. Only the d most relevant ones are retained in a new structure Q_1. Then, in Line 3 the d selected components are analyzed to eliminate the attributes whose weights are below μ. Line 4 creates a new dataset $\{\vec{z}_i\}_{i=1}^{p}$. It is important to mention that this dataset will contain the same amount of samples (rows, given by index i), however, the number of attributes (columns) for each sample is going to be smaller, that is, $\|\vec{z}\| < \|\vec{x}\|$. Then, in Line 4 PCA is applied again over the new dataset, to obtain q_2 principal components. Each of those components will be a linear combination of the genes selected in Line 3.

Algorithm 1: SELECT() finds subset of genes to be used in classification.

Input: $\mathcal{T}, \nu \in \mathbb{N}, \mu \in \mathbb{R}$.

Output: Q_2, the set of selected components.

1 $Q \leftarrow \texttt{PCA}(\{\vec{x}_i\}_{i=1}^{p}, \nu)$
2 $Q_1 \leftarrow \texttt{LOGIT}(Q, d)$
3 $\{\vec{z}_i\}_{i=1}^{p} \leftarrow \texttt{RemoveAttributes}(Q_1, \mu)$
4 $Q_2 \leftarrow \texttt{PCA}(\{\vec{z}_i\}_{i=1}^{p}, q_2)$
5 **return** Q_2

3.5. Classification

In order to analyze the effectiveness and generality of the feature selection method that we propose, we run experiments with a total of eight different classifiers. They are: Lattice Neural Network with Dendritic Processing (LNNDP), Support Vector Machine (SVM) with two kernels: linear and radial basis, Extreme Learning Machine (ELM), Bayes Net (BN), Naive Bayes (NB), Multi-Layer Perceptron (MLP) and Radial Basis Function Neural Networks (RBF).

We implemented the LNNDP using the training method proposed by Sossa and Guevara in [55]. One of the advantages of this method is that it requires no parameter configuration, nor random initialization values.

For the case of Support Vector Machines, we used the LIBSVM Library and trained the classifier as suggested in [7]. We used two different kernels: linear and radial basis function (RBF). In order to find the best parameters, we divided the training set in two parts: one for training, and the other for cross validation. The cross validation set was composed of 33% of the elements of the samples in the training set, chosen randomly.

For the Extreme Learning Machine [25], we used the basic ELM implementation of the Nanyang Technological University available at http://www.ntu.edu.sg/home/egbhuang/elm_codes.html. The only parameter to configure in this implementation is the number of neurons in the hidden layer. We used a

Table 1
Coefficients obtained after applying logistic regression to the training set $\mathcal{T}$ of the Lymphoma dataset, where β_i represents the coefficients and $|\beta_i|$ represents its magnitude

β_i	β_i	β_i	β_i	β_i	β_i
β_1	-0.0967	b_6	**0.4252**	b_{11}	0.2309
β_2	0.0452	b_7	0.2781	b_{12}	0.1347
β_3	**0.4591**	b_8	-0.0559	b_{13}	-0.0543
β_4	-0.1747	b_9	0.0275	b_{14}	0.03943
β_5	**0.3073**	b_{10}	**-0.4174**	b_{15}	0.1210

Table 2
Number of genes that are present in different components. The columns with numbers 0, 1, 2, 3 and 4, show the number of times that the genes appeared in the components

Dataset	μ	0	1	2	3	4
Lymph.	0.006	236	1117	2296	2508	972
Leuk.	0.01	1846	2861	2000	422	–

similar methodology than the one used for the SVM. We divided the train set in two subsets and then used cross validation to find the best number of neurons to use. To select the number of neurons we searched in the range $[1, 100]$.

As for Naïve Bayes, Bayes Networks, Multi-Layer Perceptron and Radial Basis Function Neural Networks (RBFNN), we used Weka [24]. In the specific case of RBFNN, we tested using different values for the processing units in the hidden layer, from 2 to 20, in steps of 2. The configuration that showed the best values was the one with 12 units.

4. Experiments and results

We implemented PCA on the normalized training set and selected the q first components needed to retain at least 90% of the variance in order to create a new training set Q with the same amount of samples, but each being represented by q attributes (the chosen principal components). In the case of the Lymphoma dataset $q = 15$, while in the Leukemia dataset $q = 27$. Given that our interest is to identify which of the q components discriminate between the two classes of the dataset, we applied logistic regression over the training set $\{\vec{z}_i\}_{i=1}^{p}$ and, retained the components whose coefficient's magnitude were above a certain threshold μ. Both, μ and ν, the percentage of variance retained were chosen empirically. Table 1 shows in bold the coefficients selected in the Lymphoma dataset (components 3, 5, 6 and 10). A similar procedure was implemented over the Leukemia dataset. In this last case we selected components 3, 11, and 26.

All our experiments were run on a HP Proliant G8 server with 2 Intel processors, 8 cores each and 256 Gb RAM.

Thus, the threshold μ was determined by analyzing the coefficients' magnitude distribution of the selected components based on a box-plot. We set $\mu_{lymphoma} = 0.006$ and $\mu_{leukemia} = 0.01$ and analyzed how many attributes (genes) had a coefficient different from zero in each component and analyzed which of them where present in more than one component. The results are shown in Table 2. By removing the genes with less contribution to the class discrimination capability of the logistic regression, we were able to reduce the dimensionality of the datasets from 7,129 attributes to only 972 for Lymphoma and 422 for Leukemia, which represent 13.63% and 5.91% of the original ones. In addition, these attributes might have biological and medical significance, considering that they represent the actual genes. We then created a new Lymphoma dataset, and a new Leukemia dataset, using only the genes present in all the selected components.

Having reduced the actual number of attributes in the dataset, we run again PCA over the new datasets. After this step, eight different classifiers were trained and evaluated using the first 3, 5, 7 and 15 components. The number of principal components used is represented by q_2. For comparison purposes, we also

Table 3
Comparison of classification precision using PCA over the original Leukemia dataset, and using PCA over the proposed reduced dataset. In bold the highest precision of each method

q_2	LNNDP		SVM RBF		SVM LIN		ELM	
	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.
3	79.41%	76.47%	69.73%	80.15%	70.56%	**85.88%**	66.73%	**79.27%**
5	73.53%	76.47%	71.00%	77.29%	71.15%	78.50%	69.09%	75.47%
7	70.59%	76.47%	72.09%	77.97%	71.74%	79.62%	66.68%	76.77%
15	73.53%	**88.24%**	77.35%	**82.59%**	78.21%	83.24%	72.68%	79.21%

q_2	MLP		Bayes Net		Naive Bayes		RBF 12	
	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.
3	57.91%	**92.42%**	63.48%	**85.78%**	60.21%	**87.47%**	60.14%	**86.82%**
5	59.47%	89.51%	63.17%	85.60%	59.57%	84.20%	61.51%	82.86%
7	57.54%	89.48%	62.97%	84.82%	57.30%	86.20%	60.82%	83.85%
15	55.80%	87.45%	61.25%	83.69%	55.17%	83.33%	59.60%	80.87%

Table 4
Comparison of classification precision using PCA over the original Lymphoma dataset, and using PCA over the proposed reduced dataset. In bold the highest precision of each method

q_2	LNNDP		SVM RBF		SVM LIN		ELM	
	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.
3	74.55%	65.46%	60.49%	50.82%	63.69%	53.64%	64.36%	57.40%
5	69.09%	**83.64%**	64.00%	**65.15%**	70.76%	69.06%	62.47%	66.13%
7	60.00%	81.82%	64.91%	63.67%	72.47%	**73.95%**	64.26%	66.64%
15	61.82%	72.73%	49.75%	59.95%	71.31%	71.22%	67.38%	**68.22%**

q_2	MLP		Bayes Net		Naive Bayes		RBF 12	
	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.	Orig.	Prop.
3	68.76%	87.32%	74.17%	**76.86%**	66.29%	76.29%	66.89%	76.19%
5	64.98%	**91.60%**	73.91%	76.29%	63.74%	**77.14%**	65.55%	78.11%
7	62.62%	87.98%	73.55%	76.16%	64.22%	76.76%	69.80%	**78.80%**
15	57.49%	78.85%	72.64%	75.87%	67.74%	76.40%	72.92%	76.44%

trained and evaluated the eight classification algorithms using the same number of components obtained from the original datasets. That is, with all the 7,129 attributes. Table 3 shows the results for the case of the Leukemia dataset. From the results, it is possible to observe that the best accuracy was achieved with the Multi-Layer Perceptron (94.42%), and the second one was the LNNDP (88.24%). Moreover, MLP required a smaller number of components to reach that precision. It is also important to point out that the results obtained with the method proposed in this work were consistently better than the ones using the original dataset.

In a similar way, Table 4 shows the results for the case of the Lymphoma dataset. In this case, the best classifier was again the MLP (91.60%), and the second one was the LNNDP (83.84%). For this dataset, both methods required the same amount of components to reach their respective best values. As in the Leukemia dataset, the results generated with the proposed method were always better than using the original dataset. These results suggest that classifying the patterns directly after the implementation of PCA yields suboptimal results. This behavior is observed regardless of the dataset and the classifier.

By reducing the number of genes considered, we not only reduce the computational cost of the analysis, but also improved the accuracy of all the tested classification methods. In some cases the improvement was above 10%. Figure 4 summarizes graphically the results. The graph shows the performance

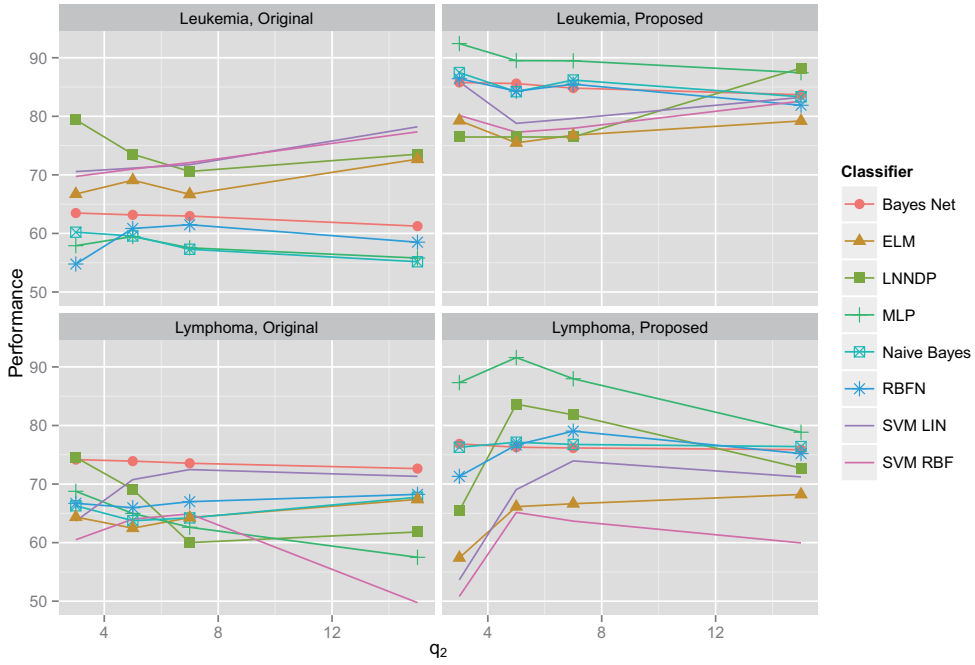

Fig. 4. Performance of the classifiers for Leukemia and Lymphoma for eight classifiers and different values for the selected components q_2.

for the eight classifiers for each dataset and for different numbers of components q_2 used, both from the original dataset and the proposed reduced dataset. From the graphs it is interesting to observe that the performances with the reduced dataset are always better than with the original dataset. However, it also seems that in particular for Lymphoma, increasing the number of components leads to a decrease in the performance. This effect is also seen in Leukemia but it is smaller.

5. Conclusions and future work

Providing computational tools for the processing, analysis and visualization of genomic information is an important current trend. DNA microarrays are being considered as an alternative to better diagnose diseases such as cancer. As long as more genetic information is generated, and analyzed, this technology could be improved. For instance, by reducing the amount of probes (experiments) carried out on a microarray. That could be achieved by analyzing data using techniques like the one described in this work. Of course all these processes need to be consistent with biological knowledge. If results like the ones obtained for this paper have biological meaning, it could imply that microarrays to detect Lymphoma or Leukemia could require to consider just a subset of the original amount of gene expressions. That is cheaper microarrays that could be used in more cases.

When a reduction in the number of variables is needed, one of the first approaches is undoubtedly to use the principal component analysis. By detecting the axis in which the variability of the original information changes the most, it is possible to transform the problem into another one in a new space, where those components are the main axis. However, as it was mentioned before, what happens is that the components are actually just a linear combination of the original variables. So, in reality, no variables are

eliminated, it is just that they are expressed in a different, more compact way. By using our methodology, the process guarantees that the principal components are related with the classes in the problem, and then we look at the components that have a bigger impact in the discrimination step. We show here that by using this methodology, the number of attributes (genes in this case) diminished from 7,129 to less than a thousand attributes in both datasets. At the same time, we even increase the performance of the classifier. Interestingly, we discovered that the most relevant components for the classification were not necessarily the first ones computed by PCA. That fact suggests that using the first k principal components might not be the best option to achieve optimal classification. However, if PCA is applied following the methodology described in this work, it is possible to increase the performance of the classifier without a decrease in the classification rate. Some interesting aspects to consider in future work could be: 1) To make a fair comparison with other feature selection methods, 2) To test the approach with data additional to the public datasets, and 3) To test the performance of this approach with multi-class problems.

Acknowledgments

The authors thank Tecnológico de Monterrey, Campus Guadalajara, for their support under the Research Chair in Information Technologies and Electronics, as well as IPN-CIC under project SIP 2015-1187, and CONACYT under projects 155014 and 65-Investigación en Fronteras de la Ciencia 2015 for the economical support to carry out this research.

References

[1] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick et al., Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, *Nature* **403**(6769) (2000), 503–511.

[2] A. Antoniadis, S. Lambert-Lacroix and F. Leblanc, Effective dimension reduction methods for tumor classification using gene expression data, *Bioinformatics* **19**(5) (2003), 563–570.

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

[4] C. Bielza, V. Robles and P. Larrañaga, Regularized logistic regression without a penalty term: An application to cancer classification with microarray data, *Expert Systems with Applications* **38**(5) (2011), 5110–5118.

[5] J.L. Brewster, K.B. Beason, T.T. Eckdahl and I.M. Evans, The microarray revolution: Perspectives from educators, *Biochemistry and Molecular Biology Education* **32**(4) (2004), 217–227.

[6] P.O. Brown and D. Botstein, Exploring the new world of the genome with DNA microarrays, *Nature Genetics* **21** (1999), 33–37.

[7] C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology (TIST)* **2**(3) (2011), 27.

[8] X.-W. Chen, Gene selection for cancer classification using bootstrapped genetic algorithms and support vector machines, in: *Proc of IEEE Bioinformatics Conference*, (2003), 504–505.

[9] Y.-M. Chiang, H.-M. Chiang and S.-Y. Lin, The application of ant colony optimization for gene selection in microarray-based cancer classification, in: *Proc of International Conference on Machine Learning and Cybernetics* **7** (2008), 4001–4006.

[10] S.-B. Cho and H.-H. Won, Machine learning in DNA microarray analysis for cancer classification, in: *Proc of the First Asia-Pacific Bioinformatics Conference on Bioinformatics*, APBC '03, Australian Computer Society, Inc. (2003), 189–198.

[11] H.-L. Chou, C.-T. Yao, S.-L. Su, C.-Y. Lee, K.-Y. Hu, H.-J. Terng, Y.-W. Shih, Y.-T. Chang, Y.-F. Lu, C.-W. Chang et al., Gene expression profiling of breast cancer survivability by pooled cdna microarray analysis using logistic regression, artificial neural networks and decision trees, *BMC Bioinformatics* **14**(1) (2013), 100.

[12] F. Chu and L. Wang, Applications of support vector machines to cancer classification with microarray data, *Int Journal of Neural Systems* **15**(6) (2005), 475–484.

[13] J. De Vos, T. Thykjaer, K. Tarte, M. Ensslen, P. Raynaud, G. Requirand, F. Pellet, V. Pantesco, T. Reme, M. Jourdan et al., Comparison of gene expression profiling between malignant and normal plasma cells with oligonucleotide arrays, *Oncogene* **21**(44) (2002), 6848–6857.

[14] J.M. Diaz, R.C. Pinon and G. Solano, Lung cancer classification using genetic algorithm to optimize prediction models, in: *The 5th International Conference on Information, Intelligence, Systems and Applications*, (2014), 1–6.
[15] C. Ding and H. Peng, Minimum redundancy feature selection from microarray gene expression data, *Journal of Bioinformatics and Computational Biology* **3**(2) (2005), 185–205.
[16] M. Dolled-Filhart, L. Rydén, M. Cregger, K. Jirström, M. Harigopal, R.L. Camp and D.L. Rimm, Classification of breast cancer using genetic algorithms and tissue microarrays, *Clinical Cancer Research* **12**(21) (2006), 6459–6468.
[17] A. El Akadi, A. Amine, A. El Ouardighi and D. Aboutajdine, A two-stage gene selection scheme utilizing MRMR filter and GA wrapper, *Knowledge and Information Systems* **26**(3) (2011), 487–500.
[18] N. Friedman, D. Geiger and M. Goldszmidt, Bayesian network classifiers, *Machine Learning* **29**(2–3) (1997), 131–163.
[19] T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer and D. Haussler, Support vector machine classification and validation of cancer tissue samples using microarray expression data, *Bioinformatics* **16**(10) (2000), 906–914.
[20] B.A. Garro, R.A. Vazquez and K. Rodríguez, Classification of DNA microarrays using artificial bee colony (ABC) algorithm, in: *Advances in Swarm Intelligence*, Springer (2014), 207–214.
[21] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science* **286**(5439) (1999), 531–537.
[22] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, Gene selection for cancer classification using support vector machines, *Machine Learning* **46**(1–3) (2002), 389–422.
[23] J. Hair, W. Black, B. Babin and R. Anderson, *Multivariate Data Analysis*, 7th edition, Prentice Hall, USA, 2010.
[24] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten, The WEKA data mining software: An update, *ACM SIGKDD Explorations Newsletter* **11**(1) (2009), 10–18.
[25] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, Extreme learning machine: Theory and applications, *Neurocomputing* **70**(1–3) (2006), 489–501.
[26] J. Huang, J. Lu and C.X. Ling, Comparing naive bayes, decision trees, and SVM with AUC and accuracy, in: *Data Mining, 2003 ICDM 2003 Third IEEE International Conference on*, IEEE (2003), 553–556.
[27] E.B. Huerta, B. Duval and J.-K. Hao, A hybrid GA/SVM approach for gene selection and classification of microarray data, in: *Applications of Evolutionary Computing*, Springer (2006), 34–44.
[28] H.T. Huynh, J.-J. Kim and Y. Won, DNA microarray classification with compact single hidden-layer feedforward neural networks, in: *Frontiers in the Convergence of Bioscience and Information Technologies*, (2007), 193–198.
[29] N.C. Institute, SEER Data, 1973–2010, http://http://seer.cancer.gov/data/, accessed: 2014-03-26.
[30] I.B. Jeffery, D.G. Higgins and A.C. Culhane, Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data, *BMC Bioinformatics* **7**(1) (2006), 359.
[31] I. Jolliffe, *Principal Component Analysis*, Wiley Online Library, 2002.
[32] J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson et al., Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, *Nature Medicine* **7**(6) (2001), 673–679.
[33] R. Kohavi and G.H. John, Wrappers for feature subset selection, *Artificial Intelligence* **97**(1) (1997), 273–324.
[34] D. Koller and M. Sahami, Toward optimal feature selection, Technical report, Stanford InfoLab, Stanford University, 1996.
[35] J.W. Lee, J.B. Lee, M. Park and S.H. Song, An extensive comparison of recent classification tools applied to microarray data, *Computational Statistics & Data Analysis* **48**(4) (2005), 869–885.
[36] W. Li and Y. Yang, How many genes are needed for a discriminant microarray data analysis, in: *Methods of Microarray Data Analysis*, Springer, (2002), 137–149.
[37] J. Liao and K.-V. Chin, Logistic regression for disease classification using microarray data: model selection in a large p and small n case, *Bioinformatics* **23**(15) (2007), 1945–1951.
[38] R. Linder, T. Richards and M. Wagner, Microarray data classified by artificial neural networks, in: *Microarrays*, Springer (2007), 345–372.
[39] B. Liu, Q. Cui, T. Jiang and S. Ma, A combinational feature selection and ensemble neural network method for classification of gene expression data, *BMC Bioinformatics* **5**(1) (2004), 136.
[40] H. Liu and R. Setiono, A probabilistic approach to feature selection-a filter solution, in: *ICML*, Citeseer **96** (1996), 319–327.
[41] X. Liu, A. Krishnan and A. Mondry, An entropy-based gene selection method for cancer classification using microarray data, *BMC Bioinformatics* **6**(1) (2005), 76.
[42] S. Ma and J. Huang, Regularized ROC method for disease classification and biomarker selection with microarray data, *Bioinformatics* **21**(24) (2005), 4356–4362.
[43] A.M. Mahmoud, B.A. Maher, E.-S.M. El-Horbaty and A.B.M. Salem, Analysis of machine learning techniques for gene selection and classification of microarray data, in: *Proc ICIT 2013 The 6th International Conference on Information Technology*, (2013).

[44] S. Mukherjee, Classifying microarray data using support vector machines, in: *A Practical Approach to Microarray Data Analysis*, D.P. Berrar, W. Dubitzky and M. Granzow, eds, Springer US, 2003, pp. 166–185.
[45] S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, T. Golub, J. Mesirov and T. Poggio, Support vector machine classification of microarray data, Technical report, Massachusetts Institute of Technology, 1999.
[46] D.V. Nguyen and D.M. Rocke, Tumor classification by partial least squares using microarray gene expression data, *Bioinformatics* **18**(1) (2002), 39–50.
[47] W.S. Noble et al., Support vector machine applications in computational biology, *Kernel Methods in Computational Biology* (2004), 71–92.
[48] R. Ocampo, M.A. de Luna, R. Vega, G. Sanchez-Ante, L.E. Falcon-Morales and H. Sossa, Pattern analysis in DNA microarray data through PCA-based gene selection, in: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, E. Bayro-Corrochano and E. Hancock, eds, volume 8827 of *Lecture Notes in Computer Science*, Springer International Publishing, 2014, pp. 532–539.
[49] M. Pirooznia, J. Yang, M.Q. Yang and Y. Deng, A comparative study of different machine learning methods on microarray gene expression data, *BMC Genomics* **9**(1) (2008), S13.
[50] T. Revathi and P. Sumathi, A novel microarray gene ranking and classification using extreme learning machine algorithm, *Journal of Theoretical and Applied Information Technology* **68**(3) (2014).
[51] R. Ruiz, J.C. Riquelme and J.S. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, *Pattern Recognition* **39**(12) (2006), 2383–2392.
[52] J. Ryu and S.-B. Cho, Towards optimal feature and classifier for gene expression classification of cancer, in: *Advances in Soft Computing, AFSS 2002*, Springer (2002), 310–317.
[53] M. Shah, M. Marchand and J. Corbeil, Feature selection with conjunctions of decision stumps and learning from microarray data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(1) (2012), 174–186.
[54] L. Shen and E. Chong-Tan, Reducing multiclass cancer classification to binary by output coding and SVM, *Computational Biology and Chemistry* **30**(1) (2006), 63–71.
[55] H. Sossa and E. Guevara, Efficient training for dendrite morphological neural networks, *Neurocomputing* **131** (2014), 132–142.
[56] A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin and S. Levy, A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis, *Bioinformatics* **21**(5) (2005), 631–643.
[57] J.G. Thomas, J.M. Olson, S.J. Tapscott and L.P. Zhao, An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles, *Genome Research* **11**(7) (2001), 1227–1236.
[58] D.L. Tong and A.C. Schierz, Hybrid genetic algorithm-neural network: Feature extraction for unpreprocessed microarray data, *Artificial Intelligence in Medicine* **53**(1) (2011), 47–56.
[59] I. Tsamardinos and C.F. Aliferis, Towards principled feature selection: Relevancy, filters and wrappers, in: *Proc of the Ninth International Workshop on Artificial Intelligence and Statistics*, (2003).
[60] Y. Wang, I.V. Tetko, M.A. Hall, E. Frank, A. Facius, K.F. Mayer and H.W. Mewes, Gene selection from microarray data for cancer classification-a machine learning approach, *Computational Biology and Chemistry* **29**(1) (2005), 37–46.
[61] E.P. Xing, M.I. Jordan, R.M. Karp et al., Feature selection for high-dimensional genomic microarray data, in: *ICML*, Citeseer **1** (2001), 601–608.
[62] H. Yu, G. Gu, H. Liu, J. Shen and J. Zhao, A modified ant colony optimization algorithm for tumor marker gene selection, *Genomics, Proteomics & Bioinformatics* **7**(4) (2009), 200–208.
[63] H. Yu, S. Hong, X. Yang, J. Ni, Y. Dan and B. Qin, Recognition of multiple imbalanced cancer types based on DNA microarray data using ensemble classifiers, *BioMed Research International*, (2013).
[64] H. Zhang, A.L. Cohen, S. Krishnakumar, I.L. Wapnir, S. Veeriah, G. Deng, M.A. Coram, C.M. Piskun, T.A. Longacre, M. Herrler et al., Patient-derived xenografts of triple-negative breast cancer reproduce molecular features of patient tumors and respond to mTOR inhibition, *Breast Cancer Res* **16**(2) (2014), R36.
[65] R. Zhang, G.-B. Huang, N. Sundararajan and P. Saratchandran, Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis, *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* **4**(3) (2007), 485–495.
[66] X. Zhou, K.-Y. Liu and S.T. Wong, Cancer classification and prediction using logistic regression with Bayesian gene selection, *Journal of Biomedical Informatics* **37**(4) (2004), 249–259.