Applied Soft Computing xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Classification of DNA microarrays using artificial neural networks and ABC algorithm

Q1 Beatriz A. Garro [a,*], Katya Rodríguez [a], Roberto A. Vázquez [b]

[a] Instituto en Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Ciudad Universitaria, México, D.F., Mexico
[b] Intelligent Systems Group, Facultad de Ingeniería – Universidad La Salle, Benjamin Franklin 47, Col. Condesa, CP 06140 México, D.F., Mexico

ARTICLE INFO

Article history:
Received 27 April 2015
Received in revised form
13 September 2015
Accepted 3 October 2015
Available online xxx

Keywords:
DNA microarrays
Artificial neural networks
Pattern recognition
Cancer classification
Artificial Bee Colony algorithm

ABSTRACT

DNA microarray is an efficient new technology that allows to analyze, at the same time, the expression level of millions of genes. The gene expression level indicates the synthesis of different messenger ribonucleic acid (mRNA) molecule in a cell. Using this gene expression level, it is possible to diagnose diseases, identify tumors, select the best treatment to resist illness, detect mutations among other processes. In order to achieve that purpose, several computational techniques such as pattern classification approaches can be applied. The classification problem consists in identifying different classes or groups associated with a particular disease (e.g., various types of cancer, in terms of the gene expression level). However, the enormous quantity of genes and the few samples available, make difficult the processes of learning and recognition of any classification technique. Artificial neural networks (ANN) are computational models in artificial intelligence used for classifying, predicting and approximating functions. Among the most popular ones, we could mention the multilayer perceptron (MLP), the radial basis function neural network (RBF) and support vector machine (SVM). The aim of this research is to propose a methodology for classifying DNA microarray. The proposed method performs a feature selection process based on a swarm intelligence algorithm to find a subset of genes that best describe a disease. After that, different ANN are trained using the subset of genes. Finally, four different datasets were used to validate the accuracy of the proposal and test the relevance of genes to correctly classify the samples of the disease.

© 2015 Published by Elsevier B.V.

1. Introduction

DNA microarray is an essential technique in molecular biology that allows, at the same time, to know the expression level of millions of genes. The DNA microarray consists in immobilizing a known deoxyribonucleic acid (DNA) molecule layout in a glass container and then this information with other genetic information are hybridized. This process is the base to identify, classify or predict diseases such as different kind of cancer [1–4].

The process to obtain a DNA microarray is based on the combination of a healthy DNA reference with a testing DNA. Using fluorophores and a laser it is possible to generate a color spot matrix and obtain quantitative values that represent the expression level of each gene [5]. This expression level is like a signature useful to diagnose different diseases. Furthermore, it can be used to identify genes that modify their genetic expression when a medical treatment is applied, identify tumors and genes that make regulation genetic networks, detect mutations among other applications [6].

Computational techniques combined with DNA microarrays can generate efficient results. The classification of DNA microarrays can be divided into three stages: gene finding, class discovery, and class prediction [7,8]. The DNA microarray samples have millions of genes and selecting the best genes set in such a way that get a trustworthy classification is a difficult task. Nonetheless, the evolutionary and bio-inspired algorithms, such as genetic algorithm (GA) [9], particle swarm optimization (PSO) [10], bacterial foraging algorithm (BFA) [11] and fish school search (FSS) [12], are excellent options to solve this problem. However, the performance of these algorithms depends of the fitness function, the parameters of the algorithm, the search space complexity, convergence, etc. In general, the performance of these algorithms is very similar among them, but depends of adjusting carefully their parameters. Based on that, the criterion that we used to select the algorithm for finding the set of most relevant genes was in term of the number of

Q2 * Corresponding author. Tel.: +52 5556223899.
E-mail addresses: beatriz.garro@iimas.unam.mx (B.A. Garro), katya.rodriguez@iimas.unam.mx (K. Rodríguez), ravem@lasallistas.org.mx (R.A. Vázquez).

UNCORRECTED PROOF

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

parameters of each algorithm. In that sense, the ABC algorithm was chosen because it has fewer parameters to adjust compared with other evolutionary algorithms. Moreover, literature reports that the ABC algorithm presents faster convergence than other techniques. According to [13], results to solve multi-modal and multi-variate problems are better or similar to other evolutionary algorithms. Additionally, ABC presents a higher population diversity avoiding premature convergence. However, other bio-inspired techniques, such as differential evolution, particle swarm optimization, etc., will be evaluated and compared in future works.

Artificial neural networks (ANN) are excellent computational models that have been implemented to solve different kind of problems. The pattern classification, forecasting and regression problems are areas where the ANN have demonstrated to be an efficient technique [14]. ANN have been widely applied in DNA microarrays. For example, in [15], the authors used a multilayer perceptron (MLP) with back-propagation learning and a dimensional reduction method based on k-means and principal component analysis (PCA) techniques. In [16], the authors described an application based on ANN aimed to cancer studies. In [17], the authors diagnosed disease categories using small round blue cell tumors (SRBCT) by means of reducing the dimensionality data using PCA and training ANN models with no hidden layers. In other works, like [18], the authors selected a set of genes using a filter and k-means technique to train a support vector machine (SVM) and a multilayer perceptron (MLP). In [19], the author used mutual information techniques for selecting the most relevant genes before performing the classification task. In [20], an ANN with a sample filtering algorithm is designed for separating the wrongly labeled samples from the training set, and used to construct one more ANN just for the wrong samples classified. In [21], the authors described the singular value decomposition (SVD) technique for training a single layer feed-forward neural network. The authors in [22] performed a selection of genes based on k-means and PCA; finally, an ANN was training during a recursive feature elimination to classify BRCA1 and BRCA2 mutations and childhood SRBCT. In [23], the authors performed a feature selection in DNA microarrays using an ensambling learning technique. Also, they used an algorithm that converts a multiclass problem into multiple binary classes to reduce the complexity of the problem. In [24], the authors analyzed a generalized radial basis function (GRBF), where the coefficients of the neural network were tuned by a hybrid evolutionary algorithm. In [25], the authors used a neurofuzzy model (NFM) for identify distinct prognostic genes with a carcinogenic pathways.

On the other hand, bioinspired and evolutionary algorithms have been widely applied to select the set of genes that best describe a disease. For example, in [26], the authors used the ant colony optimization algorithm (ACO) for selecting the most representative genes from a DNA microarray. A nonparallel plane proximal classifier (NPPC) is described in [27], where the authors used genetic algorithms for selecting genes for a cancer diagnosis and the results are compared against a support vector machine (SVM). In [28], the authors described a genetic bee colony algorithm in order to select the most predictive and informative genes for cancer classification. In [29], the authors presented an improved genetic algorithm that selects the gene subset from the high dimensional gene data for breast cancer diagnosis. In [30], the authors proposed a novel feature selection approach for the classification of high dimensional cancer microarray data, which uses filtering technique such as signal-to-noise ratio (SNR) score and optimization techniques as particle swarm optimization (PSO).

In this research, we introduce a new approach for classifying DNA microarray data based on artificial neural networks and dimensional reduction technique, previously described in [31]. The proposed methodology uses the Artificial Bee Colony (ABC) algorithm as an optimization technique for selecting the set of genes, from a DNA microarray, that best described a particular disease. After that, this information is used to train three types of ANN (multilayer perceptron (MLP), radial basis function (RBF) and support vector machines (SVM)) for classifying the DNA microarrays associated to a disease. In order to test the accuracy of the proposed methodology, four different datasets were used.

It is important to remark that other strategies, applied to DNA microarrays classification, implement the ANN in the fitness function and at the same time perform a dimensional reduction, provoking that the individual evaluation be more expensive in time and computational resources. The main contribution of this paper is firstly reduced the number of genes by means of the ABC algorithm. The proposed fitness function was computed in terms of the classification error using an Euclidean distance. Then, the reduced genes set is used to train an ANN in order to classify the DNA microarray data.

The rest of this paper is organized as follows: Section 2 presents an introduction to DNA microarrays. A brief explanation of ABC algorithm is presented in Section 3. Section 4 presents the basic concepts related to artificial neural networks. In addition, the propose methodology is outlined in Section 5 followed by the experimental results in Section 6. Finally, conclusions of this research are given in Section 7.

2. DNA microarrays

The human genome sequencing was completed in 2001 [32,33]. This discovery impacted the world because has allowed better diagnostics, to know the genes that participate in an illness for doing a better treatment and even more, to know about the human evolution and other advantages in sciences like biomedics, genetics, biology and so forth. In [34], the authors described the use of DNA microarray technologies, presented an overview of their frequent biomedical applications and described the steps of a typical laboratory procedure to obtain information with this powerful technique.

DNA microarray is a container that immobilize DNA molecule, complementary DNA or oligonucleotides for hybridizing with DNA molecule marked to be analyzed. The container is made of glass, nylon or silicone. There are two types of DNA microarrays: the ergonomic and the transcriptomics. The first one is divided into two kinds: that can detect lost or profit genes, and that can detect mutations. The second one measures the mRNA levels [35]. DNA microarray allows to use the genome sequencing information to measure quantitatively the expression level of millions of genes at the same time. This expression level is like a signature useful to diagnose diseases, identify tumors, select the best treatment to resist illness and detect mutations.

To obtain the expression level of a DNA microarray sample is necessary to compare the healthy DNA reference, called “data control”, against a testing DNA (the sample to be studied), see Fig. 1 [34]. First, the messenger ribonucleic acid (mRNA) of both tissues is isolated. Then, it is necessary to obtain the corresponding complementary DNA (cDNA). Additionally, these molecules should be marked with a different fluorophore: Cy3 for the experimental sample (red color) and Cy5 for the control sample (green color). Furthermore, the marked molecules are mixed for the hybridizing process that consists of the union of the cDNA of each sample [36]. The result is a matrix with many colored spots. The red color indicates that a particular gene (spot) is more expressed in the diseased sample. The green color means that a particular gene is more expressed in the healthy sample. The yellows spots indicate that the gene is equally expressed in healthy and diseased samples.

DNA microarray is an efficient technology that presents many advantages. It allows the analysis of thousand of genes at the same time, decreasing the spend time to its study. Also it increases

UNCORRECTED PROOF

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

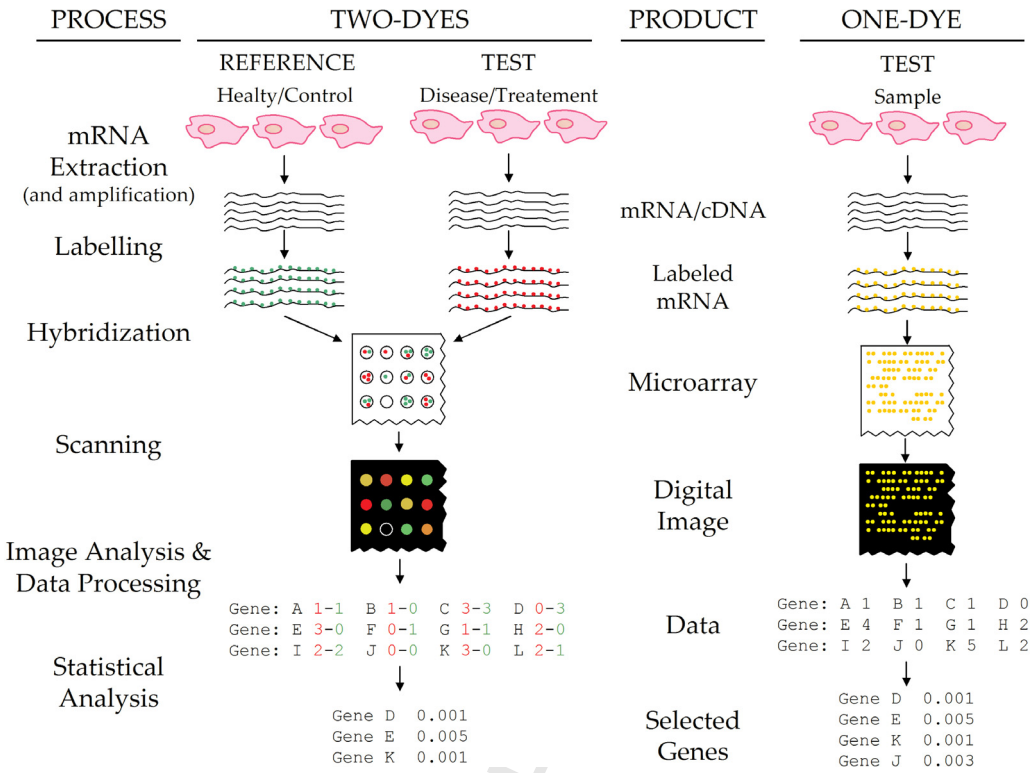

Fig. 1. Schematic representation of a gene expression in DNA microarray [34].

the speed to detect illness, selecting the best treatment for each patient. On the other hand, it also allows to know the iterations between genes under certain conditions that can derivative in amazing discoveries in biology, medicine, informatics and so on. The disadvantage with this kind of data is the enormous quantity of information to be analyzed, patterns with millions of genes. Furthermore, only few samples of different disease are available. In this sense, the application of computational techniques could give valuable information about various tasks related to DNA microarrays, mainly in the selection of the set of genes that best describes a disease, classification of DNA microarrays, etc.

3. Artificial Bee Colony (ABC) algorithm

An excellent optimization technique is Artificial Bee Colony (ABC) algorithm based on the metaphor of the bees foraging behavior [37]. It is composed of a population of NB bees $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, \ldots, NB$ represented by the position of the food sources (possible solutions). Three classes of bees are used to achieve the convergence near to the optimal solution i.e. the best set of genes.

Employed bees: They search for new neighbor food source near of their hive. After that, they compare the food source against the old one using Eq. (1). Then, a greedy selection is done.

$$v_i^j = x_i^j + \phi_i^j(x_i^j - x_k^j) \tag{1}$$

in which $k \in \{1, 2, \ldots, NB\}$ and $j \in \{1, 2, \ldots, n\}$ are randomly chosen indexes and $k \neq i$. ϕ_i^j is a random number between $[-a, a]$.

After that, the bee evaluates the quality of each solution based on the fitness function.

Onlooker bees: The onlooker bee probabilistically chooses a food source depending on the amount of nectar shown by each employed bee, see Eq. (2).

$$p_i = \frac{fit_i}{\sum_{k=1}^{NB} fit_k} \tag{2}$$

in which fit_i is the fitness value of the solution i and NB is the number of food sources that are equal to the number of employed bees.

Scout bees: This kind of bees randomly create new solutions when a food source or solution cannot be improved anymore during a period called "limit" or "abandonment criteria", see Eq. (3).

$$x_i^j = x_{\min}^j + rand(0, 1)(x_{\max}^j - x_{\min}^j) \tag{3}$$

The pseudo-code of the ABC algorithm is shown in Algorithm 1.

Algorithm 1. Pseudo-code ABC algorithm.

```
Initialize the population of solutions x_i ∀_i, i = 1, ..., NB.
Evaluate the population x_i ∀_i, i = 1, ..., NB.
for cycle = 1 to maximum cycle number MCN do
Produce and evaluate new solutions v_i from the
employed bees by using v_i^j = x_i^j + φ_i^j(x_i^j − x_k^j).
Apply the greedy selection process.
Calculate the probability values p_i for the solutions x_i
by using p_i = fit_i / Σ_{k=1}^{NB} fit_k.
Produce and evaluate the new solutions v_i for the
onlookers from the solutions x_i selected depending on
p_i.
Apply the greedy selection process.
Replace the abandoned solutions with a new one
randomly produced x_i by scout bees using
x_i^j = x_min^j + rand(0, 1)(x_max^j − x_min^j).
Memorize the best solution achieved so far.
cycle = cycle + 1
end for
```

4. Artificial neural networks

An artificial neural network (ANN) is a mathematical and computational model that simulates the communication among neurons in the human brain for classifying, forecasting, regression, optimization among some applications. This system performs a mapping between an input and output pattern that represents a real problem [38]. It is composed of a set of neurons (represented by functions) connected to others organized in different layers where

UNCORRECTED

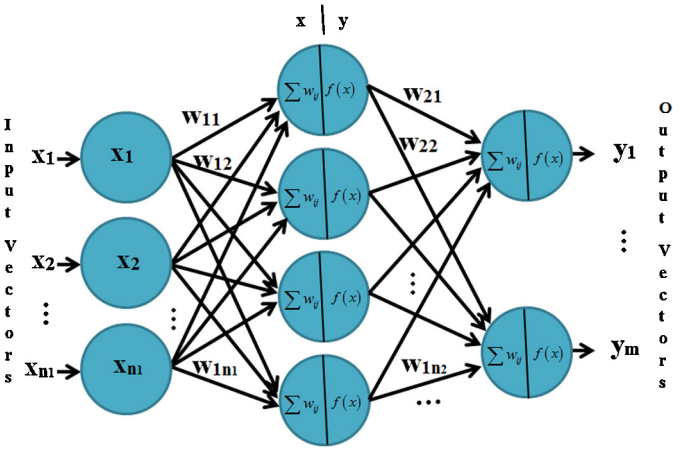

Fig. 2. Schematic representation of an artificial neural network.

each layer is composed of N_L neurons. The problem to be solved is represented by input patterns $\mathbf{x} \in \mathbb{R}^{N_L}$ which are sent through the layers. The information is mapped by means of the corresponding synaptic weights $\mathbf{w} \in \mathbb{R}^{N_L}$. Notice that N_L also represents the number of synaptic weights that arrives to the neuron in each layer. The neurons in the following layers perform an integration of this information depending on whether there exists a connection between them. In addition, another input called *bias* is considered. This bias is a threshold that represents the minimum level that a neuron needs for activating and is represented by θ. A typical integration function is presented in Eq. (4).

$$o(\mathbf{w}, \mathbf{x}, \theta) = \sum_{j=1}^{N_L} x_j w_j + \theta \tag{4}$$

Then, the result of the summation is evaluated in a transfer functions $F(o)$ activated by the input neuron. The result is the output neuron, and this information is sent to the other connected neurons until they reach the last layer, see Fig. 2.

For the case of a multilayer perceptron (MLP), the output of the NN is obtained by Eq (5).

$$y_i = F_i(o_i(\mathbf{w}_i, \mathbf{x}, \theta)) \tag{5}$$

in which the transfer functions F could be represented by sigmoid, Gaussian, piecewise linear, sine, hyperbolic tangent, etc.

For the case of the radial basis function (RBF), the output of the ANN is obtained by Eq. (6).

$$y_i = o_i(\mathbf{w}_i, \phi(\mathbf{x}), \theta)) \tag{6}$$

in which $\phi(\mathbf{x})$ is defined as Eq. (7):

$$\phi(x) = \exp\left(-\frac{\|x-\mu\|}{2\sigma^2}\right) \tag{7}$$

in which μ determines the center of basis function ϕ and σ the standard deviation.

For the case of support vector machine (SVM), the typical integration function is presented in Eq. (8).

$$o(\mathbf{w}, \mathbf{x}, \theta) = \sum_{q=1}^{m} K\left(\mathbf{x}, \mathbf{x}_q\right) w_q + \theta \tag{8}$$

in which $\mathbf{x}_q$ is a support vector and m the number of support vectors. Finally, the output of the SVM is obtained by Eq (9).

$$y_i = F_i(o_i(\mathbf{w}_i, K(\mathbf{x}, \mathbf{x}_q), \theta)) \tag{9}$$

in which F_i is defined as the *sing* function and $K(\mathbf{x}, \mathbf{x}_q)$ is the transformation kernel such as linear (Eq. (10)), quadratic or polynomial (Eq. (11)).

$$K(\mathbf{x}_p, \mathbf{x}_q) = \mathbf{x}_p{}^T \cdot \mathbf{x}_q \tag{10}$$

$$K(\mathbf{x}_p, \mathbf{x}_q) = (\mathbf{x}_p \cdot \mathbf{x}_q + 1)^d \tag{11}$$

in which d is the polynomial degree.

The synaptic weight adaptation consists in changing its value until it reaches the desired behavior. The output is evaluated to measure the efficiency of the ANN in terms of the mean square error (MSE) given by Eq. (12). If the output is not the desired, the weights set has to be changed or adjusted in terms of the input patterns $\mathbf{x} \in \mathbb{R}^N$ (supervised learning).

$$e = \frac{1}{p \cdot M} \sum_{\xi=1}^{p} \sum_{i=1}^{M} (d_i{}^{\xi} - y_i{}^{\xi})^2 \tag{12}$$

Given the training sample $\mathbf{T}^{\xi}$, as defined in Eq. (13), the requirement is to compute the neural network free parameters so that the actual output $\mathbf{y}^{\xi}$ of the neural network due to $\mathbf{x}^{\xi}$ is close enough to $\mathbf{d}^{\xi}$ for all ξ in a statistical sense [33].

$$\mathbf{T}^{\xi} = \{(\mathbf{x}^{\xi} \in \mathbb{R}^N, \mathbf{d}^{\xi} \in \mathbb{R}^M)\} \quad \forall \xi = 1, \ldots, p \tag{13}$$

in which $\mathbf{x}$ is the input pattern, $\mathbf{d}$ the desired response, M is the size of desired pattern and p is the number of patterns.

For the learning task, it is essential to divide the input data into two parts: training and generalization sets. Next, these sets are used in the two stages: training (learning) and testing (generalization). Training consists of adjusting the synaptic weights using the training and validation data. Moreover, the validation set helps to avoid the over-fitting problem in ANN. When the best synaptic weight values of an ANN are found, and the learning phase achieves an acceptable accuracy, the generalization phase is carried out using the testing set. The ANN processes this set and the generalization error is computed. This stage is the most important because the results reflect the ANN capacity to solve a problem.

There are several algorithms that adjust the synaptic weights (learning task) to obtain the minimum error. One of the most popular technique is the classical Backpropagation (BP) algorithm [39,40], widely applied for training multilayer perceptrons (MLP) and radial basis function (RBF). This algorithm, like others, is based on the descendant gradient technique.

5. Proposed methodology

In this section, we introduce a new methodology for solving DNA microarray classification problems. The proposed methodology was divided into two main stages, see Fig. 3. The first one is devoted to the selection of the most relevant features, the choice of the genes that best describes a disease. Due to the number of samples (or patterns), much lower than the number of genes (characteristics), it is necessary to perform a dimensionality reduction. For the dimensionality reduction of each DNA microarray, instead of using the conventional method as PCA, we decided to use an artificial bee colony algorithm as described in [41].

It is important to remember that one of the major problems with DNA microarrays is that the number of samples from the dataset is much lower than the number of genes (or features of each sample) and this balance plays a relevant role to classify successfully DNA microarrays. After selecting the set of genes, the number of samples is greater than the number of genes.

Once the best set of genes is obtained, during the second stage, we trained an artificial neural network, adjusting its synaptic

UNCORRECTED PROOF

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

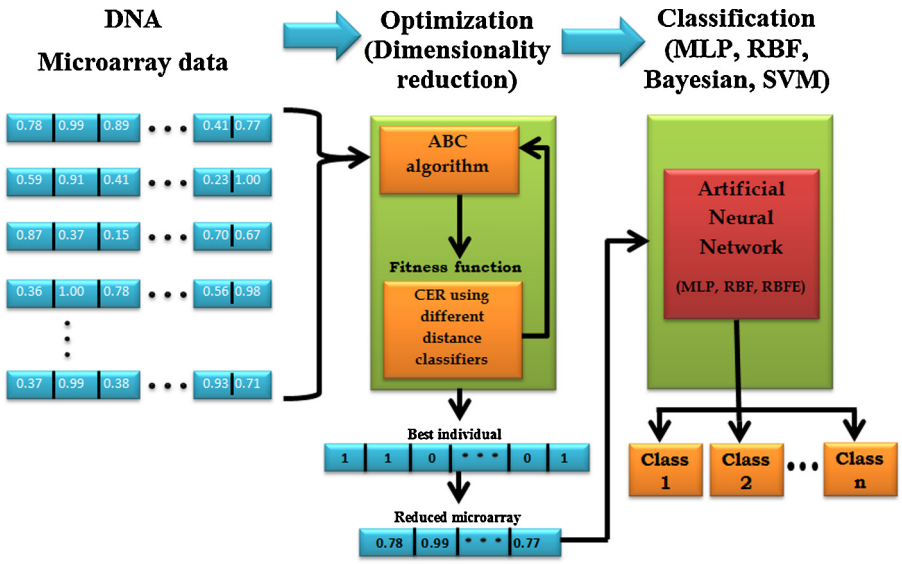

Fig. 3. Schematic representation of the proposed methodology.

weights until the learning process achieves the best classification rate, using the set of genes selected in the first phase.

5.1. Stage of dimensionality reduction

The problem to be solved can be defined as follows: Giving a set of p input patterns $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$, $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1 \ldots, p$ and its corresponding set of desired classes associated to each pattern $\mathbf{d} = \{d_1, \ldots, d_p\}$, where $d_i \in \{1, \ldots, K\}$ and K is the number of classes, find a subset of genes $G \in \{0, 1\}^n$ such that a function defined by $\min(F(\mathbf{X}|_G, \mathbf{d}))$ is minimized.

The problem solution is represented by a subset of genes and is defined by an array $I \in \mathbb{R}^n$. Each individual I_q, $q = 1, \ldots, NB$ is binarized using Eq. (14) with a threshold level th. This threshold select the best set of genes defined as $G^k = T_{th}(I^k)$, $k = 1, \ldots, n$; when the component is set to 1, indicates that this gene will be selected to make up the subset of genes. In other words, th determines the probability that a gene can be selected.

$$T_{th}(x) = \begin{Bmatrix} 0, & x < th \\ 1, & x \geq th \end{Bmatrix} \tag{14}$$

To evaluate the solutions found by the ABC algorithm and determine which is the best solution, it is necessary to define a fitness function. This function is described in the next section.

5.1.1. Classification error function CER

The aptitude of an individual is represented by the classification error function (CER), defined in Eq. (15). This fitness function measures how many samples have been wrongly predicted.

$$F(\mathbf{X}|_G, \mathbf{d}) = \frac{\sum_{i=1}^{p} \left(\left| \arg\min_{k=1}^{K} (D(\mathbf{x}_i|_G, \mathbf{c}^k|_G)) - d_i \right| \right)}{tng} \tag{15}$$

in which tng is the total number of gene expressions to be classified, D is a distance measure, K is the number of classes and $\mathbf{c}$ is the center of each category. Different distance measures could be applied to classify the gene expression samples, for example, classic Euclidean distance given by Eq. (16).

$$D(\mathbf{r}, \mathbf{s}) = \sqrt[2]{\sum_{j=1}^{n} |r^j - s^j|^2} \tag{16}$$

in which $\mathbf{r} \in \mathbb{R}^n$ and $\mathbf{s} \in \mathbb{R}^n$.

5.2. Stage of pattern classification

Once selected the set of features that best describes the disease, the next step is to train an artificial neural network. In this stage, any artificial neural network can be used. However, we decided to use a multilayer perceptron (MLP), a radial basis function neural network (RBF) and a support vector machine (SVM).

These three types of ANN were chosen because they are the most applied to solve different problems. In addition, there are some differences among them, which could be interesting to know. Some differences are that the RBF has a training stage faster than MLP using only one layer with a neurons number increasing while a MLP works with a more that one layer with multiple neurons. In addition, the learning process that is crucial in the performance of an ANN is different between a MLP and RBF because one links the inputs and outputs with hyperplanes by means of a distributed learning, and the other uses hyperspheres by means of a local learning [42]. On the other hand, SVM belongs to a family of generalized linear classifiers and can be interpreted as an extension of the perceptron where simultaneously minimize the classification error and maximize the geometric margin. For these reasons, we decided to include in this paper the results using these types of ANN.

The inputs of these artificial neural networks are feed with the genes previously selected using the ABC algorithm. Before starting to train the ANN, the dataset with the best features was partitioned into two datasets: training and testing subsets. After that, the ANN is trained with a defined number of epochs until a goal error is reached.

Once trained the ANN, we proceed to evaluate its generalization capabilities using the testing subset. To measure the accuracy of the ANN, we computed the classification performance by means of Eq. (17).

$$CP = \frac{npbc}{tpc} \tag{17}$$

in which $npbc$ represents the number of patterns well classified and tpc is the total of tested patterns.

UNCORRECTED PROOF

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

6. Experimental results

In this section, the proposed methodology is experimentally analyzed in order to determine its accuracy. Different high-dimensional biomedical DNA microarray dataset benchmarks were used to establish the accuracy of the proposed methodology. The original data division between training and testing phases for each dataset is next described.

Leukemia ALL-AML data set: It consists of 38 bone marrow samples for training (27 ALL and 11 AML), over 7129 probes from 6817 human genes. Also, 34 samples testing data are provided, with 20 ALL and 14 AML [1].

Breast cancer data set: The training dataset contains 78 patient samples, 34 labelled as "relapse", the rest 44 labelled as "non-relapse". The testing dataset contains 12 relapses and seven non-relapse samples. The number of genes is 24,481 [4,43].

DLBCL-NIH data set: The diffuse Large B-Cell Lymphoma (National Institutes of Health) contains 240 patients divided into two groups: the training group contains 160 patients, and the testing group contains 80 patients. The number of genes of each microarray is 7399 [44].

Prostate cancer data set: The training set contains 52 prostate tumor samples and 50 non-tumor (labeled as "normal") prostate samples with around 12,600 genes. The testing set contains 25 tumor samples and nine normal samples [45].

We performed several experiments to validate the accuracy of the propose methodology. We divided these sets of experiments into two sections. The first one is aimed to corroborate if the proposed methodology was capable of finding the best set of genes that best describe DNA microarray associated with the disease. In this stage, we evaluate how the ABC algorithm combined with a fitness function, in terms of the Euclidean distance classifier, selects the best set of genes. Afterward, in the second section, we evaluated how an ANN, trained with the set of genes found by the ABC algorithm, improves their capability of predicting and classifying the diseases samples.

6.1. Dimensionality reduction results

This section shows the experimental results obtained while evaluating the behavior of the ABC algorithm during the task of discovering the most representative genes. The parameters for the ABC algorithm during the training phase were defined as: population size ($NB = 40$), maximum number of cycles $MNC = 2000$, limit $l = 100$ and food sources $NB/2$.

As we described in the methodology section, the parameter th plays an important role to control the probability of selecting the genes that will compose the set of most representative genes. The first group of experiments consists on evaluating the threshold value and knowing how this probability affects the selection of most representative genes set. Five threshold values were applied: 0.1, 0.3, 0.5, 0.7, 0.9. The labels numbered from 1 to 5 shown these values respectively.

Due to ABC algorithm is a stochastic search method, it cannot guaranty the global optimum solution in each run. For that reason, it is necessary to perform several runs to statistically validate the accuracy of the proposed methodology for discovering the most representative genes associated with a particular disease. Once the proposed methodology was validated, we could select the solution that provided the best accuracy or the solution that used the minimum number of genes. At the end, although we cannot guarantee the global optimum solution in each run, we can select the solution that provides the best accuracy in order to train an ANN during the next stage.

In order to statistically validate the experimental results, the proposed methodology was executed 30 runs using the fitness function defined in terms of the euclidean distance (see Eq. (16)) for each configuration threshold. In each execution, the original partitions for training and testing set were used.

Fig. 4 shows the evolution of learning error for the five threshold configurations and the four cancer datasets. In this figure, it is possible to appreciate that, for three cancer problems, the best solution improves when the threshold value decreases. In the case of evolution error for the Leukemia dataset, the results are stable despite changing the threshold value.

Once obtained the set of genes that maximizes the accuracy, the testing dataset was analyzed using an Euclidean distance classifier.

Table 1 shows relevant information about the experimental results obtained with the proposed methodology for the Leukemia ALL-AML dataset. The training classification rate achieved with a threshold ($th = 0.9$) was of 100%. For the case of testing classification rate, the average accuracy was of 79.3%. In average, the minimum genes number was 712. Moreover, the best results were using $th = 0.3$, achieving an efficiency of 100% during the training phase and 88.2% during the testing phase. Furthermore, these results were obtained only with three genes. The best genes found by the methodology, according to the dataset are: SLC17A2 Solute carrier family 17 (L13258_at), MLC gene (M22919_rna2_at) and FBN2 Fibrillin 2 (U03272_at).

Table 2 shows the experimental results obtained with the proposed methodology for the breast cancer dataset. The best average accuracy was achieved with $th = 0.1$, obtaining an efficiency of 85.6% for the training dataset and above the 65.3% for the case of testing dataset. However, the best solution was obtained with the threshold configuration $th = 0.5$. This setting provides a 83.3% with the training dataset and 84.2% for testing dataset. In this set of experiments, we observed that the solution with the less quantity of genes corresponds to the best set of genes found in the proposed methodology. The best five genes found are Contig36809-RC, Contig19623-RC, AL080059, NM-015641 and NM-000698 all according to the database.

Table 3 shows the results for the DLBCL-NIH dataset. With the $th = 0.1$, the methodology obtained the best average accuracy, achieving an efficiency of 71.3% and 58.1% for the training and the testing stages, respectively. However, the best result obtained was 75.6% and 67.5% in the training and testing stages, respectively. The genes founded by the methodology to obtain the best accuracy were 17: 16,549, 16,562, 17,444, 15,910, 34,774, 32,012, 17,163, 25,205, 26,372, 31,467, 27,752, 27,734, 19,389, 34,316, 29,960, 17,521 and 30,471. On the other hand, we observed that the methodology generates a solution with a less quantity of genes, however, the results were not as good as we could desire.

For the case of prostate cancer, we observed that the best average accuracy was obtained with a threshold value of $th = 0.7$. With this configuration, the proposed methodology achieved an efficiency of 81.3% for the training and 67.6% for the testing stage (Table 4). The best accuracy also was achieved by the same threshold, with an increase of the 87.3% and 100% in training and testing phases. The name of the best genes that obtain the best accuracy are 32076-at, 1930-at and 1276-g-at.

6.2. ANN: pattern classification results

Instead of trained an ANN with the enormous quantity of genes of each DNA microarray sample, we used the set of genes obtained during the first stage of the proposed methodology. By using these genes combined with the ANN, we expect to improve the results achieved with the Euclidean distance classifier.

In this subsection, three of the most popular ANN are used: the multilayer perceptron (MLP), the radial basis function neural network (RBF) and the support vector machine (SVM). Two different experiments were done in order to analyze how much these

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

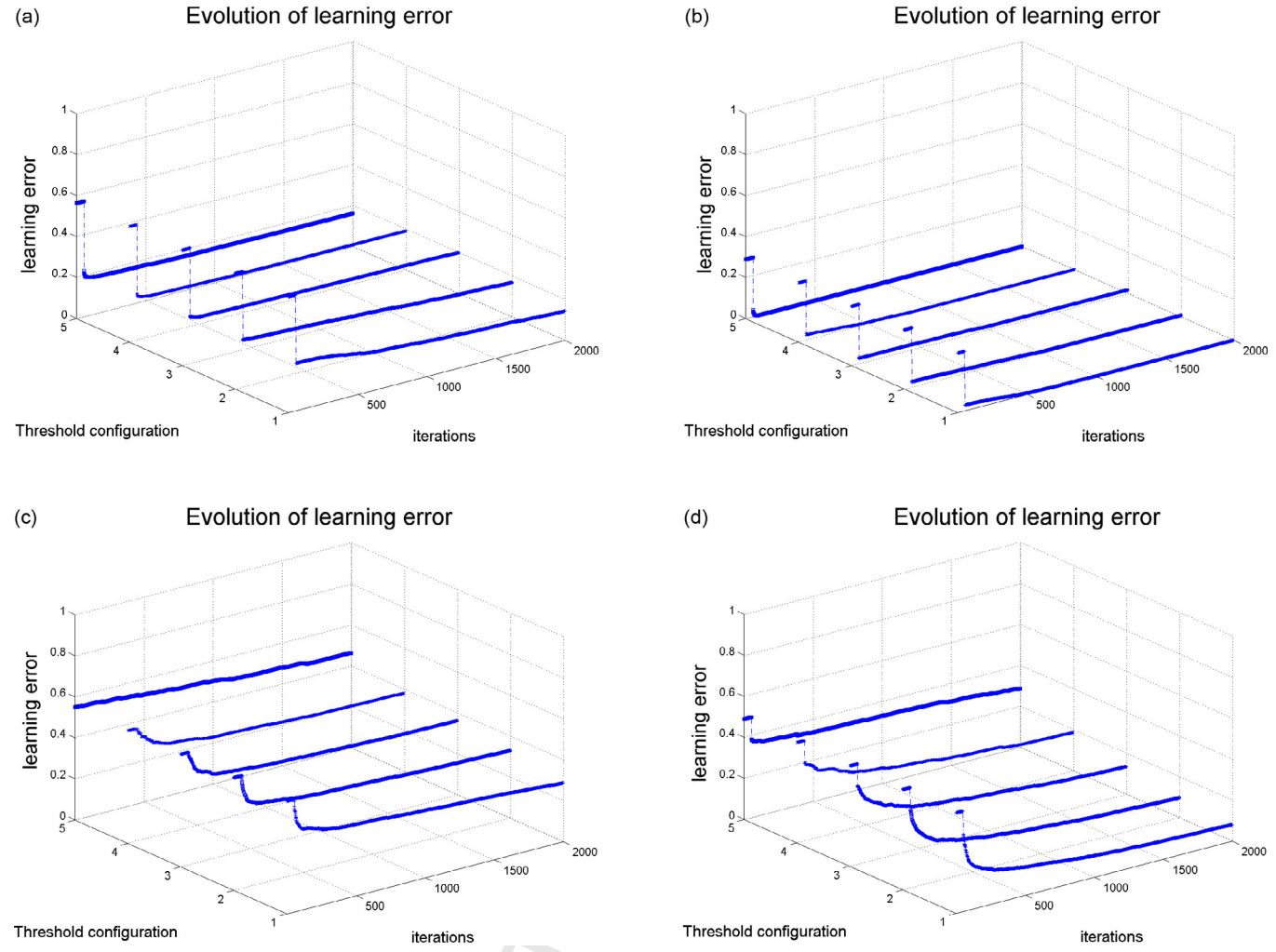

Fig. 4. Evolution of learning error using the fitness function in terms of the Euclidean distance. (a) Learning error evolution for the Breast cancer dataset. (b) Learning error evolution for Leukemia dataset. (c) Learning error evolution for Diffuse Large B-Cell Lymphoma dataset. (d) Learning error evolution for prostate cancer dataset.

Table 1
Behavior of the proposed methodology using a euclidean distance classifier for Leukemia ALL-AML dataset.

th	Average accuracy		Ave. # of genes	Ave. # of iter.	Best accuracy		# of genes	# of iter.
	Tr. cl.	Te. cl.			Tr. cl.	Te. cl.		
0.1	0.998 ± 0.00	0.751 ± 0.07	434.8	430.0	1.000	0.853	5	318
0.3	0.992 ± 0.01	0.746 ± 0.07	1506.7	724.7	1.000	0.882	3	253
0.5	0.982 ± 0.01	0.769 ± 0.03	2618.3	933.5	1.000	0.824	4	446
0.7	0.996 ± 0.00	0.770 ± 0.01	2054.8	452.4	1.000	0.794	3	335
0.9	1.000 ± 0.00	0.793 ± 0.03	712.0	63.0	1.000	0.853	695	62

Tr. cl. = Training classification rate, Te. cl. = Testing classification rate.

Table 2
Behavior of the proposed methodology using a euclidean distance classifier for breast cancer dataset.

th	Average accuracy		Ave. # of genes	Ave. # of iter.	Best accuracy		# of genes	# of iter.
	Tr. cl.	Te. cl.			Tr. cl.	Te. cl.		
0.1	0.856 ± 0.01	0.653 ± 0.07	14.7	2001.0	0.885	0.789	18	2001
0.3	0.831 ± 0.01	0.616 ± 0.08	582.1	2001.0	0.872	0.737	14	2001
0.5	0.800 ± 0.01	0.633 ± 0.05	8153.5	2001.0	0.833	0.842	5	2001
0.7	0.809 ± 0.00	0.632 ± 0.00	7344.3	2001.0	0.821	0.632	7388	2001
0.9	0.838 ± 0.00	0.632 ± 0.00	2438.1	2001.0	0.859	0.632	2481	2001

Tr. cl. = Training classification rate, Te. cl. = Testing classification rate.

UNCORRECTED

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

Table 3
Behavior of the proposed methodology using a euclidean distance classifier for DLBCL-NIH dataset.

th	Average accuracy		Ave. # of genes	Ave. # of iter.	Best accuracy		# of genes	# of iter.
	Tr. cl.	Te. cl.			Tr. cl.	Te. cl.		
0.1	0.713 ± 0.02	0.581 ± 0.06	17.0	2001.0	0.756	0.675	17	2001
0.3	0.669 ± 0.01	0.537 ± 0.07	6.7	2001.0	0.719	0.662	13	2001
0.5	0.638 ± 0.02	0.511 ± 0.09	3.8	2001.0	0.706	0.637	7	2001
0.7	0.619 ± 0.02	0.520 ± 0.07	1.8	2001.0	0.675	0.637	1	2001
0.9	0.538 ± 0.06	0.520 ± 0.10	0.8	2001.0	0.637	0.650	1	2001

Tr. cl. = Training classification rate, Te. cl. = Testing classification rate.

techniques can improve the results obtained with the euclidean distance classifier (EDC). The first group of experiments uses the set of genes that generate the best classification results. The second group of experiments takes the solution with the less quantity of genes. Notice that for some datasets, the solution with the best set of genes and less quantity of genes corresponds to the same genes.

Two MLP, composed of one hidden layer (MLP1) and two hidden layers (MLP2), were designed. Their input layer contains *in* neurons that correspond to the number of genes obtained during the first stage of the proposed methodology.

For the case of the hidden layers, it is well-known that if the number of neurons are less as compared to the complexity of the problem data, then under-fitting may occur. If unnecessary neurons are present in the network, then overfitting may occur. Although selecting the number of neurons in hidden layers could be treated as a heuristic task, there are some rules that allows to determine this number in terms of the input, the output and the type of transfer function.

In that sense, the hidden layer was composed of *hn* sigmoid neurons according to Eq. (18)

$$hn = (n_g + n_c) * 3 \tag{18}$$

in which n_g corresponds to the number of genes and n_c represents the number of classes.

It is important to mention, that in the previous rule, we considered n_c for computing the number of hidden neurons due to we use a winner-take-all technique to determine to which class the input pattern belongs. In order to do that, the neurons of the output layer enter into a competition stage where the neuron with the highest output is set to 1 and the remaining neurons are set to 0. The neuron set with 1 determines the class to which the input pattern belongs.

For the MLP with two hidden layers (MLP2), the number of neurons for first hidden layer and second hidden layer was determined according to Eqs. (19) and (20).

$$hn1 = 0.6hn \tag{19}$$

$$hn2 = 0.4hn \tag{20}$$

The designed MLP was set according to the next parameters: the learning rate was set to 0.1, the number of epoch for the training phase was set to 5000, and the goal error was set to 0. The Levenberg–Marquardt algorithm was used during the training phase of the MLP.

For the case of the RBF, we followed two different approaches. The first method (RBFE) uses an RBF with the next architecture. The first layer has *in* Gaussian neurons, that corresponds to the same number of genes found in the previous stage. The second layer contains *sn* linear neurons corresponding to the number of classes of the problem. The spread for the gaussian neurons was set to 0.75, and the goal error was set to 0.

The second method (RBF) uses an incremental architecture with the same parameters as RBFE. At the beginning, the first layer has no neurons. These neurons are added during the learning process until a specific error is reached. The weights of the added neuron are set to the same values of the input vector that generates the greatest error during the simulation. The spread for the gaussian neurons was set to 0.75, and the goal error was set to 0.0001.

For the case of the SVM we use different kernels such as linear (SVML) quadratic (SVMQ) and polynomial (SVMP).

In order to statistically validate the results obtained with the ANN, 30 executions were performed. In each execution, the dataset was partitioned following two different criteria. Original partition (O) where data is partitioned as in the original dataset. Random partition (R) where 80% of the samples were selected randomly from the original dataset to construct the training dataset and the remaining for the testing dataset.

Table 5 shows the results for Leukemia problem using the ANN as classifier. According to the results presented in Table 1, the number of genes used to train the ANN was three. In this case, the solution with less quantity of genes corresponds to the solution that generates the best accuracy. In Table 5, it is possible to see that the best average performance was obtained using a random partition of the data in combination with a SVM (linear kernel). The best average accuracy provided by the proposed methodology was of 94.0%. Moreover, the best accuracy obtained with the proposed methodology was 100%, using the MLP and SVM.

In Table 6, we analyzed the results for the breast cancer problem. The best average accuracy was obtained using the original partition combined with the SVM (linear kernel) achieving an efficiency of 84.2%. In this case, the solution with the best number of genes corresponds to that with the less quantity of genes, (see Table 2). Nonetheless, the best accuracy was achieved using the MLP over the original and random partitions. The efficiency obtained with this configuration was 94.7%.

Table 4
Behavior of the proposed methodology using a euclidean distance classifier for Prostate tumor dataset.

th	Average accuracy		Ave. # of genes	Ave. # of iter.	Best accuracy		# of genes	# of iter.
	Tr. cl.	Te. cl.			Tr. cl.	Te. cl.		
0.1	0.919 ± 0.01	0.663 ± 0.16	13.4	2001.0	0.951	0.941	14	2001
0.3	0.902 ± 0.01	0.669 ± 0.17	10.6	2001.0	0.931	0.971	13	2001
0.5	0.864 ± 0.02	0.621 ± 0.19	8.4	2001.0	0.931	0.941	5	2001
0.7	0.813 ± 0.04	0.676 ± 0.19	5.3	2001.0	0.873	1.000	3	2001
0.9	0.711 ± 0.06	0.576 ± 0.21	207.9	2001.0	0.863	0.912	2	2001

UNCORRECTED PROOF

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

Table 5
Behavior of the proposed methodology applying neural networks for Leukemia ALL-AML dataset.

P	Type NN	Average accuracy		Best accuracy		Worst accuracy		# of genes
		Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	
O	MLP1	0.819 ± 0.213	0.740 ± 0.170	1.000	0.941	0.289	0.412	3
R	MLP1	0.844 ± 0.185	0.833 ± 0.182	0.983	1.000	0.328	0.357	3
O	MLP2	0.909 ± 0.136	0.832 ± 0.128	1.000	0.941	0.421	0.529	3
R	MLP2	0.910 ± 0.123	0.876 ± 0.146	0.983	1.000	0.345	0.357	3
O	RBF	1.000 ± 0.000	0.559 ± 0.000	1.000	0.559	1.000	0.559	3
R	RBF	1.000 ± 0.000	0.507 ± 0.146	1.000	0.857	1.000	0.286	3
O	RBFE	1.000 ± 0.000	0.500 ± 0.000	1.000	0.500	1.000	0.500	3
R	RBFE	1.000 ± 0.000	0.517 ± 0.127	1.000	0.857	1.000	0.286	3
O	SVMQ	0.974 ± 0.000	0.824 ± 0.000	0.974	0.824	0.974	0.824	3
R	SVMQ	0.960 ± 0.016	0.931 ± 0.066	1.000	1.000	0.931	0.714	3
O	SVML	1.000 ± 0.000	0.853 ± 0.000	1.000	0.853	1.000	0.853	3
R	SVML	0.959 ± 0.012	0.940 ± 0.053	0.983	1.000	0.931	0.786	3
O	SVMP	1.000 ± 0.000	0.912 ± 0.000	1.000	0.912	1.000	0.912	3
R	SVMP	0.964 ± 0.011	0.919 ± 0.059	0.983	1.000	0.948	0.786	3

Tr. cl. = Training classification rate, Te. cl. = Testing classification rate.

For the next two datasets, the best solutions found were different to the solution found with a less quantity of genes. In this sense, we present two different tables for each dataset with the results obtained using the different set of genes.

For the case of the DLBCL dataset, using a solution with the less quantity of genes, the best average accuracy was obtained combining the SVM (linear and polynomial kernel) with the original partition (see Table 7), achieving an accuracy of 65.0%. Furthermore, the best accuracy was reached using a SVM (polynomial kernel) with a random data partition, obtaining an efficiency of 77.1%. Something to remark is that these results were generated using only one gene (called 32064) as found in Table 3.

Table 8 presents the results for the DLBCL problem using the best set of genes found. In this case, 17 genes were found by ABC algorithm (see Table 3). Moreover, Table 8 shows that, on average, it was possible to obtain an accuracy of 61.3% using a SVM (quadratic kernel) with the original partition. The best accuracy achieved was 79.2% using a SVM (quadratic kernel) with random partitions. Taking into account the best classification accuracy, we can observe that by choosing the set of genes with the best accuracy instead of the minimum genes number, the accuracy increases.

Table 9 presents the results for the prostate cancer dataset. These results were generated using a solution with the less quantity of genes, see Table 4. The best average accuracy was obtained with a SVM (linear kernel) over the original partition data, achieving an accuracy of 94.1%. Furthermore, the best accuracy was reached using a MLP2 with a random data partition, obtaining an efficiency of 100%.

Finally, Table 10 represents the results for the prostate cancer dataset using the set of genes that generates best accuracy, (Table 4). The best average accuracy was obtained with the SVM (linear kernel) using the original data partition, achieving an accuracy of 100.0%. Furthermore, the best accuracy reached was 100% using MLP1, MLP2 and SVM (linear kernel).

7. Results and discussion

The results obtained in previous section, show that the ABC algorithm could be considered as a significant dimensional reduction technique for discovering the most representative genes of DNA microarray associated with a particular disease. Moreover, even when the methodology uses less than one percentage of the information, see Table 11, the accuracy obtained during training and testing stages is highly acceptable.

In [31], the authors demonstrate that bioinspired algorithms such as ABC algorithm performs better than PCA methods for selecting the most relevant genes for further classification. Opposite to PCA-based methods that perform a transformation of the DNA microarray into another space, the ABC algorithm selects the most relevant genes from the DNA microarray and the solution is directly associated with a particular gene. This association is a great advantage for understanding which genes are more relevant to detect, predict and classify a specific disease.

On the other hand, we expected to improve the efficiency using an ANN instead of a euclidean distance classifier (EDC). The results obtained corroborated the capabilities of the ANN and suggested

Table 6
Behavior of the proposed methodology applying neural networks for breast cancer dataset.

P	Type NN	Average accuracy		Best accuracy		Worst accuracy		# of genes
		Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	
O	MLP1	0.651 ± 0.142	0.667 ± 0.164	0.846	0.947	0.397	0.263	5
R	MLP1	0.674 ± 0.149	0.682 ± 0.163	0.897	0.947	0.346	0.316	5
O	MLP2	0.750 ± 0.076	0.733 ± 0.153	0.859	0.947	0.551	0.316	5
R	MLP2	0.735 ± 0.105	0.644 ± 0.119	0.910	0.842	0.487	0.421	5
O	RBF	1.000 ± 0.000	0.737 ± 0.000	1.000	0.737	1.000	0.737	5
R	RBF	1.000 ± 0.000	0.547 ± 0.108	1.000	0.737	1.000	0.316	5
O	RBFE	1.000 ± 0.000	0.684 ± 0.000	1.000	0.684	1.000	0.684	5
R	RBFE	1.000 ± 0.000	0.530 ± 0.100	1.000	0.737	1.000	0.368	5
O	SVMQ	0.808 ± 0.000	0.684 ± 0.000	0.808	0.684	0.808	0.684	5
R	SVMQ	0.860 ± 0.031	0.698 ± 0.100	0.923	0.895	0.795	0.526	5
O	SVML	0.795 ± 0.000	0.842 ± 0.000	0.795	0.842	0.795	0.842	5
R	SVML	0.817 ± 0.019	0.775 ± 0.083	0.859	0.895	0.769	0.579	5
O	SVMP	0.949 ± 0.000	0.474 ± 0.000	0.949	0.474	0.949	0.474	5
R	SVMP	0.955 ± 0.017	0.621 ± 0.104	0.987	0.842	0.910	0.474	5

UNCORRECTED PROOF

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

Table 7
Behavior of the proposed methodology applying neural networks for DLBCL-NIH dataset (using the minimum genes number).

P	Type NN	Average accuracy		Best accuracy		Worst accuracy		# of genes
		Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	
O	MLP1	0.535 ± 0.058	0.577 ± 0.115	0.606	0.662	0.431	0.362	1
R	MLP1	0.567 ± 0.061	0.572 ± 0.071	0.620	0.688	0.375	0.438	1
O	MLP2	0.560 ± 0.027	0.632 ± 0.057	0.594	0.675	0.438	0.362	1
R	MLP2	0.598 ± 0.027	0.580 ± 0.076	0.646	0.708	0.526	0.396	1
O	RBF	0.619 ± 0.000	0.613 ± 0.000	0.619	0.613	0.619	0.613	1
R	RBF	0.620 ± 0.016	0.555 ± 0.052	0.646	0.667	0.589	0.458	1
O	RBFE	0.619 ± 0.000	0.613 ± 0.000	0.619	0.613	0.619	0.613	1
R	RBFE	0.618 ± 0.014	0.579 ± 0.072	0.646	0.708	0.599	0.396	1
O	SVMQ	0.562 ± 0.000	0.637 ± 0.000	0.562	0.637	0.562	0.637	1
R	SVMQ	0.605 ± 0.022	0.572 ± 0.080	0.646	0.729	0.562	0.375	1
O	SVML	0.613 ± 0.000	0.650 ± 0.000	0.613	0.650	0.613	0.650	1
R	SVML	0.567 ± 0.023	0.586 ± 0.066	0.620	0.688	0.542	0.417	1
O	SVMP	0.575 ± 0.000	0.650 ± 0.000	0.575	0.650	0.575	0.650	1
R	SVMP	0.604 ± 0.025	0.590 ± 0.069	0.651	0.771	0.526	0.500	1

Tr. cl. = Training classification rate, Te. cl. = Testing classification rate.

Table 8
Behavior of the proposed methodology applying neural networks for DLBCL-NIH dataset (using the genes with the best accuracy).

P	Type NN	Average accuracy		Best accuracy		Worst accuracy		# of genes
		Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	
O	MLP1	0.597 ± 0.081	0.575 ± 0.086	0.762	0.662	0.456	0.338	17
R	MLP1	0.596 ± 0.082	0.572 ± 0.087	0.797	0.729	0.427	0.375	17
O	MLP2	0.665 ± 0.079	0.549 ± 0.077	0.856	0.662	0.544	0.362	17
R	MLP2	0.680 ± 0.098	0.567 ± 0.069	0.938	0.750	0.458	0.438	17
O	RBF	1.000 ± 0.000	0.600 ± 0.000	1.000	0.600	1.000	0.600	17
R	RBF	1.000 ± 0.000	0.569 ± 0.077	1.000	0.708	1.000	0.417	17
O	RBFE	1.000 ± 0.000	0.588 ± 0.000	1.000	0.588	1.000	0.588	17
R	RBFE	1.000 ± 0.000	0.548 ± 0.057	1.000	0.646	1.000	0.438	17
O	SVMQ	1.000 ± 0.000	0.613 ± 0.000	1.000	0.613	1.000	0.613	17
R	SVMQ	1.000 ± 0.000	0.591 ± 0.064	1.000	0.792	1.000	0.521	17
O	SVML	0.681 ± 0.000	0.588 ± 0.000	0.681	0.588	0.681	0.588	17
R	SVML	0.663 ± 0.020	0.572 ± 0.061	0.703	0.688	0.625	0.458	17
O	SVMP	1.000 ± 0.000	0.500 ± 0.000	1.000	0.500	1.000	0.500	17
R	SVMP	1.000 ± 0.000	0.564 ± 0.061	1.000	0.688	1.000	0.417	17

Tr. cl. = Training classification rate, Te. cl. = Testing classification rate.

that, selecting the correct set of genes, it is possible to detect, predict and classify a particular disease with an acceptable accuracy. In general, we observed that the accuracy of the proposed methodology increased when the ANN was training with the set of genes discovered in the first stage of the proposal. We also observed that in general, the accuracy obtained by the ANN was better than the obtained with Euclidean Distance.

Concerning to the types of ANN used in this research, we observed that the MLP1 and MLP2 achieved a better accuracy than the SVM and RBF. However, this observation does not mean that the MLP is the best neuronal model to solve this problem because in some cases the SVM provide similar results. It is necessary further analysis over the parameters and topologies of these models to assert which is the best neural model. Although, the suggested analysis is out of the scope of this paper, we could say that, with the selected parameters, the goal of this research was successfully covered. We observed an improvement when the ANN was trained with the most relevant set of genes found with this proposal.

Table 9
Behavior of the proposed methodology applying neural networks for Prostate tumor dataset (using the minimum genes number).

P	Type NN	Average accuracy		Best accuracy		Worst accuracy		# of genes
		Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	
O	MLP1	0.738 ± 0.167	0.775 ± 0.237	0.892	0.941	0.490	0.265	2
R	MLP1	0.825 ± 0.127	0.781 ± 0.134	0.927	0.926	0.422	0.259	2
O	MLP2	0.813 ± 0.113	0.867 ± 0.151	0.902	0.941	0.490	0.265	2
R	MLP2	0.839 ± 0.105	0.827 ± 0.139	0.927	1.000	0.578	0.519	2
O	RBF	0.951 ± 0.000	0.559 ± 0.000	0.951	0.559	0.951	0.559	2
R	RBF	0.948 ± 0.015	0.788 ± 0.071	0.991	0.926	0.927	0.630	2
O	RBFE	0.951 ± 0.000	0.559 ± 0.000	0.951	0.559	0.951	0.559	2
R	RBFE	0.938 ± 0.014	0.791 ± 0.076	0.963	0.963	0.917	0.630	2
O	SVMQ	0.863 ± 0.000	0.882 ± 0.000	0.863	0.882	0.863	0.882	2
R	SVMQ	0.883 ± 0.016	0.889 ± 0.061	0.917	0.963	0.853	0.741	2
O	SVML	0.882 ± 0.000	0.941 ± 0.000	0.882	0.941	0.882	0.941	2
R	SVML	0.880 ± 0.017	0.883 ± 0.062	0.917	0.963	0.853	0.741	2
O	SVMP	0.892 ± 0.000	0.882 ± 0.000	0.892	0.882	0.892	0.882	2
R	SVMP	0.899 ± 0.015	0.869 ± 0.048	0.936	0.963	0.872	0.778	2

UNCORRECTED PROOF

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

Table 10
Behavior of the proposed methodology applying neural networks for Prostate tumor dataset (using the genes with the best accuracy).

P	Type NN	Average accuracy		Best accuracy		Worst accuracy		# of genes
		Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	
O	MLP1	0.634 ± 0.143	0.695 ± 0.315	0.824	1.000	0.490	0.265	3
R	MLP1	0.758 ± 0.136	0.756 ± 0.177	0.881	0.963	0.413	0.296	3
O	MLP2	0.753 ± 0.094	0.794 ± 0.201	0.882	1.000	0.412	0.235	3
R	MLP2	0.763 ± 0.160	0.747 ± 0.156	0.927	0.926	0.321	0.296	3
O	RBF	0.990 ± 0.000	0.412 ± 0.000	0.990	0.412	0.990	0.412	3
R	RBF	0.993 ± 0.007	0.557 ± 0.108	1.000	0.741	0.982	0.333	3
O	RBFE	0.980 ± 0.000	0.412 ± 0.000	0.980	0.412	0.980	0.412	3
R	RBFE	0.991 ± 0.008	0.565 ± 0.090	1.000	0.741	0.982	0.296	3
O	SVMQ	0.804 ± 0.000	0.971 ± 0.000	0.804	0.971	0.804	0.971	3
R	SVMQ	0.861 ± 0.024	0.835 ± 0.077	0.908	0.963	0.817	0.704	3
O	SVML	0.784 ± 0.000	1.000 ± 0.000	0.784	1.000	0.784	1.000	3
R	SVML	0.841 ± 0.018	0.851 ± 0.061	0.872	0.926	0.807	0.704	3
O	SVMP	0.853 ± 0.000	0.559 ± 0.000	0.853	0.559	0.853	0.559	3
R	SVMP	0.884 ± 0.020	0.815 ± 0.074	0.927	0.926	0.844	0.667	3

Tr. cl. = Training classification rate, Te. cl. = Testing classification rate.

Table 11
Dimensional reduction capabilities of the proposed methodology.

Dataset	# of genes			% of information	
	Original	Best	Min	Best	Min
ALL-AML	7129	3	3	0.0420	0.0420
Breast	24,481	5	5	0.0204	0.0204
DLBCL	7399	17	1	0.2297	0.0135
Prostate	12,600	3	2	0.0238	0.0158

Table 12
Best accuracy obtained with different classification technique using the original partition.

Dataset	ALL-AML dataset		Breast dataset		DLBCL dataset		Prostate dataset	
	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.
EDC	1.000	0.882	0.833	0.842	0.756	0.675	0.873	1.000
MLP1	1.000	0.941	0.846	0.947	0.762	0.662	0.824	1.000
MLP2	1.000	0.941	0.859	0.947	0.856	0.662	0.882	1.000
RBF	1.000	0.559	1.000	0.737	1.000	0.600	0.990	0.412
RBFE	1.000	0.500	1.000	0.684	1.000	0.588	0.980	0.412
SVMQ	0.974	0.824	0.808	0.684	1.000	0.613	0.804	0.971
SVML	1.000	0.853	0.795	0.842	0.681	0.588	0.784	1.000
SVMP	1.000	0.912	0.949	0.474	1.000	0.500	0.853	0.559

Tables 12 and 13 show a comparison for each different cancer problem in terms of the best accuracy obtained with the Euclidean distance classifier and different artificial neural networks. Something interesting to observe is the fact that the accuracy of the ANN increases when random partitions are used.

Finally, in Table 14, we compare the results against those obtained by other authors using different techniques. In general, we observed that the proposed methodology performs better than those revised from the literature (for the case of ALL-AML and Prostate datasets).

From these results, we observed that for the case of the Leukemia AML-ALL problem, the MLP1, MLP2, SVML, SVMQ and SVMP with a random partition achieved better accuracy than the Euclidean distance. Furthermore, compared against other techniques, our proposal achieved an accuracy of 100% using only three genes. For the breast cancer problem, the best accuracy was obtained with the MLP1 achieving 94.7% with random partition. Although other technique achieved 100% of accuracy using 20 genes, our proposal used only five genes. This result represents a trade-off between accuracy and resource efficiency.

Table 13
Best accuracy obtained with different classification technique using random partition.

Dataset	ALL-AML dataset		Breast dataset		DLBCL dataset		Prostate dataset	
	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.	Tr. cl.	Te. cl.
EDC	1.000	0.882	0.833	0.842	0.756	0.675	0.873	1.000
MLP1	0.983	1.000	0.897	0.947	0.797	0.729	0.881	0.963
MLP2	0.983	1.000	0.910	0.842	0.938	0.750	0.927	0.926
RBF	1.000	0.857	1.000	0.737	1.000	0.708	1.000	0.741
RBFE	1.000	0.857	1.000	0.737	1.000	0.646	1.000	0.741
SVMQ	1.000	1.000	0.923	0.895	1.000	0.792	0.908	0.963
SVML	0.983	1.000	0.859	0.895	0.703	0.688	0.872	0.926
SVMP	0.983	1.000	0.987	0.842	1.000	0.688	0.927	0.926

UNCORRECTED PROOF

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

Table 14
Comparison against other techniques.

Dataset	Classification technique	Selection technique	# of genes	% of accuracy	References
ALL-AML	MLP	ABC	3	1.0000	Proposed methodology
ALL-AML	SVM	PSO	10	1.0000	[30]
ALL-AML	SVM	PLS-RFE	16	1.0000	[46]
ALL-AML	KNN	MAHP	5	0.9736	[47]
ALL-AML	SMV	GBC	3	0.9583	[28]
ALL-AML	KNN	DRF-CONS	4	0.9412	[48]
Breast	SVM	PSO	20	1.0000	[30]
Breast	SVM	ABC	5	0.9470	Proposed methodology
Breast	J48	GA	41	0.9381	[29]
Breast	SMV	DRF0-IG	44	0.8421	[48]
Prostate	MLP	ABC	3	1.000	Proposed methodology
Prostate	SVM	PLS-RFE	11	0.9804	[46]
Prostate	SMV	DRF-IG	52	0.9706	[48]
Prostate	LDA	GA-M2/E2	18	0.9550	[49]
Prostate	LDA	MAHP	5	0.9118	[47]

The obtained results show that the DLBCL-NIH is a challenging dataset. The best accuracy was achieved with the SVMQ, obtaining an efficiency of 79.2% using a random partition with 17 genes.

The EDC, MLP1, MLP2 and SVLM achieved an accuracy of 100% for the Prostate cancer dataset with the original partition.

8. Conclusions

The extensive experimentation allowed us to determine the behavior of the proposed methodology in the classification of DNA microarrays. In the first stage of this proposal, the dimensionality reduction was applied in order to select the set of genes that best describe a particular disease. We posted the problem of dimensional reduction as an optimization problem, due to the dimensional reduction of a DNA microarray could be seen as a combinatorial problem trying to find from millions of genes the most relevant. The results obtained with the proposed methodology suggested that the ABC algorithm is a competitive candidate for reducing the dimensionality of a microarray. The results obtained used less than the one percentage of the genes for performing a detection or classification task. Furthermore, solutions using only one gene were found. On the other hand, we also observed that the threshold value for selecting the best set of genes directly impacted on the accuracy of the proposal.

Once discover the best set of genes, we evaluated the accuracy of a simple distance classifier. The results obtained showed that all dataset were solved with a good precision. Nonetheless, in the second stage, we tried to improve the results using an ANN.

During the second stage, we evaluated the capabilities of the ANN for correctly classified the different diseases. These ANN were trained using the set of genes discovered by the proposed methodology during the first stage. We also compared the accuracy achieved with three different ANN: the MLP, SVM and the RBF neural networks. Through several experiments, we observed that ANN obtained better results compared to those reached with the distance classifier. Furthermore, the ANN was trained with two different set of genes, the best set of genes and the solution with the less quantity of genes. The results showed that the best accuracy was achieved when the ANN was trained with the best set of genes. Nonetheless, the solution with the less quantity of genes also provide better results than the distance classifier.

On the other hand, the experimental results showed that MLP and SVM achieved a better performance than the results obtained with the RBF. In this sense, it is necessary to perform a broad analysis of the parameters, and the topology used to train and design the ANN. Finally, we concluded that the proposed methodology is capable of selecting the correct set of genes to detect, predict and classify a particular disease with an acceptable accuracy.

Nowadays, we are evaluating a new fitness function to improve the accuracy of the proposed methodology and reduce, at the same time, the number of genes used to train the ANN. In addition, we are preparing a deep analysis of the parameters used to train the ANN.

UNCORRECTED PROOF

Acknowledgements

The authors thank DGAPA, UNAM and Universidad La Salle for the economic support under grants number IN107214 and NEC-03/15, respectively. Beatriz Garro thanks CONACYT for the posdoctoral scholarship.

References

[1] 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, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[2] A. R. C.G., Á. E., Z. de la Rosa G., Á. N. M., G. P., Microarreglos de adn y cáncer cervicouterino: identificación de marcadores tumorales, Ginecología y obstetricia de México 75 (2007) 205–213.
[3] C. Sotiriou, M.J. Piccart, Taking gene-expression profiling to the clinic: when will molecular signatures become relevant to patient care? Nat. Rev. Cancer 7 (July) (2007) 545–553.
[4] M.J. van de Vijver, Y.D. He, L.J. van 't Veer, H. Dai, A.A. Hart, D.W. Voskuil, G.J. Schreiber, J.L. Peterse, C. Roberts, M.J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E.T. Rutgers, S.H. Friend, R. Bernards, A gene-expression signature as a predictor of survival in breast cancer, N. Engl. J. Med. 347 (2002) 1999–2009 (PMID: 12490681).
[5] L.B. Bribiesca, Los microarreglos de dna y su aplicación clínica, Acta Médica Grupo Ángeles 2 (2004) 125–127.
[6] C. Vallin Plous, Microarreglos de adn y sus aplicaciones en investigaciones biomédicas, Revista CENIC. Ciencias Biológicas 38 (2007) 132–135.
[7] T.K. Karakach, R.M. Flight, S.E. Douglas, P.D. Wentzell, An introduction to DNA microarrays for gene expression analysis, Chemomet. Intell. Lab. Syst. 104 (2010) 28–52.
[8] H.M. Alshamlan, G.H. Badr, Y. Alohali, A study of cancer microarray gene expression profile: objectives and approaches, in: Proceedings of the World Congress on Engineering 2, 2013.
[9] R.M.L. Baena, D. Urda, J.L. Subirats, L. Franco, J.M. Jerez, Analysis of cancer microarray data using constructive neural networks and genetic algorithms, in: I. Rojas, F.M.O. Guzman (Eds.), International Work-Conference on Bioinformatics and Biomedical Engineering, IWBBIO 2013, Granada, Spain, March 18–20, 2013, Proceedings, Copicentro Editorial, 2013, pp. 55–63.
[10] Q. Shen, W.-M. Shi, W. Kong, Research article: Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data, Comput. Biol. Chem. 32 (2008) 53–60.
[11] J. Cho, D. Kim, Intelligent feature selection by bacterial foraging algorithm and information theory, in: T.-H. Kim, H. Adeli, R. Robles, M. Balitanas (Eds.), Advanced Communication and Networking, volume 199 of Communications in Computer and Information Science, Springer, Berlin, Heidelberg, 2011, pp. 238–244.

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002

[12] L. Xiao, A clustering algorithm based on artificial fish school, in: in: 2010 2nd International Conference on Computer Engineering and Technology (ICCET), volume 7, 2010, V7-766-V7-769.
[13] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[14] A.K. Jain, J. Mao, K.M. Mohiuddin, Artificial neural networks: a tutorial, Computer 29 (1996) 31–44.
[15] L. Peterson, M. Ozen, H. Erdem, A. Amini, L. Gomez, C. Nelson, M. Ittmann, Artificial neural network analysis of dna microarray-based prostate cancer recurrence, in: in: CIBCB'05. Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, 2005, pp. 1–8.
[16] L.J. Lancashire, C. Lemetre, G.R. Ball, An introduction to artificial neural networks in bioinformatics application to complex microarray and mass spectrometry datasets in cancer studies, Brief. Bioinform. 10 (2009) 315–329.
[17] J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, P.S. Meltzer, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nat. Med. 7 (2001) 673–679.
[18] J. Dela Rosa, A. Magpantay, A. Gonzaga, G. Solano, Cluster center genes as candidate biomarkers for the classification of leukemia, in: in: The 5th International Conference on Information, Intelligence, Systems and Applications, IISA 2014, 2014, pp. 124–129.
[19] C.D.A. Vanitha, D. Devaraj, M. Venkatesulu, Gene expression data classification using support vector machine and mutual information-based gene selection, Procedia Comp. Sci. 47 (2015) 13–21 (Graph Algorithms, High Performance Implementations and Its Applications (ICGHIA 2014)).
[20] W. Chen, H. Lu, M. Wang, C. Fang, Gene expression data classification using artificial neural network ensembles based on samples filtering, in: in: International Conference on Artificial Intelligence and Computational Intelligence AICI'09, volume 1, 2009, pp. 626–628.
[21] H.T. Huynh, J. ja Kim, Y. Won, Classification study on DNA micro array with feed forward neural network trained by singular value decomposition, Int. J. Bio-Sci. Bio-Technol. (2009) 17–24.
[22] L. Peterson, M. Coleman, Comparison of gene identification based on artificial neural network pre-processing with k-means cluster and principal component analysis, in: I. Bloch, A. Petrosino, A. Tettamanzi (Eds.), Fuzzy Logic and Applications, volume 3849 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2006, pp. 267–276.
[23] H. Yu, S. Hong, X. Yang, J. Ni, Y. Dan, B. Qin, Recognition of multiple imbalanced cancer types based on dna microarray data using ensemble classifiers, BioMed Res. Int. (2013) 13.
[24] F. Fernandez-Navarro, C. Hervás-Martínez, R. Ruiz, J.C. Riquelme, Evolutionary generalized radial basis function neural networks for improving prediction accuracy in gene classification using feature selection, Appl. Soft Comput. 12 (2012) 1787–1800.
[25] J.W. Catto, M.F. Abbod, P.J. Wild, D.A. Linkens, C. Pilarsky, I. Rehman, D.J. Rosario, S. Denzinger, M. Burger, R. Stoehr, R. Knuechel, A. Hartmann, F.C. Hamdy, The application of artificial intelligence to microarray data: Identification of a novel gene signature to identify bladder cancer progression, Eur. Urol. 57 (2010) 398–406.
[26] H. Yu, J. Ni, J. Zhao, Acosampling: an ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data, Neurocomputing 101 (2013) 309–318.
[27] S. Ghorai, A. Mukherjee, S. Sengupta, P.K. Dutta, Cancer classification from gene expression data by NPPC ensemble, IEEE/ACM Trans. Comput. Biol. Bioinform. 8 (2011) 659–671.
[28] H.M. Alshamlan, G.H. Badr, Y.A. Alohali, Genetic bee colony (GBC) algorithm: a new gene selection method for microarray cancer classification, Comput. Biol. Chem. 56 (2015) 49–60.
[29] S. Sasikala, S.A. alias Balamurugan, S. Geetha, A novel feature selection technique for improved survivability diagnosis of breast cancer, Procedia Comp. Sci. 50 (2015) 16–23 (Big data, cloud and computing challenges).
[30] B. Sahu, D. Mishra, A novel feature selection algorithm using particle swarm optimization for cancer microarray data, Procedia Eng. 38 (2012) 27–31 (International Conference on Modelling Optimization and Computing).
[31] B.A. Garro, R.A. Vazquez, K. Rodríguez, Classification of dna microarrays using artificial bee colony (abc) algorithm, in: Y. Tan, Y. Shi, C. Coello (Eds.), Advances in Swarm Intelligence, volume 8794 of Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 207–214.
[32] J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, H.O. Smith, M. Yandell, C.A. Evans, R.A. Holt, The sequence of the human genome, Science 291 (2001) 1304–1351.
[33] E.S. Lander, et al., Initial sequencing and analysis of the human genome, Nature 409 (2001) 860–921.
[34] V. Trevino, F. Falciani, H.A. Barrera-Salda na, DNA microarrays: a powerful genomic tool for biomedical and clinical research, Mol. Med. (Cambridge, MA) 13 (2007) 527–541.
[35] G. L. L., C. M. A., Microarreglos: herramienta para el conocimiento de las enfermedades, Revista Colombiana de Reumatología 12 (2005) 263-267.
[36] F.J.T. Staal, M. van der Burg, L.F.A. Wessels, B.H. Barendregt, M.R.M. Baert, C.M.M. van den, C. Burg, A.W. Van Huffel, V.H.J. Langerak, M.J.T. van der Velden, J.J.M. Reinders, van Dongen, DNA microarrays for comparison of gene expression profiles between diagnosis and relapse in precursor-b acute lymphoblastic leukemia: choice of technique and purification influence the identification of potential diagnostic markers, Leukemia 17 (2003) 1324–1332.
[37] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.
[38] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Internal Representations by Error Propagation, MIT Press, Cambridge, MA, USA, pp. 673–695. Q4
[39] J.A. Anderson, An Introduction to Neural Networks, The MIT Press, 1995.
[40] P.J. Werbos, Backpropagation through time: what it does and how to do it, Proc. IEEE 78 (2002) 1550–1560.
[41] E. Alba, et al., Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms, in: IEEE CEC 2007, 2007, pp. 284–290.
[42] M. Buscema, The general philosophy of the artificial adaptive systems, in: V. Capecchi, M. Buscema, P. Contucci, B. D'Amore (Eds.), Applications of Mathematics in Models, Artificial Neural Networks and Arts, Springer, Netherlands, 2010, pp. 197–226.
[43] L.J. van 't Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A.M. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, S.H. Friend, Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (2002) 530–536.
[44] A. Rosenwald, G. Wright, W.C. Chan, J.M. Connors, E. Campo, R.I. Fisher, R.D. Gascoyne, H.K. Muller-Hermelink, E.B. Smeland, J.M. Giltnane, E.M. Hurt, H. Zhao, L. Averett, L. Yang, W.H. Wilson, E.S. Jaffe, R. Simon, R.D. Klausner, J. Powell, P.L. Duffey, D.L. Longo, T.C. Greiner, D.D. Weisenburger, W.G. Sanger, B.J. Dave, J.C. Lynch, J. Vose, J.O. Armitage, E. Montserrat, A. Lpez-Guillermo, T.M. Grogan, T.P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, J. Delabie, H. Holte, P. Krajci, T. Stokke, L.M. Staudt, The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma, N. Engl. J. Med. 346 (2002) 1937–1947.
[45] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. D'Amico, J.P. Richie, Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209.
[46] A. Wang, N. An, G. Chen, L. Li, G. Alterovitz, Improving plsrfe based gene selection for microarray data classification, Comp. Biol. Med. 62 (2015) 14–24.
[47] T. Nguyen, A. Khosravi, D. Creighton, S. Nahavandi, A novel aggregate gene selection method for microarray data classification, Pattern Recogn. Lett. 6061 (2015) 16–23.
[48] V. Boln-Canedo, N. Snchez-Maroo, A. Alonso-Betanzos, Distributed feature selection: an application to microarray data classification, Appl. Soft Comput. 30 (2015) 136–150.
[49] E. Bonilla Huerta, B. Duval, J.-K. Hao, Gene selection for microarray data by a LDA-based genetic algorithm, in: M. Chetty, A. Ngom, S. Ahmad (Eds.), Pattern Recognition in Bioinformatics, volume 5265 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2008, pp. 250–261.

UNCORRECTED PROOF

Please cite this article in press as: B.A. Garro, et al., Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.10.002