Neurocomputing 73 (2010) 2375–2383

Contents lists available at ScienceDirect

Neurocomputing

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

A hybrid LDA and genetic algorithm for gene selection and classification of microarray data

Edmundo Bonilla Huerta [a], Béatrice Duval [b], Jin-Kao Hao [b,*]

[a] Instituto Tecnologico de Apizaco, av Instituto Tecnologico S/N, Apizaco, Tlaxcala 90300, Mexico
[b] LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49045 Angers, France

ARTICLE INFO

Available online 31 May 2010

Keywords:
Gene selection
Classification
Dedicated genetic algorithm
Linear discriminant analysis

ABSTRACT

In supervised classification of Microarray data, gene selection aims at identifying a (small) subset of informative genes from the initial data in order to obtain high predictive accuracy. This paper introduces a new embedded approach to this difficult task where a genetic algorithm (GA) is combined with Fisher's linear discriminant analysis (LDA). This LDA-based GA algorithm has the major characteristic that the GA uses not only a LDA classifier in its fitness function, but also LDA's discriminant coefficients in its dedicated crossover and mutation operators. Computational experiments on seven public datasets show that under an unbiased experimental protocol, the proposed algorithm is able to reach high prediction accuracies with a small number of selected genes.

1. Introduction

The DNA microarray technology permits to monitor and to measure gene expression levels for 10s of 1000s of genes simultaneously in a cell mixture. This technology enables to consider cancer diagnosis based on gene expressions [3,5,1,16]. Given the very high number of genes, it is useful to select a limited number of relevant genes for classifying tissue samples.

Three main approaches have been proposed for gene selection. *Filter methods* achieve gene selection independently of the classification model. They rely on a criterion that depends only on the data to assess the importance or relevance of each gene for class discrimination. A relevance scoring provides a ranking of the genes from which the top-ranking ones are generally selected as the most relevant genes. As many filter methods are univariate, they ignore the correlations between genes and provide gene subsets that may contain redundant information.

In *wrapper approaches*, gene subset selection is performed in interaction with a classifier. The goal is to find a gene subset that achieves the best prediction performance for a particular learning model. To explore the space of gene subsets, a search algorithm (e.g. a genetic algorithm—GA) is "wrapped" around the classification model which is used as a black box to assess the predictive quality of the candidate gene subsets. Wrapper methods are computationally intensive since a classifier is built for each candidate subset.

Embedded methods are similar to wrapper methods in the sense that the search of an optimal subset is performed for a specific learning algorithm, but they are characterized by a deeper interaction between gene selection and classifier construction.

Generally, wrapper and embedded methods use a search algorithm for the purpose of exploring a given space of gene subsets. Among different options, GAs are probably the most popular choice. Indeed, one finds from the literature a large number of studies using GAs (often in combination with other techniques) for gene selection. For some representative examples, see the following references [23,34,31,20,24,6,29,50,19,26]. All these methods share a set of common characteristics: they represent a pre-selected gene subsets by a binary vector, employ standard (blind) or specific crossover and mutation operators to evolve a population and use a specific classifier (kNN, SVM…) for fitness evaluation.

In this paper, we propose a new embedded approach to gene subset selection for Microarray data classification. Starting with a set of pre-selected genes using a filter (typically more than 100 genes), we use a dedicated genetic algorithm combined with Fisher's linear discriminant analysis (LDA) to explore the space of gene subsets. More specifically, our dedicated GA uses the LDA classifier to assess the fitness of a given candidate gene subset and LDA's discriminant coefficients to inform the genetic search operators. This LDA-based GA is to be contrasted with many GAs that rely only on random search operators without taking into account problem knowledges.

To evaluate the usefulness of the proposed LDA-based GA, we carry out extensive experiments on seven public datasets and compare our results with 15 best performing algorithms from the literature. We observe that our approach is able to achieve a high prediction accuracy (from 96% to 100%) with a small number of

* Corresponding author.
E-mail address: hao@info.univ-angers.fr (J.-K. Hao).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.03.024

informative genes (often less than 20). To understand the behavior of the proposed algorithm, we also study the impact of the LDA-based genetic operators on the evolutionary process as well as other important parameters of the proposed algorithm.

The remainder of this paper is organized as follows. Section 2 recalls the main characteristics of Fisher's LDA and discusses the calculus that must be done in the case of samples of small size. Section 3 presents our LDA-based GA for gene selection and classification. Section 4 shows the experimental results and comparisons. Section 5 is dedicated to additional studies on the LDA-based crossover operator and other important parameters of the proposed algorithm. Finally conclusions are provided in Section 6.

2. LDA and small sample size problem

2.1. Linear discriminant analysis

LDA is a well-known dimension reduction and classification method, where the data are projected into a low dimension space such that the classes are well separated. LDA has recently been used for Microarray data analysis [12,47,48].

As we use this method for binary classification problems, we shall restrict the explanations to this case. We consider a set of n samples belonging to two classes C_1 and C_2, with n_1 samples in C_1 and n_2 samples in C_2. Each sample is described by q variables. So the data form a matrix $X=(x_{ij})$,i$=1,\ldots,n$; $j=1,\ \ldots,q$. We denote by μ_k the mean of class C_k and by μ the mean of all the samples:

$$\mu_k = \frac{1}{n_k}\sum_{x_i \in C_k} x_i \quad \text{and} \quad \mu = \frac{1}{n}\sum_{x_i} x_i = \frac{1}{n}\sum_k n_k \mu_k$$

The data are described by two matrices S_B and S_W, where S_B is the between-class scatter matrix and S_W the within-class scatter matrix defined as follows:

$$S_B = \sum_k n_k(\mu_k-\mu)(\mu_k-\mu)^t \tag{1}$$

$$S_W = \sum_k \sum_{x_i \in C_k} (x_i-\mu_k)(x_i-\mu_k)^t \tag{2}$$

If we denote by S_V the covariance matrix for all the data, we have $S_V=S_B+S_W$.

LDA seeks a linear combination of the initial variables on which the means of the two classes are well separated, measured relatively to the sum of the variances of the data assigned to each class. For this purpose, LDA determines a vector w such that w^tS_B w is maximized while w^tS_W w is minimized. This double objective is realized by the vector w_{opt} that maximizes the criterion:

$$J(w) = \frac{w^tS_Bw}{w^tS_Ww} \tag{3}$$

One can prove that the solution w_{opt} is the eigen vector associated to the sole eigen value of $S_W^{-1}S_B$, when S_W^{-1} exists. Once this axis w_{opt} is determined, LDA provides a classification procedure (classifier), but in our case we are particularly interested in the *discriminant coefficients* of this vector: the absolute value of these coefficients indicates the importance of the q initial variables for the class discrimination.

2.2. Generalized LDA for small sample size problems

When the sample size n is smaller than the dimensionality of samples q, S_W is singular, and it is not possible to compute S_W^{-1}. To overcome the singularity problem, recent works have proposed different methods like the null space method [48], orthogonal LDA [46], uncorrelated LDA [47,46] (see also [33] for a comparison of these methods). The two last techniques use the pseudo inverse method to solve the small sample size problem and this is the approach we apply in this work. When S_w is singular, the eigen problem is solved for $S_w^+S_b$, where S_w^+ is the pseudo inverse of S_w. The pseudo-inverse of a matrix can be computed by singular value decomposition. For a matrix A of size $m\times p$, the singular values of A, noted σ_i, are the non-negative square roots of A^tA. The singular value decomposition of A is $A=U\Sigma V^t$, where U of size $m\times m$ and V of size $n\times n$ are orthogonal and

$$\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$$

with D a diagonal matrix containing the elements σ_i. If we note Σ^+ the matrix

$$\Sigma^+ = \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

(where the 0 blocks have the appropriate sizes), the pseudo inverse of A is then defined as $A^+ = V\Sigma^+U^t$.

2.3. Application to gene selection

Microarray data generally contain less than 100 samples described by at least several 1000s of genes. We limit this high dimensionality by a first pre-selection step, where a filter criterion (t-statistic here) is applied to determine a subset of relevant genes (see Section 4.2 for more details). In this work, we typically retain 100 genes from which an intensive exploration is realized using a genetic algorithm to select smaller subsets. In this process, LDA is used as a classification method to evaluate the classification accuracy that can be achieved on a candidate gene subset. Moreover the coefficients of the eigen vector calculated by LDA are used to evaluate the importance of each gene for class discrimination.

For a selected gene subset of size p, if $p\le n$, we rely on the classical LDA (Section 2.1) to obtain the projection vector w_{opt}, otherwise we apply the generalized LDA (Section 2.2) to obtain this vector. We explain in Section 3 how the LDA-based GA explores the search space of gene subsets.

3. LDA-based genetic algorithm

Given a set of p top ranking genes pre-selected with a filter (typically $p\ge 100$, in this work, $p=100$ or 150), our LDA-based GA is used to conduct a combinatorial search within the space of size 2^p. The purpose of this search is to determine among the possible gene combinations small sized gene subsets allowing a high predictive accuracy. In what follows, we present the general procedure and then describe the components of the LDA-based genetic algorithm. In particular, we explain how LDA is combined with the Genetic Algorithm.

3.1. General GA procedure

Our LDA-based genetic algorithm (LDA-GA) follows the conventional schema of a generational GA and uses also an elitism strategy.

- *Initial population*: The initial population is generated randomly in such a way that each chromosome contains a number of genes ranging from $p\times 0.6$ to $p\times 0.75$. The population size is fixed at 100 in this work.
- *Evolution*: The chromosomes of the current population P are sorted according to the fitness function (see Section 3.3). The "best" 10% chromosomes of P are directly copied to the next

population P' and removed from P. The remaining 90% chromosomes of P' are then generated by using crossover and mutation.

- *Crossover and mutation*: Parent chromosomes are determined from the remaining chromosomes of P by considering each pair of adjacent chromosomes. By applying our specialized crossover operator (see Section 3.4), one child is created each time. This child undergoes then a mutation operation (see Section 3.5) before joining the next population P'.
- *Stop condition*: The evolution process ends when a pre-defined number of generations is reached or when one finds a chromosome in the population having a very small gene subset (fixed at two genes in this work).

Notice that LDA-GA shares some similarities (encoding) with the embedded genetic algorithm of [19], but remains different as to the design of the dedicated crossover and mutation operators.

3.2. Chromosome encoding

Conventionally, a chromosome is used simply to represent a candidate gene subset. In our GA, a chromosome encodes more information and is defined by a couple:

$$I=(\tau;\phi)$$

where τ and ϕ have the following meaning. The first part (τ) is a *binary vector* and effectively represents a *candidate gene subset*. Each allele τ_i indicates whether the corresponding gene g_i is selected ($\tau_i=1$) or not selected ($\tau_i=0$). The second part of the chromosome (ϕ) is a real-valued vector where each ϕ_i corresponds to the *discriminant coefficient* of the eigen vector for gene g_i. As explained in Section 2, the discriminant coefficient defines the contribution of gene g_i to the projection axis w_{opt}. A chromosome thus can be represented as follows:

$$I=(\tau_1,\tau_2,\ldots,\tau_p;\phi_1,\phi_2,\ldots,\phi_p)$$

The length of τ and ϕ is defined by p, the number of genes preselected with the t-statistic filter.

Notice that this chromosome encoding is more general and richer than those used in most genetic algorithms for feature selection in the sense that in addition to the candidate gene subset, the chromosome includes other information (LDA discriminant coefficients here) which are useful for designing powerful crossover and mutation operators (see Sections 3.4 and 3.5).

3.3. Fitness evaluation

The purpose of the genetic search in our LDA-GA approach is to seek "good" gene subsets having the minimal size and the highest prediction accuracy. To achieve this double objective, we devise a fitness function taking into account these (somewhat conflicting) criteria.

To evaluate a chromosome $I=(\tau;\phi)$, the fitness function considers the classification accuracy of the chromosome (f_1) and the number of selected genes in the chromosome (f_2). More precisely, f_1 is obtained by evaluating the classification accuracy of the gene subset τ using the LDA classifier on the training dataset and is formally defined as follows[1]:

$$f_1(I)=\frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

where TP and TN represent, respectively, the true positive and true negative samples, i.e. the correct classifications; FP (FN) is the number of negative (positive) samples misclassified into the positive (negative) samples.

The second part of the fitness function f_2 is calculated by the formula:

$$f_2(I)=\left(1-\frac{m_\tau}{p}\right) \tag{5}$$

where m_τ is the number of bits having the value "1" in the candidate gene subset τ, i.e. the number of selected genes; p is the length of the chromosome corresponding to the number of the pre-selected genes from the filter ranking.

Then the fitness function f is defined as the following weighted aggregation:

$$f(I)=\alpha f_1(I)+(1-\alpha)f_2(I) \quad \text{subject to } 0\le\alpha\le 1 \tag{6}$$

where α is a parameter that allows us to allocate a relative importance factor to f_1 or f_2. Assigning to α a value greater than 0.5 will push the genetic search toward solutions of high classification accuracy (probably at the expense of having more selected genes). Inversely, using small values of α helps the search toward small sized gene subsets. So varying α will change the search direction of the genetic algorithm. In Section 5.2, we provide a study on the influence of this parameter on the performance of the algorithm.

Finally, notice that f takes values from interval [0,1]; a solution with a large value is then better than a solution with a small value.

3.4. LDA-based crossover

It is now widely acknowledged that, whenever it is possible, genetic operators such as crossover and mutation should be tailored to the target problem. In other words, in order for genetic operators to fully play their role, it is preferable to integrate problem-specific knowledges into these operators. In our case, we use the discriminant coefficients from the LDA classifier to design our crossover and mutation operators. Here, we explain how our LDA-based crossover operates.

The crossover combines two parent chromosomes I^1 and I^2 to generate a new chromosome I^c in such a way that (1) top ranking genes in both parents are conserved in the child and (2) the number of selected genes in the child I^c is no greater than the number of selected genes in the parents. The first point ensures that "good" genes are transmitted from one generation to another while the second property is coherent with the optimization objective of small-sized gene subsets.

More formally, let $I^1=(\tau^1;\phi^1)$ and $I^2=(\tau^2;\phi^2)$ be two parent chromosomes, $I^c=(\tau^c;\phi^c)$ the child which will be generated by crossover, $\kappa\in[0,1)$ a parameter indicating the percentage of genes that will not be transmitted from the parents to the child. Then our LDA-based crossover performs the following steps to generate I^c, the child chromosome.

1. According to κ determine the number of genes of I^1 and I^2 (more precisely, τ^1 and τ^2) that will be discarded, denote them by n_1 and n_2.
2. Remove, respectively, from τ^1 and τ^2, the n_1 and n_2 least ranking genes according to the LDA discriminant coefficients.
3. Merge the modified τ^1 and τ^2 by the logic AND operator to generate τ^c.
4. Apply the LDA classifier to τ^c, fill ϕ^c by the resulting LDA discriminant coefficients.
5. Create the child $I^c=(\tau^c;\phi^c)$.

Before inserting the child into the next population, I^c undergoes a mutation operation. In Section 5.1, we will show a

[1] For simplicity reason, we use I (chromosome) instead of τ (gene subset part of I) in the fitness function even if it is the gene subset τ that is effectively evaluated.

comparative study of this LDA-based crossover operator with the well-known one-point and uniform crossovers.

3.5. LDA-based mutations

In a conventional GA, the purpose of mutation is to introduce new genetic materials for diversifying the population by making local changes in a given chromosome. For binary coded GAs, this is typically realized by flipping the value of some bits ($1 \rightarrow 0$, or $0 \rightarrow 1$). In our case, mutation is used for dimension reduction; each application of mutation eliminates a single gene ($1 \rightarrow 0$). To determine which gene is discarded, two criteria are used, leading to two mutation operators.

- *Mutation using discriminant coefficient* (*M1*): Given a chromosome $I = (\tau; \phi)$, we identify the smallest LDA discriminant coefficient in ϕ and remove the corresponding gene (this is the least informative genes among the current candidate gene subset τ).
- *Mutation by discriminant coefficient and frequency* (*M2*): This mutation operator relies on a frequency information of each selected gene. More precisely, a frequency counter is used to count the number of times a selected gene is classified (according to the LDA classifier) as the least informative gene within a gene subset. Based on this information, we remove the gene that has the highest counter, in other words, the gene that is frequently considered as a poor predictor by the classifier.

3.6. Chromosome regeneration

As explained above, our crossover and mutation operators remove progressively irrelevant genes at each generation. As such, when the search progresses, a chromosome could become "empty", i.e. all the genes of the chromosome are removed. When this happens, a simple regeneration strategy is applied to replace the "empty" chromosome by a new chromosome generated according to procedure explained in Section 3.1.

4. Experiments on microarray datasets

In this section, we present experimental results provided by our LDA-GA algorithm. We first recall the characteristics of the publicly available microarray datasets used in our experiments and then define precisely the experimental setting in order to give an unbiased evaluation of the quality of our solutions. Computational results are also contrasted with several other gene selection methods proposed recently in the literature.

4.1. Microarray gene expression datasets

To assess the performance of our LDA-based genetic algorithm, we decide to perform our experiments on seven well-known public datasets that provide binary classification problems: leukemia, colon cancer, lung cancer, prostate cancer, central nervous system embryonal tumor (CNS), ovarian cancer and lymphoma (DLBCL). A summary about the datasets is provided in Table 1, where we recall the number of genes, the number of samples and the first publication that has presented an analysis of this dataset.

4.2. Pre-selection with t-statistic

Given the very high dimension of the datasets, embedded approaches usually begin with a pre-selection step to retain a reduced number of p genes to limit the computation time of the learning step. In our case, we employ a ranking-based t-statistic filter to pre-select 100–150 genes. The t-statistic filter is a conventional tool for the selection of differentially expressed genes.

In our context, the pre-selection step is applied to the training samples of each fold of a 10-fold cross-validation process (see next section and Fig. 1).

In addition to t-statistic, we also experimented two other filters (BSS/WSS and Wilcoxon) for pre-selection. We observed that the classification accuracy is often slightly better with t-statistic than with these other filters when the number of the pre-selected genes is high ($p \geq 100$). Moreover, we assessed the possibility of using LDA for pre-selection. We noticed that with our 10-fold cross-validation process, this is too time-consuming to be an interesting option (due to the matrix operations explained in Section 2.2).

Table 1
Summary of datasets used for experimentation.

Dataset	Genes	Samples	References
Leukemia	7129	72	Golub et al. [16]
Colon	2000	62	Alon et al. [3]
Lung	12 533	181	Gordon et al. [17]
Prostate	12 600	109	Singh et al. [41]
CNS	7129	60	Pomeroy et al. [37]
Ovarian	15 154	253	Petricoin et al. [36]
DLBCL	4026	47	Alizadeh et al. [1]

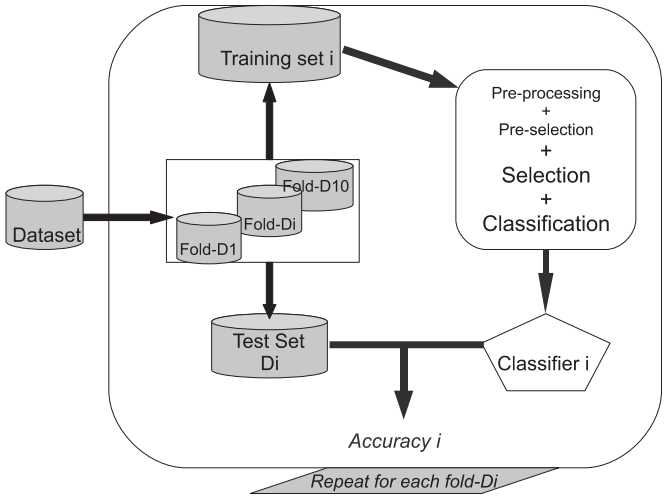

Fig. 1. Cross-validation schema for gene selection and classification.

4.3. Experimental setting

The quality of a selected gene subset is assessed by its capability to lead to an efficient classifier. The importance of a rigorous estimation of classifier accuracy is a well-known issue in machine learning. The evaluation of a classification model built on a training set may be performed on an independent test set. Such a protocol is meaningful when a great number (at least several 100s) of labeled examples are available, so that the initial dataset can be split into a training dataset and a test set of reasonable sizes.

When the labeled samples are scarce, which is the case for Microarray data, the estimation of prediction accuracy can be realized via cross-validation. In the k-fold cross-validation protocol,

the initial dataset D is split into k subsets of approximately the same size $D_1,\ldots,D_k$. The learning algorithm is applied k times to build k classifiers: in step i, the data subset D_i is left out as a test set, the classifier is induced from the training dataset $D-D_i$ and its accuracy Acc_i is estimated on D_i. The accuracy estimate computed by k-fold cross-validation is then the mean of the Acc_i, for $1 \leq i \leq k$. When the number of folds (iterations) is equal to the number of initial samples, the so-called leave-one-out-cross-validation (LOOCV) protocol provides an unbiased estimate of the generalization accuracy. However, this estimate has a high variance [13] and requires important computational efforts, since a classifier is trained in each iteration. Empirical studies [21,8] have shown that 10-fold cross-validation is a good choice to obtain an almost unbiased estimate, with small variance and reasonable computational time. It is also recommended to use stratified cross-validation where each fold contains approximately the same proportions of classes as the initial dataset.

As pointed out in [4,40,22,2], it is important to include gene selection into the cross-validation schema in order to avoid the *selection bias* that overestimates predictive accuracy. Selection bias occurs when the accuracy of a model is assessed on samples that play a role in the construction of the model. Therefore, selecting a subset of genes on the entire dataset and then performing cross-validation to estimate a classifier model is a biased protocol. In a correct experiment design, the dataset must be split before gene selection is achieved: each step of cross-validation performs gene selection and classification. Moreover, since we integrate a pre-selection step (see Section 4.2), only the training samples of each fold are used to calculate the t-statistic score of each gene. The whole experimental process is described in Fig. 1.

Our experiments are conducted according to this schema. LDA-GA is applied on the training set in order to select a relevant gene subset and to obtain a classifier. We have seen that our fitness function relies on two criteria, the accuracy f_1 and the number of genes f_2, that are aggregated in a unique value by the formula $f(I)=\alpha f_1(I)+(1-\alpha)f_2(I)$. In this experiment, we focus on the accuracy, so the fitness function is defined with $\alpha=0.75$. In Section 5, we present further experiments with different values of α. Therefore, to have clear and consistent notations, we name $F_{0.75}$ the fitness function used in this current experiment.

Because of the stochastic nature of our LDA-GA algorithm, we run 10 executions of this schema and we retain the best solution found during these 10 executions.

We have explained in Section 3 that our LDA-GA can apply two kinds of mutation (M 1 and M 2). That is why we report in the following subsection two results for each dataset: $LDA-GA_{M\ 1}/F_{0.75}$ uses mutation M 1 and the fitness function defined by parameter $\alpha=0.75$. $LDA-GA_{M\ 2}/F_{0.75}$ uses mutation M 2 and the same fitness function.

4.4. Computational results

A great number of works study the problem of gene selection and classification of Microarray data. These studies are based on a variety of approaches on the one hand, and employ often different pre-processing techniques, evaluation schema, and experiment protocols on the other hand. Due to these variations, a fair and exhaustive comparison among any methods becomes a very challenging and even impossible task.

The main purpose of this section is to show the range of performance that can be achieved by our LDA-GA on the seven tested datasets, under the strict and unbiased experimental protocol with 10-fold cross-validation described in Section 4.3. In order to give some idea of the performance of our algorithm relative to other state-of-the-art approaches and only for this purpose, we selected 15 recent gene selection algorithms (published since 2004) which represent a variety of approaches (Bayesian learning, boostrapping, ensemble machine learning, ensemble of neural networks, combined SVM, minimum redundancy…). All the methods use a process of cross-validation, notice, however, that sometimes the papers do not explain precisely how the experimentation is conducted.

Table 2 presents the results obtained by our LDA-GA together with results reported in these 15 references on the seven datasets. An entry with the symbol (–) means that the paper does not treat the corresponding dataset. It should be clear that any numerical comparison must be interpreted with caution.

From Table 2, one observes that LDA-GA (two last lines) is able to reach a high classification accuracy of at least 96% for all the seven datasets (100% for four of them) with at most 24 genes. Now, let us give more comments on the results of LDA-GA.

For the Leukemia dataset, $LDA-GA_{M\ 2}$ obtains a perfect classification with three or five genes. To complete the information given in Table 2, we must precise that LDA-GA also found other gene subsets (with 5–10 genes) achieving a perfect cross-validation classification. More precisely, one of these subsets contains the genes in positions 3, 12, 63, 72, and 81 ranked by t-statistic filter. Another gene subset that achieves a perfect classification contains the genes ranked in positions: 1, 2, 19, 72 and 81. These solutions show the limit of filter-based approaches that can miss easily relevant genes because they typically retain a small number of top-ranking genes (say 30) for classification. At the same time, these results show the usefulness

Table 2
Results of our LDA-based GA equipped with mutation operators M 1 and M 2 (two last lines).

Author	Leuk.	Colon	Lung	Prostate	CNS	Ovar.	DLBCL
Ye et al. [47]	97.5	85.0	–	92.5	–	–	–
Liu et al. [28]	**100**	91.9	**100**	97.0	–	99.2	98
	30	30	30	30	–	75	30
Tan and Gilbert [42]	91.1	95.1	93.2	73.5	88.3	–	–
Ding and Peng [10]	**100**	93.5	97.2	–	–	–	–
Cho and Won [9]	95.9	87.7	–	–	–	–	93.0
	25	25					25
Yang et al. [45]	73.2	84.8	–	86.88	–	–	–
Peng et al. [35]	98.6	87.0	**100**	–	–	–	–
	5	4	3				
Wang et al. [43]	95.8	**100**	–	–	–	–	95.6
	20	20					20
Huerta et al. [6]	**100**	91.4	–	–	–	–	–
Pang et al. [32]	94.1	83.8	91.2	–	65.0	98.8	–
	35	23	34		46	26	
Li et al. [25]	97.1	83.5	–	91.7	68.5	99.9	93.0
	20	20		20	20	20	20
Zhang et al. [49]	**100**	90.3 30	**100**	95.2	80	–	92.2
	30	30	30	30	30		30
Yue et al. [48]	83.8	85.4	–	–	–	–	–
	100	100					
Hernandez et al. [19]	91.5	84.6	–	–	–	–	–
	3	7					
Li et al. [26]	**100**	93.6	–	–	–	–	–
	4	15					
$LDA-GA_{M\ 1}/F_{0.75}$	**100**	91.9	99.3	96.0	86.6	**100**	**100**
	3	4	10	53	34	18	4
$LDA-GA_{M\ 2}/F_{0.75}$	**100**	98.3	99.3	96.0	**100**	98.8	**100**
	5	14	7	8	24	17	3

These results are obtained by strictly following the unbiased experimental protocol defined in Section 4.3. For indicative purpose, are equally shown the results of 15 other recent methods which are based on a cross-validation process. Each cell contains the classification accuracy and the number of genes when this is available. Any numerical comparison must be interpreted with caution.

of exploring larger gene subsets including those that are not top-ranked.

For the Colon tumor dataset which is known to be difficult for many methods, LDA-GA obtains a good performance: 98.3% with 14 genes and 91.9% with only four genes. It has been pointed out in [14] that six samples are often misclassified. These six samples are the tumor tissues T30, T33 and T36 and the normal tissues: N8, N34 and N36 (more details in [3]). If we exclude these six samples from the dataset, our method achieves a 100% accuracy. So our results are perfectly consistent with the observations about this dataset and this confirms the quality of our approach.

For the Lung dataset, our algorithm achieves an almost perfect prediction accuracy (99.3%) with 7 and 10 genes.

For the Prostate cancer we obtain a high performance with a very reduced number of genes: 96% with eight genes.

For the CNS dataset, $LDA-GA_{M\ 2}$ reaches a perfect discrimination with 24 genes.

For the Ovarian cancer dataset, we obtain a perfect classification with 18 genes. Notice that a perfect rate is reported in [30] with 50 genes. However, the dataset used in [30] (30 cancerous and 24 normal samples, 1536 genes) is different from the Ovarian cancer dataset described in Table 1 (91 normal and 162 cancerous samples, 15,154 genes).

The most remarkable results for our approach concern the DLBCL dataset. We obtain a perfect prediction with only three or four genes while the previously reported results are below 98% with at least 20 genes.

These results show that LDA-GA performs globally very well on these different datasets.

Finally, let us mention that the computing time of LDA-GA (programmed in Matlab) for processing a dataset varies from several minutes to one hour on a PC with 3.4 GHz CPU and 2G RAM.

4.5. Discussion

As mentioned previously, the proposed approach is able to explore many gene combinations and return different (small) gene subsets with a high prediction rate. This feature is particularly interesting from a practical point of view since these multiple results may constitute valuable sources for further biological studies. For example, based on these gene subsets, one can easily identify frequently selected genes so that a particular attention can be put on them. According to a preliminary examination that we carried out on Leukemia, Colon cancer and DLBCL, we observe that the algorithm is able to find repeatively a number of genes which are known as biomarkers for the concerned cancer. Moreover, for those genes which are repeatively selected, yet not reported in any biological study, they may constitute interesting candidates for more focused biological investigations.

5. Additional studies of the proposed algorithm

One originality of our approach is the design of a dedicated crossover operator that is informed by the coefficients of a LDA classifier. The first part of this section studies the impact of this specific crossover operator on the evolutionary process. In a second part, we also provide experimental results that show the influence of some other parameters or components of the genetic algorithm.

5.1. LDA-based crossover vs. random crossovers

For simplicity reason, we use AND to designate our dedicated crossover operator because its main operation is to take the intersection of the parent genes in order to transmit relevant information to the offspring. We believe that this LDA-based crossover operator is one of the driving forces of our genetic algorithm. To confirm this, we compare the performance of AND against two random crossover operators (single point and uniform crossovers). These standard operators combine parts of two parents in a stochastic way without the help of problem specific knowledge. The performance of each crossover is based on a running profile of the best fitness of the population as a function of the number of generations.

In order to enable a fair comparison, all the crossover operators are tested based on the LDA-GA algorithm under the same experimental conditions on three datasets (Leukemia, Colon cancer, Lung cancer). More specifically, the following parameters were used in this experiment: (a) population size $|P|=50$, (b) maximal number of generations is fixed at 250, (c) individual length (number of pre-selected genes) $p=150$, (d) weighting coefficient of the fitness function $\alpha=0.5$, (e) mutation probability for random mutation is fixed at 0.01, and (f) single point and uniform crossover probability is fixed at 0.875. The general settings for our LDA based crossover operator are explained in Section 3. Fig. 2 shows the comparative results of the same GA with those different crossover operators. Given the stochastic nature of the GA, each curve represents the evolution of the best fitnesses averaged over 10 independent runs.

From this figure, it is observed that our informed operator (AND) allows the algorithm to obtain consistently better solutions. More specifically, on the Colon dataset, a fitness value of 0.80 is rapidly reached by the best individual within 20 generations with the AND operator. With the two other crossover operators, this fitness value is reached only after 100 generations. A weak dominance of AND can still be observed for the Leukemia dataset for which the curves of AND and Uniform remain close until 120 generations. This is not so surprising because previous studies have shown that the discrimination is easy for this dataset and consequently even a random operator can obtain good results. For Lung dataset, the profile of AND is globally better than those of the competing crossovers, leading to the highest fitness value from 120 generations. In conclusion, this experiment tends to confirm the superiority of the informed crossover operator in guiding the GA in its exploration of the search space. The performance of our LDA-GA can thus be partially attributed to the dedicated crossover operator.

5.2. Influence of algorithm parameters

The fitness function defined in Section 3.3 combines two criteria, the classifier accuracy and the number of the selected genes. These criteria are aggregated in a sole value ($f \in [0,1]$) by a weighted sum where the parameter α determines the importance of each criterion (see Eq. (6)). Varying the value of α would allow the search to focus on one criterion or another. We study the effect of this parameter α in the following experiments.

We carry out an experiment where the fitness function uses $\alpha=0.25$ (call it $LDA-GA/F_{0.25}$); therefore the algorithm puts more emphasis on the number of selected genes. In another experiment ($LDA-GA/F_{0.75}$), f is based on $\alpha=0.75$ and the classification accuracy takes more importance (this last setting was used in Section 4). Since we run LGA-GA with either M 1 or M 2 mutation operator, this gives four algorithm combinations.

Additionally, we use this opportunity to check the influence of the population size by using three different sizes (30, 50 and 100) with each of the four algorithm combinations. The results of these experiments are presented in Table 3.

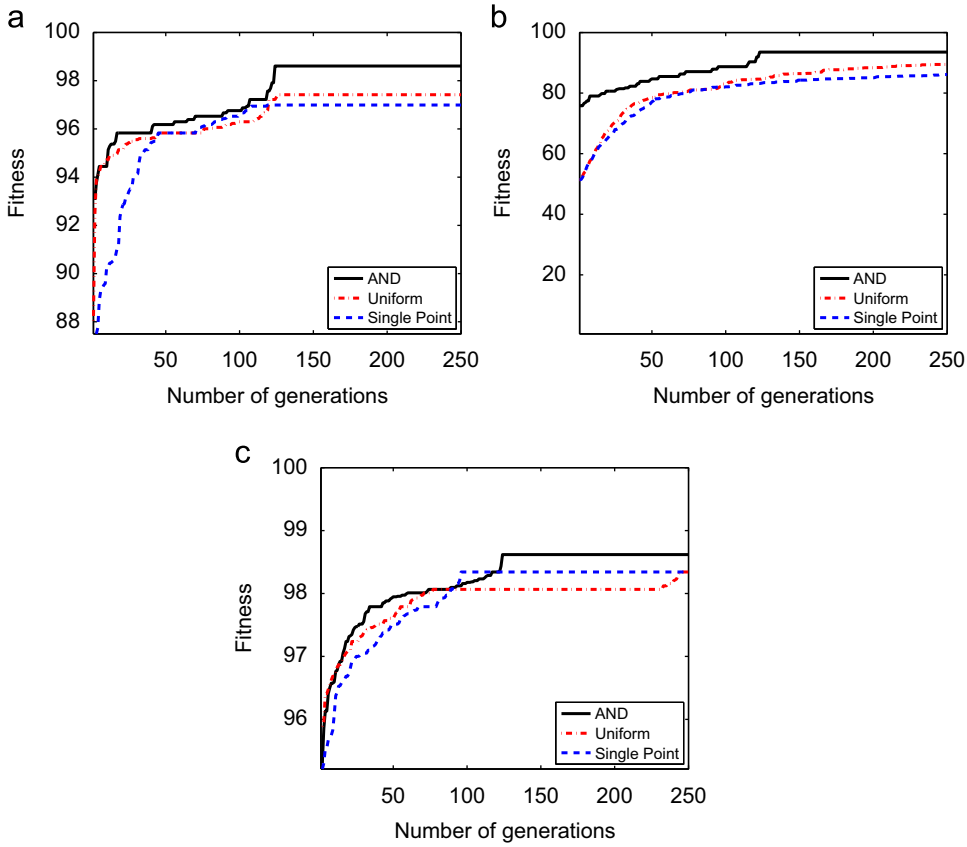

Fig. 2. Evolution profile of the best fitness of the LDA-GA equipped with the dedicated crossover operator (AND) in comparison with the profile obtained with uniform and single-point crossover operators. The curves represent the best fitness averaged over 10 independent runs: (a) leukemia dataset, (b) colon cancer dataset, and (c) lung cancer dataset.

Table 3
Influence of α used in the fitness function with $\alpha = 0.25$ and $\alpha = 0.75$.

	Leuk.	Colon	Lung	Prostate	CNS	Ovar.	DLBL
Pop. size = 30							
$LDA-GA_{M\ 1}/F_{0.25}$	93.0(1)	82.2(1)	95.9(1)	92.1(1)	73.3(1)	84.1(1)	80.8(1)
$LDA-GA_{M\ 2}/F_{0.25}$	94.4(1)	82.2(1)	95.9(1)	91.1(1)	76.6(1)	84.5(1)	80.8(1)
$LDA-GA_{M\ 1}/F_{0.75}$	(**100**)4	(98.3)10	(99.3)8	(97.0)10	(98.3)12	(**100**)28	(**100**)6
$LDA-GA_{M\ 2}/F_{0.75}$	(**100**)5	(98.3)19	(99.3)7	(98.0)35	(98.0)35	(**100**)25	(**100**)5
Pop. size = 50							
$LDA-GA_{M\ 1}/F_{0.25}$	94.4(1)	64.5(1)	95.3(1)	92.1(1)	73.3(1)	79.8(1)	91.4(1)
$LDA-GA_{M\ 2}/F_{0.25}$	94.4(1)	64.5(1)	93.2(1)	92.1(1)	73.3(1)	80.6(1)	87.2(1)
$LDA-GA_{M\ 1}/F_{0.75}$	(**100**)13	(91.9)14	(99.3)9	(94.1)18	(88.3)44	(98.8)35	(**100**)6
$LDA-GA_{M\ 2}/F_{0.75}$	(**100**)28	(98.3)10	(99.3)10	(95.0)38	(88.3)44	(**100**)34	(**100**)6
Pop. size = 100							
$LDA-GA_{M\ 1}/F_{0.25}$	83.3(2)	83.8(2)	93.9(2)	75.4(1)	60.0(1)	88.5(1)	70.2(2)
$LDA-GA_{M\ 2}/F_{0.25}$	94.4(1)	83.8(2)	95.3(1)	88.2(1)	66.6(1)	87.7(1)	72.3(1)
$LDA-GA_{M\ 1}/F_{0.75}$	(**100**)3	(91.9)4	(99.3)10	(96.0)53	(86.6)34	(**100**)18	(**100**)4
$LDA-GA_{M\ 2}/F_{0.75}$	(**100**)5	(98.3)14	(99.3)7	(96.0)8	(**100**)24	(98.8)17	(**100**)3

In each cell, the first number is the classification accuracy and the second one is the number of selected genes. The brackets indicate which of these two criteria is dominant in the fitness function. Results are reported on the LDA-GA equipped with mutation operators M 1 and M 2, run with three population sizes.

The results of this table show that α does have a clear influence on the search direction of the genetic algorithm. A smaller value of α allows the search to obtain smaller sized gene subsets at the price of lower classification accuracy and vice versa.

One also observes that the LDA-GA is able to achieve very good performance even with a small population of 30 individuals. However, the population size of 100 provides a slight improvement in the sense that it offers a better compromise between two

objectives: good classification accuracy and small number of genes. This is particularly true for the algorithms with the fitness function named $F_{0.75}$ since we obtain a perfect classification of 100% in six cases with less than 24 genes.

5.3. Influence of the pre-processing step

The search space of our LDA-GA is delimited by a first step which pre-selects a limited number of genes (100–150 in this paper) with the *t*-statistic filter criterion. One may wonder whether changing the filtering criterion and the number of selected genes affects the performance of the approach. In [7], an exhaustive study is presented concerning the influence of data pre-processing and filtering criteria on the classification performance. Three filtering criteria, BSS/WSS, *t*-statistic and Wilcoxon test were compared, and the results did not show strong dominance of one criterion with respect to the others, although *t*-statistic led to slightly better results on some datasets like Colon and CNS. However, the fuzzy pre-processing for data normalization and redundancy reduction presented in [7] does show a positive influence on the classification performance whatever the filtering criterion that is applied after.

Finally, even if we used in this paper 100 and 150 pre-selected genes as the input of the LDA-GA, the proposed LDA-GA can be applied with a larger input. In this case, the algorithm will explore more gene combinations at the price of more computing resources.

6. Conclusions and discussion

In this paper, we have introduced a new embedded approach to gene subset selection for cancer classification of microarray data. The proposed approach begins with a (*t*-statistic) filter that pre-selects a first set of genes (100 or 150 in this paper). To further explore the combinations of these genes, we rely on a hybrid genetic algorithm combined with Fisher's Linear Discriminant Analysis. In this LDA-GA, LDA is used not only to assess the fitness of a candidate gene subset, but also to inform the crossover and mutation operators. This GA and LDA hybridization makes the genetic search highly effective for identifying small and informative gene subsets.

In addition, the bi-criteria fitness function provides a flexible way for the LDA-GA to explore the gene subset space either for the minimization of the selected genes or for the maximization of the prediction accuracy.

We have extensively evaluated our LDA-based GA approach on seven public datasets (leukemia, colon, DLBCL, lung, prostate, CNS and ovarian) using a rigorous 10-fold cross-validation process. Computational results on these datasets show that under the unbiased experimental protocol, the proposed algorithm is able to reach high prediction accuracies (96% to 100%) with a small number of selected genes (3–24).

The proposed approach has another practically useful feature for biological analysis. In fact, instead of producing a single solution (gene subset), our approach can easily and naturally provide multiple non-dominated solutions that constitute valuable candidates for further biological investigations.

Finally, it should be noticed that the approach exposed in this paper is general in the sense that the LDA classifier can be replaced by other linear classifiers. Indeed, a linear classifier naturally provides ranking or discrimination information about the genes that can be favorably used in the design of search operators of a genetic algorithm.

Acknowledgments

We are grateful for comments by the referees that have improved the paper. This work is partially supported by the French Biogenouest and the Bioinformatics Program of the Region "Pays de La Loire". The first author of the paper is supported by a CoSNET research scholarship.

References

[1] A. Alizadeh, M.B. Eisen, et al., Distinct types of diffuse large (b)—cell lymphoma identified by gene expression profiling, Nature 403 (February) (2000) 503–511.
[2] D.B. Allison, X. Cui, G.P. Page, M. Sabripour, Microarray data analysis: from disarray to consolidation and consensus, Nature Reviews Genetics 7 (1) (2006) 55–65.
[3] U. Alon, N. Barkai, et al., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. USA 96 (1999) 6745–6750.
[4] C. Ambroise, G. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proc. Natl. Acad. Sci. USA 99 (10) (2002) 6562–6566.
[5] A. Ben-Dor, L. Bruhn, et al., Tissue classification with gene expression profiles, Journal of Computational Biology 7 (3–4) (2000) 559–583.
[6] E. Bonilla Huerta, B. Duval, J.K. Hao, A hybrid GA/SVM approach for gene selection and classification of microarray data, in: F. Rothlauf et al. (Ed.), Applications of Evolutionary Computing, EvoWorkshops 2006, Lecture Notes in Computer Science, vol. 3907, Springer, 2006, pp. 34–44.
[7] E. Bonilla Huerta, B. Duval, J.K. Hao, Fuzzy logic for elimination of redundant information of microarray data, Genomics, Proteomics and Bioinformatics 6 (2) (June 2008) 61–73.
[8] U. Braga-Neto, E.R. Dougherty, Is cross-validation valid for small-sample microarray classification? Bioinformatics 20 (3) (2004) 374–380.
[9] S.-B. Cho, H.-H. Won, Cancer classification using ensemble of neural networks with multiple significant gene subsets, Applied Intelligence 26 (3) (2007) 243–250.
[10] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, Bioinformatics and Computational Biology 3 (2) (2005) 185–206.
[12] S. Dudoit, J. Fridlyand, T. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, Journal of the American Statistical Association 97 (2002) 77–87.
[13] B. Efron, Estimating the error rate of a prediction rule: improvement on cross-validation, Journal of the American Statistical Association 78 (382) (1983) 316–331.
[14] T. Furey, N. Cristianini, et al., Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (10) (2000) 906–914.
[16] T. Golub, D. Slonim, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, E. Lander, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[17] G.J. Gordon, R.V. Jensen, S. Ramaswamy, W.G. Richards, D.J. Sugarbaker, R. Bueno, et al., Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma, Cancer Research 17 (62) (2002) 4963–4967.
[19] J.C. Hernandez Hernandez, B. Duval, J.K. Hao, A genetic embedded approach for gene selection and classification of microarray data, in: E. Marchiori, J.H. Moore, J.C. Rajapakse (Eds.), EvoBIO'07, Lecture Notes in Computer Science, vol. 4447, Springer, 2007, pp. 90–101.
[20] K.-J. Kim, S.-B. Cho, Prediction of colon cancer using an evolutionary neural network, Neurocomputing 61 (2004) 361–379.
[21] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection, in: S. Mateo (Eds.), Proceedings of the 14th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, CA, 1995, pp. 1137–1143.
[22] S. Lee, Mistakes in validating the accuracy of a prediction classifier in high-dimensional but small-sample microarray data, Statistical Methods in Medical Research 17 (2008) 635–642.
[23] L. Li, C.R. Weinberg, T.A. Darden, L.G. Pedersen, Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method, Bioinformatics 17 (12) (2001) 1131–1142.
[24] L. Li, W. Jiang, X. Li, K.L. Moser, Z. Guo, L. Du, Q. Wang, E.J. Topol, Q. Wang, S. Rao, A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset, Genomics 85 (2005) 16–23.
[25] G.Z. Li, X.Q. Zeng, J.Y. Yang, M.Q. Yang, Partial least squares based dimension reduction with gene selection for tumor classification, in: Proceedings of IEEE Seventh International Symposium on Bioinformatics and Bioengineering, 2007, pp. 1439–1444.
[26] S. Li, X. Wu, X. Hu, Gene selection using genetic algorithm and support vectors machines, Soft Computing 12 (7) (2008) 693–698.

[28] B. Liu, Q. Cui, T. Jiang, S. Ma, A combinational feature selection and ensemble neural network method for classification of gene expression data, BMC Bioinformatics 5:136 (138) (2004) 1–12.
[29] J.J. Liu, G. Cutler, W. Li, Z. Pan, S. Peng, T. Hoey, L. Chen, X.B. Ling, Multiclass cancer classification and biomarker discovery using GA-based algorithms, Bioinformatics 21 (11) (2005) 2691–2697.
[30] E. Marchiori, M. Sebag, Bayesian learning with local support vector machines for cancer classification with gene expression data, EvoWorkshops, Lecture Notes in Computer Science, vol. 3449, Springer, 2005, pp. 74–83.
[31] C.H. Ooi, P. Tan, Genetic algorithms applied to multi-class prediction for the analysis of gene expression data, Bioinformatics 19 (1) (2003) 37–44.
[32] S. Pang, I. Havukkala, Y. Hu, N. Kasabov, Classification consistency analysis for bootstrapping gene selection, Neural Computing and Applications 16 (2007) 527–539.
[33] H. Park, C. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition 41 (3) (2008) 1083–1097.
[34] S. Peng, Q. Xu, X.B. Ling, X. Peng, W. Du, L. Chen, Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines, FEBS Letters 555 (2) (2003) 358–362.
[35] Y. Peng, W. Li, Y. Liu, A hybrid approach for biomarker discovery from microarray gene expression data, Cancer Informatics 2 (2006) 301–311.
[36] E.F. Petricoin, A.M. Ardekani, B.A. Hitt, P.J. Levine, V.A. Fusaro, S.M. Steinberg, G.B. Mills, C. Simone, D.A. Fishman, E.C. Kohn, L.A. Liotta, Use of proteomic patterns in serum to identify ovarian cancer, Lancet 359 (2002) 572–577.
[37] S.L. Pomeroy, P. Tamayo, et al., Prediction of central nervous system embryonal tumour outcome based on gene expression, Nature 415 (2002) 436–442.
[40] R. Simon, M.D. Radmacher, K. Dobbin, L.M. McShane, Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification, Journal of the National Cancer Institute 95 (1) (January 2003) 14–18.
[41] D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, A. Renshaw, A. D'Amico, J. Richie, Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209.
[42] A.C. Tan, D. Gilbert, Ensemble machine learning on gene expression data for cancer classification, Applied Bioinformatics 2 (2) (2003) 75–83.
[43] Z. Wang, V. Palade, Y. Xu, Neuro-fuzzy ensemble approach for microarray cancer gene expression data analysis, in: Proceedings of the Evolving Fuzzy Systems, 2006, pp. 241–246.
[45] W.-H. Yang, D.-Q. Dai, H. Yan, Generalized discriminant analysis for tumor classification with gene expression data, Machine Learning and Cybernetics 1 (2006) 4322–4327.
[46] J. Ye, Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems, Journal of Machine Learning Research 6 (2005) 483–502.
[47] J. Ye, T. Li, T. Xiong, R. Janardan, Using uncorrelated discriminant analysis for tissue classification with gene expression data, IEEE/ACM Transactions on Computational Biology and Bioinformatics 1 (4) (2004) 181–190.
[48] F. Yue, K. Wang, W. Zuo, Informative gene selection and tumor classification by null space LDA for microarray data, ESCAPE'07, Lecture Notes in Computer Science, vol. 4614, Springer, 2007, pp. 435–446.
[49] L. Zhang, Z. Li, H. Chen, An effective gene selection method based on relevance analysis and discernibility matrix, in: PAKDD, Lecture Notes in Computer Science, vol. 4426, 2007, pp. 1088–1095.
[50] Z. Zhu, Y.S. Ong, M. Dash, Markov blanket-embedded genetic algorithm for gene selection, Pattern Recognition 40 (11) (2007) 3236–3248.

Edmundo Bonilla Huerta is an Assistant Professor at the Instituto Tecnologico Apizaco, Tlaxcala, Mexico. His research lies in optimization and machine learning techniques for microarray data analysis.

Béatrice Duval is an Associate Professor in Computer Science at the University of Angers (France). Her main research field concerns data mining and machine learning. For some years, she has been working on applications in bioinformatics with hybrid methods combining machine learning techniques and meta-heuristics.

Jin-Kao Hao holds a full Professor position in the Computer Science Department of the University of Angers (France) and is currently the Director of the LERIA Laboratory. His research lies in the design of effective heuristic and metaheuristic algorithms for solving large-scale combinatorial search problems. He is interested in various application areas including bioinformatics, telecommunication networks and transportation. He has co-authored more than 100 publications in international journals, book chapters and conference proceedings.