Applied Soft Computing 38 (2016) 922–932

Contents lists available at ScienceDirect

Applied Soft Computing

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

Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments

Javier Apolloni [a,*], Guillermo Leguizamón [a], Enrique Alba [b]

[a] LIDIC – Departamento de Informática, Universidad Nacional de San Luis, Ejército de Los Andes 950, 5700 San Luis, Argentina
[b] Lenguajes y Ciencias de la Computación, Universidad de Málaga, Campus de Teatinos s/n, 29071 Málaga, Spain

ARTICLE INFO

Article history:
Received 6 June 2015
Received in revised form
12 September 2015
Accepted 18 October 2015
Available online 27 October 2015

Keywords:
Hybrid Feature Selection
Binary Differential Evolution
High-dimensional data set
Microarray

ABSTRACT

Microarray experiments generally deal with complex and high-dimensional samples, and in addition, the number of samples is much smaller than their dimensions. Both issues can be alleviated by using a feature selection (FS) method. In this paper two new, simple, and efficient hybrid FS algorithms, called respectively BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$, are presented. Both algorithms combine a wrapper FS method based on a Binary Differential Evolution (BDE) algorithm with a rank-based filter FS method. Besides, they generate the initial population with solutions involving only a small number of features. Some initial solutions are built considering only the most relevant features regarding the filter method, and the remaining ones include only random features (to promote diversity). In the BDE-$\mathcal{X}_{Rank_f}$, a new fitness function, in which the score value of a solution is influenced by the frequency of the features in the current population, is incorporated in the algorithm. The robustness of BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$ is shown by using four Machine Learning (ML) algorithms (*NB*, *SVM*, *C4.5*, and *kNN*). Six high-dimensional well-known data sets of microarray experiments are used to carry out an extensive experimental study based on statistical tests. This experimental analysis shows the robustness as well as the ability of both proposals to obtain highly accurate solutions at the earlier stages of BDE evolutionary process. Finally, BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$ are also compared against the results of nine state-of-the-art algorithms to highlight its competitiveness and the ability to successfully reduce the original feature set size by more than 99%.

1. Introduction

Microarrays are tools which are capable of reporting, simultaneously and in a single testing, thousands of gene expression levels. A microarray experiment can be used to analyze samples of cells under different clinical conditions. The information obtained helps to determine the functionality of the genes, and/or helps in understanding biological processes and the effect of certain treatments. However, all of these activities require analyzing and classifying the samples [1].

In a microarray, the samples are complex, highly dimensional and prone to contamination with noise. Therefore, the classification requires special techniques such as *Machine Learning* (ML) algorithms. These techniques construct a *classifier* using one of at least two main mechanisms: *unsupervised* and *supervised* learning [2]. In this paper, the second mechanism is employed because all of the samples have an assigned class value label.

Usually, the quantity of samples in microarray experiments is several orders of magnitude lower than the number of genes in the samples. Considering the additional complexity, the ML algorithms are prone to suffer the *curse of dimensionality*. This anomaly affects both the reliability and the predictive ability of the classifier. To alleviate these effects, the reduction of the number of features by using a feature selection (FS) method must be employed during the classifier's construction.

FS algorithms are designed to identify and select the useful features contained in the samples [3]. To find the best features, FS algorithms must interact with ML techniques. According to the type of interaction, FS algorithms can be grouped into two large classes: *filter methods* and *wrapper methods*. On the one hand, the filter methods select the best features from a rank of the whole set of features. Despite the simplicity, speed, and computational efficiency, they are unable to detect indirect relationships between features. On the other hand, the wrapper methods employ the information of the classifier to find the best feature subset. Although, they discover indirect relationships between features, they usually perform computationally expensive searches on the feature space [4].

Regarding the wrapper methods, an exhaustive search is the simplest technique. However, it is prohibitive in terms of computational cost. To the contrary, the methods based on stochastic searches have a lower computational cost and maintain simplicity. For example, evolutionary algorithms (EAs) find the best solution by exploring the search space with a population of solutions. EAs have proved to be successful in solving FS problems, e.g., Genetic Algorithms [5,6], Ant Colony Optimization [7,8], Particle Swarm Optimization [9,10].

Differential Evolution (DE) is one of the simplest and most effective EAs used to solve high-dimensional optimization problems [11]. In microarray problems, DE could take advantage of the usually small number of the most informative features when initializing the population. Considering all the features arranged into a rank, the initial solutions could include only the most relevant features. However, this initialization of the population could cause a premature lack of diversity, and hence DE could converge to a local optimum. To alleviate this pressure, some initial solutions should be randomly generated.

* Corresponding author. Tel.: +54 0266 4520300x2119.
E-mail address: javierma@unsl.edu.ar (J. Apolloni).

http://dx.doi.org/10.1016/j.asoc.2015.10.037
1568-4946/

The purpose (and contribution) of this paper is to present two simple and thus computationally efficient algorithms to solve the FS problem in microarray experiments. The proposed algorithms integrate both filter and wrapper methods to obtain two *hybrid* FS methods. The experimental study was carried out using six high-dimensional well-known microarrays in the public domain. Four simple and efficient ML algorithms (*Naive Bayesian*, *Support Vector Machine*, *C4.5*, and *k-Nearest Neighborhood*) were used to evaluate the robustness, efficiency, and accuracy of the hybrid FS technique. It is worth mentioning that the simplicity of the four ML algorithms employed prevents an additional increase in the complexity of the proposed technique. The conclusions drawn were validated with several statistical tests. In addition, this paper also introduces a new and interesting fitness function. More precisely, the score of a solution is modified according to the frequency with which of their features appear inside the population.

The rest of this paper is organized as follows. Section 2 briefly summarizes the recent developments in hybrid FS methods, including the use of DE as an efficient feature selection method. A detailed description of preliminary concepts is presented in Section 3. The hybrid wrapper-filter FS algorithm is outlined in Section 4. An extensive experimental study and the introduction of a new fitness function is shown in Section 5. Finally, the conclusions and future work are explained in Section 6.

2. Previous work

Further to briefly reviewing the DE-based FS technique, this section presents a summary of the main approaches related to hybrid FS methods found in the literature. Indeed, some authors have already studied the advantages of combining a feature ranking technique and a search algorithm to create a hybrid FS method. For instance, Xie and Wang [12] and Peng et al. [13] included a feature ranking in a sequential forward search method. The former applies the F-score measure to rank the features. The latter adds a random sampling method to choose features from the ranking. In [14], Lee et al. incorporated a Fisher criterion based ranking inside a binary search.

Other authors have used a search method based on a stochastic technique. Zhang et al. [15] proposed including a ranking based on ReliefF estimation in a Genetic Algorithm. The proposal of Bonilla-Huerta et al. [16] is similar. In this case, the ranking is also employed to reduce the solution search space. In addition, the authors present a novel crossover operator based on Fisher's Linear Discrimination Analysis. Kabir et al. [17] used the information gain of the features to modify an ant's position in the Ant Colony Optimization Algorithm. Undoubtedly, there are several relevant papers that use hybrid FS techniques based on stochastic search methods. However, to the best of the authors' knowledge, there are only a few proposals that use DE as part of a hybrid FS algorithm. Indeed, there have only been a few attempts to employ DE inside a wrapper FS method.

The DE-based approaches can be grouped into two broad categories on the basis of the representation of the solutions. On the one hand, some authors propose representing the solution in terms of a schema of real-valued vectors. He et al. [18] in a short paper, presented the first of these approaches for solving the FS problem. However, a disadvantageous transformation of the solution representation is needed in order to obtain the feature subset. The island-based model of Marinaki and Marinakis [19] has a similar issue. In both cases, the experimental study was conducted on well-known data sets but with very few features. Khushaba et al. [20] investigated the use of DE with respect to one of its more interesting properties: fast convergence. Due to the representation selected, they had to apply a repair method to avoid solutions with replicated features. Al-Ani et al. [21] proposed a novel wheels-based method to deal with real-value vectors. An unfavorable characteristic of this proposal was having to use an upper bound to limit the number of features in the solutions. The experimental analysis of these two approaches was carried out on data sets with a large number of features. On the other hand, an expensive additional transformation is avoided if a binary vector is employed to represent the feature subset. The proposals for solving the FS problem presented by García-Nieto et al. [22] and Satapathy and Naik [23] are based on one such a schema. The experimental analysis of both proposals was performed on well-known data sets with few features. In both studies, the results showed the effectiveness of DE selecting a small number of features. However, the first approach reported a huge computational time.

Regarding the issue of the huge computational cost, Apolloni et al. [24] tackled this drawback using a rank-based method to create the initial population of a binary version of DE. Although the experimental study is simple, the proposal demonstrated remarkable success in selecting a small and highly accurate subset of features at a low computational cost. However, in [24] it was only used SVM as ML algorithm which it is not enough to claim the robustness of the proposal.

3. Background

This section covers the background necessary to better understand the proposal presented in this article. Feature selection is discussed in Section 3.1. A brief review of Differential Evolution is presented in Section 3.2.

3.1. Feature selection

In mathematical terms, a microarray experiment is a set $\mathcal{M}$ of m samples with n gene expression levels, $x_{ij} \in [f_{j_{\min}}, f_{j_{\max}}]$ $(1 \le i \le m, 1 \le j \le n)$. $f_{j_{\min}}$ and $f_{j_{\max}}$ are, respectively, the lower and upper bounds of the expression levels of the jth gene. The samples can be labeled with a class value from the set $\mathcal{C} = \{c_1, \ldots, c_k\}$. Here, $\mathcal{C}$ has two values, i.e., $k = 2$. Supervised ML algorithms attempt to approximate, as closely as possible, the mapping $\phi^* : \mathcal{M} \to \mathcal{C}$ using a *training set* with only a few samples.

A key issue in microarray experiments is the large number of irrelevant and redundant genes. Their elimination should make the process of obtaining the classifier easier. In addition, the classifier could then be simpler and more accurate [3]. For removing features, some FS techniques select a subset $\mathcal{F}^*$ from the original feature set $\mathcal{F} = \{f_1, \ldots, f_n\}$ without modifying the original representation of the features. An advantage of preserving the representation of the features is the low computational cost in the construction and the use of the classifier [25]. Another advantage is that it permits a clearer visualization and understanding of the final results [26]. Undoubtedly, the FS algorithm affects the construction of the classifier. However, the ML algorithm involved does not always affect the FS process. For example, the filter-based FS methods choose the features without taking the performance of the classifier into consideration. The wrapper-based FS methods, in contrast, are guided by the classifier.

Filter methods construct $\mathcal{F}^*$ with the most relevant features of $\mathcal{F}$. So, a predefined quality measure is necessary to establish the level of relevance of the features. These methods are unable to detect simultaneous correlations between the features involved. Unlike filter algorithms, the wrapper methods are able to address correlations between features because they use the performance of the classifier to optimize the subset. Several optimization algorithms implement efficient wrapper methods but they usually transform the FS problem into a *NP-hard problem* [27]. In addition, these methods have the additional cost of reconstructing the classifier each time the feature subset is modified.

The hybrid techniques combine both methods, i.e., filter and wrapper, in order to provide a more efficient solution to the FS problem. Although they incorporate the advantages of both methods, they are prone to fail in identifying some relevant features. For example, the wrapper method could be incapable of including a relevant feature which was already considered irrelevant by the filter method.

3.2. A brief review of Binary Differential Evolution

There are several binary DE implementations reported in the literature. Some of them are prone to being trapped in a local optimal or making complex vector transformations. However, the proposal presented in this paper uses the idea behind the binary implementation of Gong and Tuson [28] in order to maintain the advantages of the original version of DE.

The Binary DE (BDE) algorithm evolves a population P of N_P vectors during g_{max} generations. The d-dimensional vectors are randomly initialized with a uniform distribution over $\{0, 1\}^d$. At each generation g, the *trial vector* $\mathbf{u}_i^{g+1}$ $(1 \le i \le N_P)$ is obtained by applying the mutation, recombination, and selection operators in turn on the basis of the *target vector* $\mathbf{x}_i^g \in P$.

Firstly, the mutation operator creates a mutant vector $\mathbf{v}_i^{g+1}$ by combining some solutions of the population. The most effective operator from among those reported in the literature is given by:

$$\mathbf{v}_i^{g+1} = \mathbf{x}_{r_0}^g + F \cdot (\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g). \tag{1}$$

This operator adds the difference of two vectors scaled by a factor F to the *base vector* $\mathbf{x}_{r_0}^g$. The indexes r_0, r_1, and r_2 are integers randomly selected under a uniform distribution on $\{1, \ldots, N_P\}$, where $i \neq r_0 \neq r_1 \neq r_2$. The scaling factor $F \in (0, 1+]$ controls the speed and robustness of the search.

When the vectors are binary, the effectiveness of the mutation operator depends on a careful re-definition of its operations. In consequence, the difference is calculated as the Hamming distance, $d = D_H(\mathbf{x}_{r_1}^g, \mathbf{x}_{r_2}^g)$ between the two vectors. Since the scaled difference must be interpreted as a integer value, $d' = F \times d$ must be rounded to its nearest integer as to a probability distribution. As a result, the *any-change* mutation operator redefines Eq. (1) as:

$$D_H(\mathbf{v}_i^{g+1}, \mathbf{x}_{r_0}^g) = \begin{cases} \lfloor d' \rfloor + 1 & \text{if } \; r < d' - \lfloor d' \rfloor, \\ \lfloor d' \rfloor & \text{otherwise}, \end{cases} \tag{2}$$

where r is a random number uniformly distributed over [0,1]. Therefore, the mutant vector is obtained by selecting one from among all the vectors at a certain distance from the base vector.

Secondly, the recombination operator creates the trial vector by mixing the mutant and target vectors. The most widely applied implementation is the *binomial crossover* defined as:

$$\mathbf{u}_{ij}^{g+1} = \begin{cases} \mathbf{v}_{ij}^{g+1} & \text{if } r_j \le Cr \text{ or } j = j_r, \\ \mathbf{x}_{ij}^g & \text{otherwise}, \end{cases} \tag{3}$$

where j_r is a randomly selected index with uniform distribution over $\{1, \ldots, d\}$. j_r is used to ensure that the trail vector differs from the target vector by at least one component. The remaining positions are inherited from the target vector or from the mutation vector according to a random number r_j uniformly distributed over [0, 1]. The crossover rate, $Cr \in [0, 1]$ controls the expected number of components inherited from the mutant vector. For binary vectors, no modification is necessary as this operator manipulates each component of the vector separately.

Finally, the selection operator choses the best solution as the new member of the population in the following way:

$$\mathbf{x}_i^{g+1} = \begin{cases} \mathbf{u}_i^{g+1} & \text{if } h(\mathbf{u}_i^{g+1}) \text{ is better than } h(\mathbf{x}_i^g), \\ \mathbf{x}_i^g & \text{otherwise}, \end{cases} \tag{4}$$

where the fitness function $h(\cdot)$ quantifies the quality of a solution.

4. The proposed Hybrid Feature Selection method

The main objectives addressed in the work present here are: (1) to identify the smallest subset of relevant features, and at the same time, (2) to obtain a classifier with the highest accuracy for classifying samples of microarray experiments. To achieve the former goal, a BDE-based wrapper FS method is hybridized with a ranking-based filter FS method. The latter goal is performed by one of the four ML algorithm considered in this paper: *Naive Bayes* (NB) [29], *Support Vector Machines* (SVM) [30], *k-Nearest Neighbor* (kNN) [31], and *C4.5-based decision trees* (C4.5) [32]. As shown in Fig. 1, the hybrid proposal uses a two-stage procedure to obtain the smallest feature subset with the highest discriminative capability.

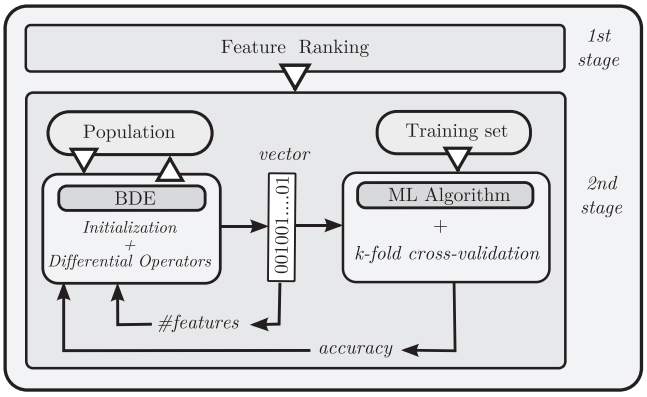


Fig. 1. Scheme of the Hybrid Feature Selection proposal.

In the first stage, a function employs the data set to quantify the predictive ability of each feature. Once the features have been ranked by their predictive quality, those at the top are more likely to better classify the sample.

In the second stage, the BDE algorithm searches for the best feature subset in the solutions space. Since the search should bias towards promising regions of the space, the score value of a solution must quantify both, the size of the subset and the predictive ability of the classifier constructed with that subset. In addition, in order to start with the best initial solutions, the BDE population is initialized with the features at the top of the ranking.

When the second stage finishes, the quality of the best obtained subset is estimated over a *testing* data set. The samples in the testing set are independent from those in the training set. This sort of validation is carried out to estimate the true accuracy value of the classifier.

The following sections show the most relevant aspects of the proposed technique. But it is important to first mention that to preserve the simplicity and efficiency of the proposal, the feature subsets are represented as binary vectors. More precisely, the size of the vector is equal to the number of features in the microarray. When the ith component of the vector is equal to 1, means that the subset includes the ith feature; otherwise, the component is set to 0.

4.1. Ranking of features

To arrange the features, rank-based filter FS techniques use a function that assesses the relevance of each feature [33]. According to relevance level, a feature is either *irrelevant*, or *weakly relevant*, or *strongly relevant*.

Following the definition given by John et al. [34] and assuming a probability distribution over the feature space $\mathcal{F}$, the relevance of $f \in \mathcal{F}$ depends conditionally on the class $c \in \mathcal{C}$. The relevance is strong when the value of f directly modifies the probability of occurrence of c, i.e., f is correlated with the class. Similarly, the relevance is weak when the probability only changes in some cases, i.e., the correlation is with other features. Finally, the feature f is irrelevant if its elimination does not modify the probability.

The dependence between a feature and the class values can be measured by the *information gain* (IG) [35]. Formally, the IG about f provided by $\mathcal{C}$ is defined as:

$$IG(\mathcal{C};f) = H(\mathcal{C}) - H(\mathcal{C}|f). \quad (5)$$

$H(\mathcal{C}) = -\sum_{c\in\mathcal{C}} p(c)\log_2 p(c)$ is the entropy of the classes, where $p(c)$ represents the prior probability of $c \in \mathcal{C}$. Let $p(c|x)$ be the posterior probability of c given the value x for f, $H(\mathcal{C}|f) = -\sum_{x\in f}\sum_{c\in\mathcal{C}} p(c|x)\log_2 p(c|x)$ is the conditional entropy.

Consequently, the higher the IG value, the more correlated the feature with the classes will be. Equation (5) will arrange the relevant features at the top of the ranking. Note that a discretization method must be applied in advance because the gene expression levels are continuous values.

To establish which are the more relevant features of the ranking, the usual criterion is to apply a threshold on the IG value. However, it could be more appropriate to employ a percentage of the features at the top of the ranking. Evidently, this policy helps to maintain a robust technique because it avoids fixing a value (i.e., the threshold) beforehand for each data set.

4.2. Population initialization

The way of initializing the vectors of the BDE population has an important role in this approach. Indeed, two different methods are applied. On the one hand, the features selected to initialize some vectors are taken from the most relevant ranked features. Assuming that the optimal solution consists of several strongly relevant features, the idea behind this method is to improve the initial solutions. On the other hand, the remaining vectors are generated following the basic approach in metaheuristics, i.e., a random initialization. These vectors will certainly provide solutions with less relevant features, so as to reduce the excessive pressure applied by the former method on some regions of the space.

Algorithm 1. Initialization of the BDE population

```
Input pPOP_R, pF, and R_pcent
                /* U(0, 1) is a generator of random numbers uniformly
                distributed on [0, 1] */
1:              for each vector x of the population P do
2:                if U(0, 1) < pPOP_R then
                    /* Ranking – based Initialization Method */
3:                    for each position j of x do
4:                      if U(0, 1) < pF/R_pcent then
5:                        set 1 to x(j)
6:                      else
7:                        set 0 to x(j)
8:                      end if
9:                    end for
10:               else
                    /* Random Initialization Method */
11:                   for each position j of x do
12:                     if U(0, 1) < pF then
13:                       set 1 to x(j)
14:                     else
15:                       set 0 to x(j)
16:                     end if
17:                   end for
18:               end if
19:             end for
```

Algorithm 1 shows an outline of the proposed initialization approach. Three user-specified parameters ($pPOP_R$, pF, and R_{pcent}) control the generation of the initial solutions. $pF \in [0, 1]$ controls the size of the initial feature subsets. A value of pF closer to 0 will produce solutions with a only few features. $pPOP_R \in [0, 1]$ indicates the rate at which the initial solutions undergo the ranking-based initialization method. In order to avoid an almost complete initial population of strongly relevant features, the value of $pPOP_R$ must be lower than 1. The boundary of the relevance on the ranking is regulated by $R_{pcent} \in [0, 1]$. The ranking-based initialization selects the features from the $100 \times R_{pcent}$ percent of the best ranked features. If R_{pcent} is over 0.5, the selection of the features will be almost random.

4.3. Fitness function

High quality solutions for feature selection problems in microarray experiments must fulfill two opposite objectives. On the one hand, a high quality solution, $\mathbf{x}$ requires having the least possible number of features. On the other hand, the classifier constructed from the solution must have the highest predictive ability. To maintain the simplicity of the model presented here, the fitness function is defined as a weighted linear aggregation function in order to handle the two objectives as a single-objective problem in the following way:

$$h(\mathbf{x}) = \beta \times accuracy + (1-\beta) \times \#features, \quad (6)$$

where *accuracy* is the percentage of samples correctly classified, $\#features$ is the size of feature subset, and $\beta \in [0, 1]$ is the weight that controls the aggregation of both objectives.

The accuracy is the most usual and adequate measure in classification problems of microarray samples. However, a small training set can produce a poor estimation of the accuracy [36]. An efficient and practical way of improving the estimation is to apply the *k-fold cross-validation* technique. Assuming the training set divided in k subsets of a similar size, the method estimates the performance of the classifier by averaging the partial estimations obtained from each subset. The division is *stratified* to keep the original proportion of samples of each class in each subset. In addition, $k = 10$ is suggested in [37] as a reasonable value to avoid an unnecessary increase in the computational cost.

5. Experimental study

For the sake of clarity, the first hybrid algorithm evaluated in this work is referred to as BDE-$\mathcal{X}_{Rank}$, where $\mathcal{X}$ stands for one of the four ML algorithms studied, i.e., $\mathcal{X} \in$ {SVM, NB, C4.5, kNN}. BDE-$\mathcal{X}_{Rank}$ was written completely in JAVA, to be used as a package within the user interface of WEKA [38]. This package extends the *attribute selection* classes of WEKA with the BDE search method. Regarding the four ML algorithms, the implementations provided by WEKA were used. Note that WEKA refers to the implementation of the *C4.5* algorithm as *J48*.

5.1. Data sets and parameter setting

BDE-$\mathcal{X}_{Rank}$ was evaluated over six well-known microarray experiment data sets. All of them are publicly available at the Kent Ridge Biomedical Dataset Repository [39]. Table 1 briefly summarizes the main characteristics for each data set. If the samples were originally arranged into a single data set, they were randomly divided in a stratified way into two subsets: the training set held two-thirds of the samples and the remaining third was placed in the testing set. In addition, the gene expression levels in the samples were normalized to $[-1, 1]$ to prevent a feature with large scale values from dominating either the selection or classification processes.

For the experimental study presented and discussed in this section, a parameter calibration of BDE-$\mathcal{X}_{Rank}$ was first conducted because DE is highly influenced by a suitable parameters' setting [46]. Table 2 shows the combination of the parameters' values selected to perform the experiments. So as to provide results with a high degree of confidence, all instances of the experiments carried out in this approach were independently executed 30 times.

Table 1
A brief summary of the data sets used in the experimental study.

Data set	No. of genes	Class	No. of samples		Reference
			Training	Testing	
Colon tumor	2000	Tumor	17	5	Alon et al. [40]
		Normal	29	11	
Leukemia	7129	ALL	27	20	Golub et al. [41]
		AML	11	14	
Lung cancer	12,533	MPM	16	15	Gordon et al. [42]
		ADCA	16	134	
Lymphoma-DLBCL	4026	DLBCL[a]	17	7	Alizadeh et al. [43]
		FL[b]	16	7	
Ovarian cancer	15,154	Tumor	108	54	Petricoin et al. [44]
		Normal	65	26	
Prostate cancer	12,600	Tumor	52	25	Singh et al. [45]
		Normal	50	9	

[a] *D*iffuse *L*arge *B*-*C*ell *L*ymphoma.
[b] *F*ollicular *L*ymphoma morphology.

Table 2
Parameter setting for the BDE algorithm and the initialization method

	Parameters' values
BDE algorithm	$N_P = 20$, $F = 0.05$, $Cr = 0.08$, $g_{max} = 2000$, $\beta = 0.3$
Initialization method	$pF = 0.04$, $pPOP_R = 0.5$, $R_{pcent} = 0.5$

It is worth mentioning that here, the best of the 30 executions for the ith proposal on the jth data set is that with the maximum value of $\{run_{ijk}\}_{k\in\{1,\ldots,30\}}$, where $run_{ijk} = (1 - \mathcal{N}(\#features_{ijk})) + \mathcal{N}(acc_{ts_{ijk}})$ relates, with a single value, to the size of the feature subset, $\#features_{ijk}$ and the accuracy on testing set, $acc_{ts_{ijk}}$. $\mathcal{N}(y) = (y - y^l)/(y^u - y^l)$ normalizes the value y into interval [0, 1]. y^l and y^u are, respectively, the lower and upper values in all the executions for all the evaluated algorithms on all the data sets. Data normalization allows a proper and honest comparison of the results of the executions.

5.2. Comparisons

Firstly, the advantages of applying a hybrid FS technique is analyzed. More precisely, the performance of BDE-$\mathcal{X}_{Rank}$ is compared with BDE-$\mathcal{X}$, i.e., the wrapper FS technique without a filter method. Table 3 reports on the accuracy (in percent) estimated from the testing set and the size of feature subset (enclosed in parentheses) for the best execution over both techniques with each ML algorithm for all the data sets. The results show that both algorithms have a similar performance. BDE-$\mathcal{X}_{Rank}$ improves the accuracy but in a few cases, it tends to increase (by one) the size of the feature subset, e.g., BDE-SVM$_{Rank}$ obtains 100% accuracy in Ovarian Cancer but increases by one the number of features with respect to BDE-SVM. In addition, it must be remarked that both algorithms achieve an accuracy percentage very close to 100% using much less than 1% of the features of the original data set. For example, BDE-C4.5$_{Rank}$ obtains 87.5% of accuracy with 2 out of 2000 genes in the Colon Tumor data set. The accuracy achieved by BDE-SVM$_{Rank}$ is 100% of the samples with only 3 out of 15,154 genes of the Ovarian Cancer data set.

Since BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}$ exhibit similar results, a statistical hypothesis test [47] was employed to determine with a certain level of confidence whether there exists a significant difference between them. Non-parametric tests were applied given the absence of enough evidence that could guarantee the conditions for applying the respective parametric tests. The Wilcoxon signed-rank test [48] carries out a pairwise comparison of the performance of two algorithms. This test evaluates the statistical significance of the null hypothesis, i.e., if the results of two algorithms come from the same population. The hypothesis is rejected when the p-value reported by the test is smaller than the significance level α.

Table 4 shows the comparison between BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}$ using Wilcoxon signed-rank test for each ML algorithm in turn. Considering a significance level of $\alpha = 0.05$, in the second column of Table 4 it can be observed that all the p-values are greater than 0.05. As a result, there are no significant differences in the final results of the two techniques.

However, a more precise comparison can reveal whether there are significant differences between the ML algorithms involved in BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}$. Specifically, the Friedman test [49] and a post-hoc test with Holm correction [50] are carried out. The first non-parametric test is a two-way analysis of variance based on ranking. Formulating the equality of medians among populations as the null hypothesis, the test allows significant differences to be detected between the results of two or more algorithms. When comparing the results of the ML algorithms involved in BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}$, the test indicates the existence of a significant difference (p-value = 1.32e−02).

Unfortunately, the Friedman test is unable to detect concrete differences between the results of the compared algorithms. Hence, a multiple pairwise test involving a control method must be carried out, and the p-values obtained must be corrected in order to

Table 3
Best results achieved by BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}$ for each classifier on the six data sets. Each cell reports the percentage of accuracy on the testing set and the size of the obtained feature subset (enclosed in parentheses). The best results for each data set are shown in bold.

	Colon tumor	Leukemia	Lung cancer	Lymphoma DLBCL	Ovarian cancer	Prostate cancer
BDE-SVM	68.8(1)	88.2(8)	98.7(3)	92.9(3)	98.8(2)	82.4(2)
BDE-SVM$_{Rank}$	75.0(3)	82.4(7)	98.0(3)	92.9(3)	**100.0(3)**	94.1(3)
BDE-kNN	87.5(3)	94.1(2)	97.3(1)	100.0(3)	92.5(1)	**97.1(3)**
BDE-kNN$_{Rank}$	81.3(2)	97.1(2)	**98.7(2)**	85.7(1)	98.8(2)	76.5(2)
BDE-NB	81.3(1)	91.2(1)	**98.7(2)**	85.7(1)	98.8(1)	29.4(2)
BDE-NB$_{Rank}$	81.3(1)	91.2(1)	97.3(2)	85.7(1)	96.3(1)	55.9(4)
BDE-C4.5	81.3(1)	**94.1(1)**	97.3(1)	**100.0(2)**	97.5(1)	73.5(3)
BDE-C4.5$_{Rank}$	**87.5(2)**	91.2(1)	97.3(1)	**100.0(2)**	97.5(1)	73.5(2)

Table 4
Comparison based on Wilcoxon signed-rank test.

Comparison	p-value
BDE-SVM vs. BDE-SVM$_{Rank}$	7.53e−01
BDE-kNN vs. BDE-kNN$_{Rank}$	3.45e−01
BDE-NB vs. BDE-NB$_{Rank}$	4.63e−01
BDE-C4.5 vs. BDE-C4.5$_{Rank}$	2.49e−01

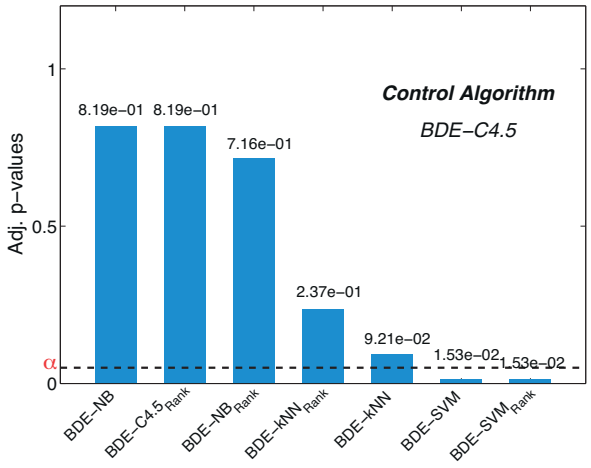


Fig. 2. Holm post-hoc test comparing the results between BDE-$\mathcal{X}$ and BDE-$\mathcal{X}_{Rank}$. The dashed line marks the significance level $\alpha = 0.05$.

avoid affecting the Family-Wise Error Rate. The Holm correction, in particular, is a powerful and suitable method to adjust these p-values.

On the basis of the ranking provided by the Friedman test on the evaluated algorithms, BDE-C4.5 appears to outperform the results of all the other algorithms. Considering BDE-C4.5 as the control method, Fig. 2 displays the results obtained by the Holm test (all the algorithms except the control algorithm are arranged on the x-axis according to the ranking). The adjusted p-values reveal that there are only significant differences with BDE-SVM and BDE-SVM$_{Rank}$. Undoubtedly, the difference is based on the ML technique rather than on the FS algorithm.

The tests have shown the absence of significant differences in the final results but with BDE-SVM and BDE-SVM$_{Rank}$. Despite this, there are cases in which BDE-$\mathcal{X}_{Rank}$ improves the searching of the subset, particularly, at the beginning of the evolutionary phase. Fig. 3 shows some examples of this behavior by tracing, in two plots, the first 500 generations of the best execution. The plots at the top show the evolution of the accuracy for the training set. The plots on the bottom trace the size of the feature subset during the evolutionary process. Analyzing the figures, both BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}$ have a similar number of features in common for the whole evolutionary process. However, BDE-$\mathcal{X}_{Rank}$ starts searching with a slightly superior accuracy with respect to BDE-$\mathcal{X}$. Moreover, for the greater part of the evolutionary process, the accuracy is relatively superior in BDE-$\mathcal{X}_{Rank}$. Clearly, the inclusion of a filter method benefits the FS process.

5.3. Fitness Function based on the relative frequency

On the basis of these results, a biased generation of the BDE initial population provides better starting solutions. Moreover, another approach potentially improving the searching of the best feature subset could be reached by modifying the fitness function (Eq. (6)). The modification proposed in this section focuses on exploiting the information contained in the BDE population. The rationale behind this new fitness function is that the most frequent features in the population will likely be the most relevant. In consequence, those vectors that include these features should have a better score.

Accordingly, the fitness function could be improved by using the relative frequency of the features involved in the current population. Let P^g be the population in the generation g, $\rho^g(j) = (1/N_P)\sum_{i=1}^{N_P} x_{ij}^g$ is the relative frequency of the jth feature. Hence, the probability of having selected all the features involved in vector $\mathbf{y}$ is $\delta(\mathbf{y}) = 1 - (\prod_{j=1}^{d} y_j(1 - \rho^g(j)))$. Finally, BDE-$\mathcal{X}_{Rank_f}$ computes the following fitness function:

$$h_f(\mathbf{x}) = \beta \times accuracy + (1 - \beta) \times \delta(\mathbf{x}) \times \#features, \quad (7)$$

where *accuracy*, *#features*, and β stand for the same elements as in Eq. (6). β is set to 0.3 (see Table 2) to provide a fair comparative study with BDE-$\mathcal{X}$ and BDE-$\mathcal{X}_{Rank}$.

Concerning the results obtained by BDE-$\mathcal{X}_{Rank_f}$, the Friedman test reports a significant statistical difference from the results of BDE-$\mathcal{X}$ (p-value = 1.53e−02). However, the adjusted p-values by post-hoc test with Holm correction depicted in Fig. 4(a) only show significant differences from BDE-SVM and BDE-SVM$_{Rank_f}$. Note that BDE-C4.5 acts as the control method since it is the best ranked algorithm by the Friedman test. In addition, the ranking establishes the order in which the algorithms should be arranged (excluding the control method) on the x-axis of Fig. 4(a).

Further to the statistical results obtained for BDE-$\mathcal{X}$ and BDE-$\mathcal{X}_{Rank_f}$, some examples can help to illustrate the difference between

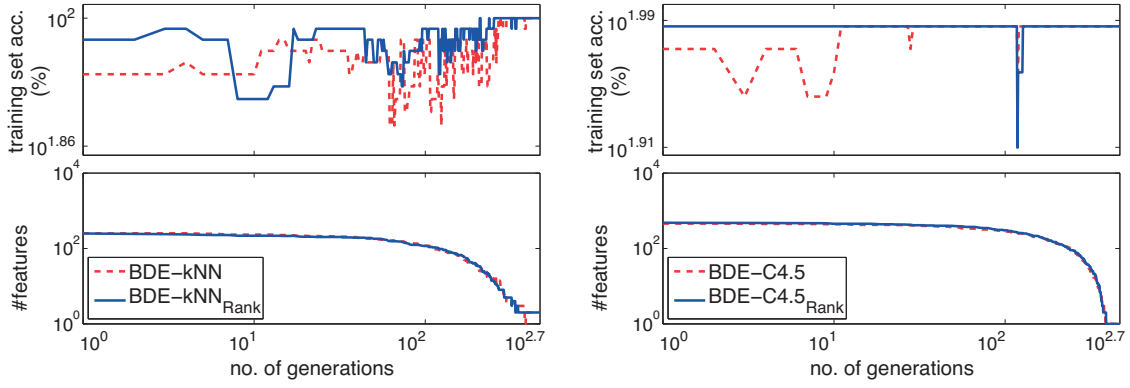


Fig. 3. Comparison of BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}$ with regard to the evolution of the accuracy value for the training set (plot at the top) and of the size of feature subset (plot on the bottom) for the first 500 generations of the best execution on two data sets. The plots are depicted on a *log-log* scale.

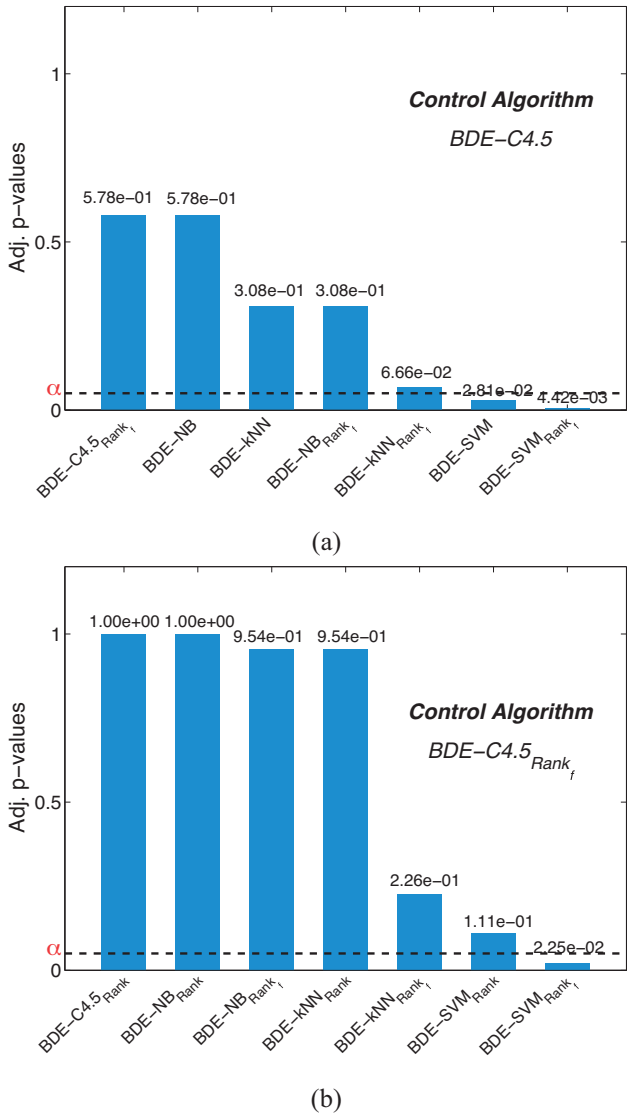


Fig. 4. Holm post-hoc test comparing to the results between (a) BDE-$\mathcal{X}_{Rank_f}$ and BDE-$\mathcal{X}$, and (b) BDE-$\mathcal{X}_{Rank_f}$ and BDE-$\mathcal{X}_{Rank}$. The dashed line marks the significance level $\alpha = 0.05$.

the two methods. Precisely, Fig. 5(a) and (b) present, as general cases, the evolutionary process carried out by both methods for two ML techniques over two data sets. Analyzing the first 500 generations of the best execution, BDE-$\mathcal{X}_{Rank_f}$ starts and ends the search yielding a better accuracy value than BDE-$\mathcal{X}$ (plots at the top for each data set). In addition, in almost all the cases, BDE-$\mathcal{X}_{Rank_f}$ and BDE-$\mathcal{X}$ maintain throughout the entire evolutionary search, similar sizes for the feature subsets, except in the Colon Tumor data set (plots on the bottom for each data set). As a result, BDE-$\mathcal{X}_{Rank_f}$ slightly improves the accuracy value with respect to BDE-$\mathcal{X}$ without increasing the size of the feature subset.

A similar conclusion can be reached when BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$ are compared. For example, the Friedman test reveals a significant difference (p-value = 3.07e−02) between the results. However, the post-hoc test with Holm's correction graphically represented in Fig. 4(b) only establishes the significant difference to the BDE-SVM$_{Rank_f}$. It is worth mentioning that BDE-C4.5$_{Rank_f}$ is the best ranked algorithm according to Friedman test.

Although the statistical tests are unable to detect differences between the results, a clear differentiation in the performance of the two methods can be observed. To highlight this, Fig. 5(c) and (d) are employed as general cases to compare the evolutionary process of BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$. Considering the plot at the top of each data set, BDE-$\mathcal{X}_{Rank_f}$ starts the evolutionary process from a feature subset with a slightly lower accuracy than BDE-$\mathcal{X}_{Rank}$. However, the accuracy value obtained by BDE-$\mathcal{X}_{Rank_f}$ over the evolutionary process outperforms the value reached by BDE-$\mathcal{X}_{Rank}$. Indeed, the improvement of the accuracy value occurs, although both methods maintain a similar number of features throughout all the evolutionary process (see the plot on the bottom for each data set). In consequence, BDE-$\mathcal{X}_{Rank_f}$ is able to find features with a better predictive ability in most of the data sets evaluated.

5.4. Comparative analysis based on ROC curve

Another manner to analyze and visualize the performance of classifiers is by the Receiver Operating Characteristics (ROC) curve. In fact, the machine learning community uses the ROC curve to incorporate additional conceptual information to the values given by a particular metric, e.g., accuracy. The ROC curve also allows a researcher to perform a better analysis of the results when the sample set has a unbalanced class and/or the samples have unequal classification error costs [51].

Regarding samples labeled with *positive class* (1) or *negative class* (−1), the ROC curve depicts on a *xy*-plane the relative trade-offs between benefits and cost of the classification. The *x*-axis is the *FP rate* and the *y*-axis represents the *TP rate*. *FP rate* and *TP rate* are computed according to following equations:

$$FPrate \approx FP/(TN + FP) \tag{8}$$

$$TPrate \approx TP/(TP + FN) \tag{9}$$

In both equation, TP and TN are, respectively, the number of positive and negative samples correctly predicted by the classifier. FN and FP are, respectively, the number of positive and negative samples wrongly predicted by the classifier.

Note that in the ROC space, the *probabilistic* classifiers (e.g., Naive Bayes) generate a continuous curve whereas the *discrete* ones (e.g., C4.5) only produce a single point. However, all the classifiers used in this paper generate a full ROC curve because the discrete classifier output is modified to produce a score rather than a class label.

Fig. 6 reports examples of the ROC curves for some of the microarray data sets used in this work. For the sake of clarity, the curves for BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$ were arranged in two different plots. In Fig. 6, the plots on the left side depict the performance of BDE-$\mathcal{X}_{Rank}$, and the ones on the right side correspond to the results of BDE-$\mathcal{X}_{Rank_f}$. Each curve plotted corresponds to the evaluation of the classifier $\mathcal{X}$ using the best feature subset obtained by the Hybrid Feature Selection algorithm. The red-dashed diagonal line $y=x$ plotted on each graph represents the classifiers randomly guessing the class. Thus, any classifier producing a curve in the lower right triangle performs worse than random guessing. On the contrary, a classifier predicting better than random guessing will generate a curve in the upper left triangle on the ROC space. Moreover, a higher predicting quality is obtained as the generated curve is closer to the northwest.

The plots shown in Fig. 6 are consistent with the statistical results revealed in the previous sections. For example, BDE-C4.5$_{Rank_f}$ and BDE-kNN$_{Rank_f}$ outperform BDE-C4.5$_{Rank}$ and BDE-kNN$_{Rank}$ respectively for Colon Tumor data set. A similar situation can be observed between BDE-C4.5$_{Rank_f}$ and BDE-C4.5$_{Rank}$ for Lung Cancer. However, BDE-NB$_{Rank}$ is outperformed by BDE-NB$_{Rank_f}$ for the Colon Tumor and Lung Cancer data sets, whereas BDE-C4.5$_{Rank_f}$ shows a behavior equal to BDE-C4.5$_{Rank}$ for Lymphoma-DLBCL data set.

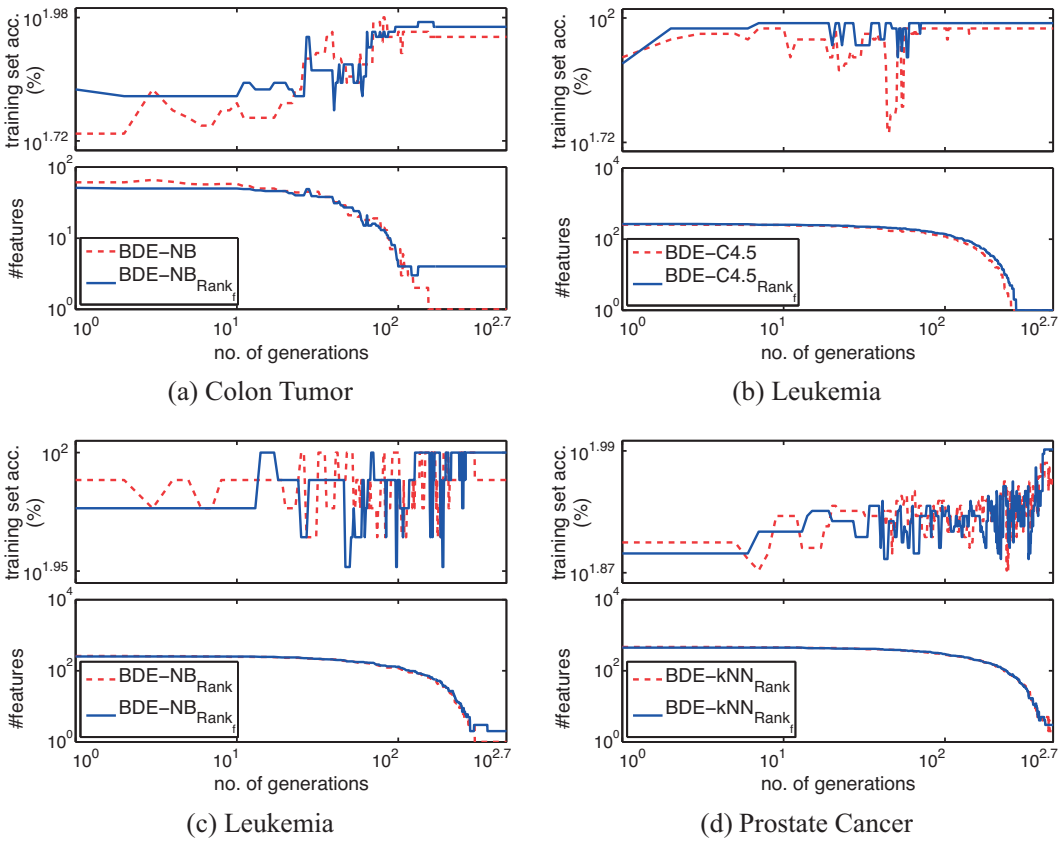


Fig. 5. Comparison of (a)–(b) BDE-$\mathcal{X}_{Rank_f}$ and BDE-$\mathcal{X}$, and (c)–(d) BDE-$\mathcal{X}_{Rank_f}$ and BDE-$\mathcal{X}_{Rank}$ regarding the evolution of the accuracy for the training set (plot at the top) and the size of feature subset (plot on the bottom) for the first 500 generations of the best execution over two data sets. The plots are depicted on the *log-log* scale.

Table 5
The AUCs of BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$ on each microarray data set. Regarding the comparison between algorithms, the best results are shown in bold.

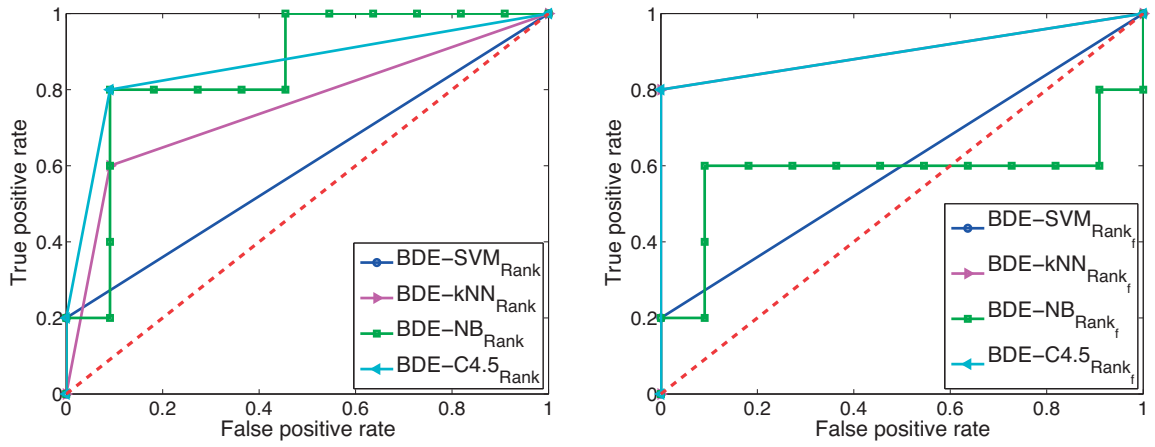
	Colon tumor	Leukemia	Lung cancer	Lymphoma DLBCL	Ovarian cancer	Prostate cancer
BDE-SVM$_{Rank}$	0.600	0.786	**0.989**	0.929	**1.000**	0.960
BDE-SVM$_{Rank_f}$	0.600	0.786	0.959	0.929	0.923	**0.980**
BDE-kNN$_{Rank}$	0.755	**0.964**	**0.963**	0.827	**0.981**	0.840
BDE-kNN$_{Rank_f}$	**0.900**	0.914	0.941	**0.857**	0.972	**0.844**
BDE-NB$_{Rank}$	**0.855**	0.964	**0.995**	0.837	0.998	0.389
BDE-NB$_{Rank_f}$	0.582	**0.968**	0.930	0.837	**1.000**	**0.600**
BDE-C4.5$_{Rank}$	0.864	0.914	0.867	1.000	0.972	0.500
BDE-C4.5$_{Rank_f}$	**0.900**	**0.939**	**0.948**	1.000	**0.981**	0.500

The ROC curve can be reduced to a single value by calculating of the area under the curve (AUC). The AUC value of a classifier represents the probability that the classifier will rank a randomly chosen positive sample higher than a randomly chosen negative sample. The AUC values for BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$ are shown in Table 5. The best results regarding the comparison between algorithms are shown in bold. Analyzing the values in this table, BDE-C4.5$_{Rank_f}$ achieves better results than BDE-C4.5$_{Rank}$. In spite of the fact that BDE-$\mathcal{X}_{Rank_f}$ shows an improvement in the feature selection process with respect to BDE-$\mathcal{X}_{Rank}$, the results depicted in this section report a similar final performance for both proposals.

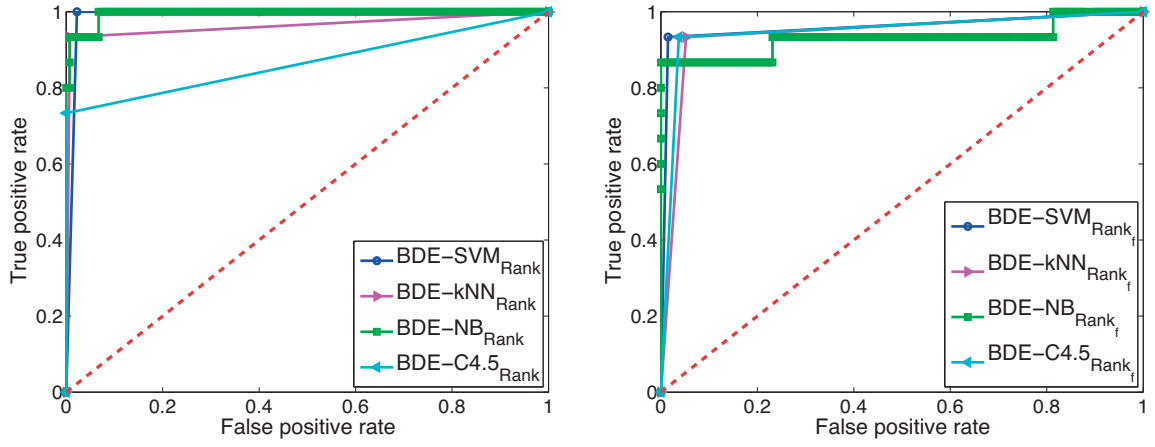
5.5. Comparison with state-of-the-art algorithms

Since BDE-$\mathcal{X}_{Rank_f}$ outperformed the results obtained by the other evaluated algorithms, this section extends the experimental study by comparing the performance de BDE-$\mathcal{X}_{Rank_f}$ with five state-of-the-art algorithms. Note that some of the algorithms involved in the comparison validate their results using cross-validation, others employ a testing data set, and the rests use a combination of both techniques. Moreover, the samples used in the validation are different from each other. For instance, García-Nieto et al. [52] split the whole data set into two subsets, one of them was used to select the features, and the other was employed to validate the final feature subset. Using a different approach, Bonilla-Huerta et al. [16] obtained the results with a 10-fold cross-validation method over the whole data set. Consequently, the heterogeneity of the aforementioned proposals and the different criteria used to validate their results mean that the conclusions cannot be definitive or precise. Nevertheless, the comparison gives an approximative measurement of the performance of BDE-$\mathcal{X}_{Rank_f}$ and any technically sound work should include, as we do, a comparison to the best techniques in the domain.

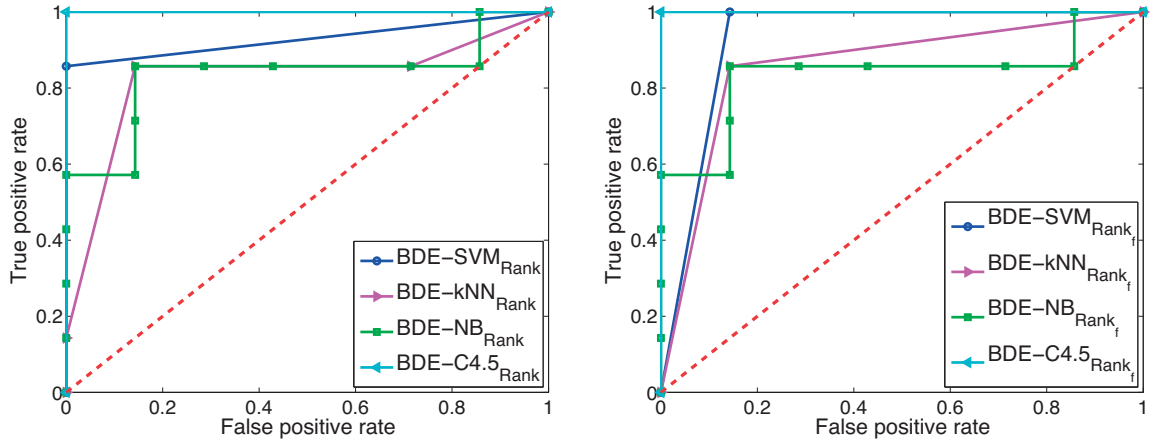
Table 6 summarizes the performance of the best found values by BDE-$\mathcal{X}_{Rank_f}$ for each ML algorithm and also the best results reported in the respective papers for the nine selected proposals being used for comparison. In addition, the best results obtained



(a) Colon Tumor



(b) Lung Cancer



(c) Lymphoma-DLBCL

Fig. 6. The ROC curve for BDE-$\mathcal{X}_{Rank}$ (on the left side) and BDE-$\mathcal{X}_{Rank_f}$ (on the right side) for each microarray data set. The red-dashed diagonal line $y{=}x$ represents the strategy of randomly guessing a class. (For interpretation of reference to color in this figure legend, the reader is referred to the web version of this article.)

by BDE-$\mathcal{X}_{Rank}$ are included in Table 6. Note that, each cell of the table contains the accuracy value together with the size of the feature subset in parenthesis. BDE-$\mathcal{X}_{Rank_f}$ shows an improvement on BDE-$\mathcal{X}_{Rank}$ for Colon Tumor and Leukemia and Ovarian Cancer and Prostate Cancer. However, both algorithms show a similar behavior for Lymphoma-DLBCL, whereas BDE-SVN$_{Rank_f}$ is marginally outperformed by BDE-kNN$_{Rank}$ for Lung Cancer.

When BDE-$\mathcal{X}_{Rank_f}$ is compared with the state-of-the-art algorithms, Table 6 shows that BDE-C4.5$_{Rank_f}$ and BDE-NB$_{Rank_f}$ achieve the lowest values with respect to the size of the feature subsets.

Table 6
Comparison of BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$ with other works of the current state-of-the-art (rows 1 to 9) for each data set (columns). The symbol "–" means that no information is available. The best results for BDE-$\mathcal{X}_{Rank_f}$ are highlighted in bold font.

	Colon tumor	Leukemia	Lung cancer	Lymphoma DLBCL	Ovarian cancer	Prostate cancer
8-S PMSO [52]	94.2(20)	98.1(20)	100(20)	98.7(20)	–	–
LDA-GA [16]	98.8(7)	99.5(3)	99.1(3)	99.5(3)	97.4(6)	99.5(3)
PLSDR [53]	83.5(20)	97.1(20)	-	93.0(20)	99.9(20)	91.7(20)
BCGS[a] [54]	83.8(23)	94.1(35)	91.2(34)	-	98.8(26)	-
GEM[b] [55]	91.2(8)	91.5(3)	-	93.3(5)	-	-
IWSS[2] [56]	-	94.4(7.9)	–	–	–	90.2(13.2)
IWSS[3]-MB-NB [56]	86.0(5.2)	97.1(6.4)	–	–	–	91.1(5.6)
DRF0-CFS [57]	90(10)	91.18(13)	98.66(17)	93.33(11)	100(16)	85.29(113)
IRLDA [58]	–	97(72)	–	–	–	–
BDE-SVM$_{Rank}$	75.0(3)	82.4(7)	98.0(3)	92.9(3)	100.0(3)	94.1(3)
BDE-kNN$_{Rank}$	81.3(2)	97.1(2)	98.7(2)	85.7(1)	98.8(2)	76.5(2)
BDE-NB$_{Rank}$	81.3(1)	91.2(1)	97.3(2)	85.7(1)	96.3(1)	55.9(4)
BDE-C4.5$_{Rank}$	87.5(2)	91.2(1)	97.3(1)	100.0(2)	97.5(1)	73.5(2)
BDE-SVM$_{Rank_f}$	75.0(4)	82.4(6)	**98.0(3)**	92.9(3)	95.0(3)	**97.1(3)**
BDE-kNN$_{Rank_f}$	93.8(4)	91.2(2)	94.6(1)	85.7(2)	97.5(2)	82.4(3)
BDE-NB$_{Rank_f}$	81.3(2)	94.1(2)	95.3(1)	85.7(1)	**100.0(2)**	61.8(2)
BDE-C4.5$_{Rank_f}$	**93.8(3)**	**94.1(1)**	96.0(1)	**100.0(2)**	97.5(1)	73.5(2)

[a] Bootstrapping consistency gene generation.
[b] Genetic embedded method.

Moreover, almost all the subsets found only have three or less features. Regarding the percentage of accuracy, BDE-C4.5$_{Rank_f}$ obtains 100% accuracy with two features on Lymphoma-DLBCL outperforming the results of all the other proposals. For the Ovarian data set, BDE-NB$_{Rank_f}$ also obtains 100% accuracy with two features. The performance of BDE-C4.5$_{Rank_f}$ is highly acceptable for Colon Tumor and Lung Cancer; although it is a little lower than proposals [52] and [16]. A similar situation exists with BDE-NB$_{Rank_f}$ for the Leukemia data set. For the Prostate Cancer data set, BDE-SVM$_{Rank_f}$ obtains an accuracy close to 100% but the value is a little lower than [16]. In addition, it must be remarked that the best feature subset obtained by BDE-$\mathcal{X}_{Rank_f}$ has less than 1% on the total amount of features into original set. Moreover, in almost all cases, BDE-$\mathcal{X}_{Rank_f}$ found a subset with a number of features much lower than the reported one by the state-of-the-art proposals presented in Table 6. For examples, BDE-NB$_{Rank_f}$ obtained 100% of accuracy with 2 out of 15,154 features in the set, whereas the state-of-the-art algorithms showed values much higher than 2.

6. Conclusions

In this paper, we have proposed two simple and efficient hybrid FS algorithms, named respectively BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$, to deal with the issues related to the classification process of high-dimensional microarray samples. Both proposals combine a filter method with a wrapper method in a two-stage algorithm. In the first stage, a rank-based filter technique sorts the features by the information gain values. In the second stage, a Binary DE-based wrapper method performs the search for a high quality solution. In addition, in our proposals, the BDE initial population is biased using the ranking of features. Therefore, some initial solutions are generated with only highly relevant ranked features. The others are initialized with only random features in order to include diversity in the initial population. A key characteristic of our proposals is that all the initial solutions are generated with a small number of features. Regarding BDE-$\mathcal{X}_{Rank_f}$, we incorporate a new fitness function in order to propose an enhanced version of the BDE-$\mathcal{X}_{Rank}$ algorithm. The score value computed by the new fitness function is influenced by the frequency of the features present in the current population.

Six high-dimensional well-known microarray experiment data sets and four ML algorithms (*NB*, *SVM*, *C4.5*, and *kNN*) have been used, in an extensive experimental study based on statistical tests, to evaluate BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$. The reported results have shown the robustness of BDE-$\mathcal{X}_{Rank}$ and BDE-$\mathcal{X}_{Rank_f}$. Both algorithms are also able to obtain more accurate solutions than BDE-$\mathcal{X}$ at the earlier stages of the evolutionary search. Despite the absence of significant differences in the final results, BDE-$\mathcal{X}_{Rank_f}$ has also shown a slight improvement in the final accuracy values with respect BDE-$\mathcal{X}_{Rank}$. In addition, the comparison with nine proposals of the state-of-the-art has proven the competitiveness of our BDE-$\mathcal{X}_{Rank_f}$. We must remark that BDE-$\mathcal{X}_{Rank_f}$ and BDE-$\mathcal{X}_{Rank}$ are able to reduce the original feature/gene set size by more than 99% and, at the same time, obtain highly competitive results.

Future work may include a crossover operator that incorporates the ranking information to improve the performance of BDE-$\mathcal{X}_{Rank_f}$. Also, an informed local search operator may be applied to speed up the convergence of the evolutionary process. Finally, it may also include a biological analysis of the results because it is information of interest to experts in the field of Biology.

Acknowledgements

J. Apolloni and G. Leguizamón are supported by PROICO 330214 from Universidad Nacional de San Luis. The work of Prof. Alba has been partially funded by the University of Málaga UMA/FEDER FC14-TIC36, *programa de fortalecimiento de las capacidades de I+D+I en las universidades 2014–2015, de la Consejería y Economía, Innovación, Ciencia y Empleo*, with European FEDER, and also partially funded by the Spanish MINECO project TIN2014-57341-R (http://moveon.lcc.uma.es).

References

[1] D.K. Tasoulis, V.P. Plagianakos, M.N. Vrahatis, Differential evolution algorithms for finding predictive gene subsets in microarray data, in: Artificial Intelligence Applications and Innovations, Springer, 2006, pp. 484–491.
[2] S.B. Kotsiantis, Supervised machine learning: a review of classification techniques, Emerg. Artif. Intell. Appl. Comput. Eng. 160 (2007) 3–24.
[3] H. Liu, H. Motoda, Computational Methods of Feature Selection, Chapman and Hall/CRC, 2007.
[4] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1) (1997) 273–324.
[5] M. Zhao, J. Ren, L. Ji, C. Fu, J. Li, M. Zhou, Parameter selection of support vector machines and genetic algorithm based on change area search, Neural Comput. Appl. 21 (1) (2012) 1–8.

[6] E.B. Huerta, B. Duval, J.-K. Hao, A hybrid GA/SVM approach for gene selection and classification of microarray data, in: Applications of Evolutionary Computing, Springer, 2006, pp. 34–44.
[7] K.R. Robbins, W. Zhang, K. Bertrand, R. Rekaya, The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification, Math. Med. Biol. 24 (4) (2007) 413–426.
[8] L. Ke, Z. Feng, Z. Ren, An efficient ant colony optimization approach to attribute reduction in rough set theory, Pattern Recogn. Lett. 29 (9) (2008) 1351–1357.
[9] Y. Chen, Y. Zhao, A novel ensemble of classifiers for microarray data classification, Appl. Soft Comput. 8 (4) (2008) 1664–1669.
[10] J. García-Nieto, E. Alba, Parallel multi-swarm optimizer for gene selection in DNA microarrays, Appl. Intell. 37 (2) (2012) 255–266.
[11] K.V. Price, R. Storn, J. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer, London, 2005.
[12] J. Xie, C. Wang, Using support vector machines with a novel hybrid feature selection method for diagnosis of erythemato-squamous diseases, Expert Syst. Appl. 38 (5) (2011) 5809–5815.
[13] Y. Peng, Z. Wu, J. Jiang, A novel feature selection approach for biomedical data classification, Biomed. Inform. 43 (1) (2010) 15–23.
[14] Y.-P. Lee, W.-Y. Han, W.-J. Lin, K.-S. Wu, A hybrid feature selection method using modular perceptron networks, Asian J. Inform. Technol. 5 (10) (2006) 1088–1094.
[15] L.-X. Zhang, J.-X. Wang, Y.-N. Zhao, Z.-H. Yang, A novel hybrid feature selection algorithm: using relieff estimation for ga-wrapper search, in: International Conference on Machine Learning and Cybernetics, Vol. 1, IEEE, 2003, pp. 380–384.
[16] E. Bonilla-Huerta, B. Duval, J.C.H. Hernández, J.-K. Hao, R. Morales-Caporal, Hybrid filter-wrapper with a specialized random multi-parent crossover operator for gene selection and classification problems, in: in: Bio-Inspired Computing and Applications, Springer, 2012, pp. 453–461.
[17] M.M. Kabir, M. Shahjahan, K. Murase, A new hybrid ant colony optimization algorithm for feature selection, Expert Syst. Appl. 39 (3) (2012) 3747–3763.
[18] X. He, Q. Zhang, N. Sun, Y. Dong, Feature selection with discrete binary differential evolution, in: International Conference on Artificial Intelligence and Computational Intelligence (AICI'09), Vol. 4, IEEE, 2009, pp. 327–330.
[19] M. Marinaki, Y. Marinakis, An island memetic differential evolution algorithm for the feature selection problem, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2013), Springer, 2014, pp. 29–42.
[20] R.N. Khushaba, A. Al-Ani, A. Al-Jumaily, Feature subset selection using differential evolution and a statistical repair mechanism, Expert Syst. Appl. 38 (9) (2011) 11515–11526, http://dx.doi.org/10.1016/j.eswa.2011.03.028.
[21] A. Al-Ani, A. Alsukker, R.N. Khushaba, Feature subset selection using differential evolution and a wheel based search strategy, Swarm Evol. Computat. (2013) 15–26.
[22] J. García-Nieto, E. Alba, J. Apolloni, Hybrid DE-SVM approach for feature selection: Application to gene expression datasets, in: Proceedings of the 2nd International Symposium on Logistics and Industrial Informatics, 2009, pp. 52–57.
[23] S.C. Satapathy, A. Naik, Hybridization of rough set and differential evolution technique for optimal features selection, in: Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012), Springer, 2012, pp. 453–460.
[24] J. Apolloni, G. Leguizamón, E. Alba, DE-SVM$_{Rank}$: a differential evolution algorithm with a rank-based feature selection process for microarray data classification, in: XVIII Congreso Argentino de Ciencias de la Computación, 2012, pp. 180–189.
[25] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[26] Y. Saeys, I.N. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[27] P.M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. Comput. 26 (9) (1977) 917–922.
[28] T. Gong, A. Tuson, Differential evolution for binary encoding, in: A. Saad, K. Dahal, M. Sarfraz, R. Roy (Eds.), Soft Computing in Industrial Applications, Vol. 39 of Advances in Soft Computing, Springer, Berlin, Heidelberg, 2007, pp. 251–262.
[29] B. Cestnik, Estimating probabilities: a crucial task in machine learning, in: ECAI, Vol. 90, 1990, pp. 147–149.
[30] C. Cortes, V. Vapnik, Support vector machine, Mach. Learn. 20 (3) (1995) 273–297.
[31] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inform. Theory 13 (1) (1967) 21–27.
[32] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[33] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, A. Nowé, A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Computat. Biol. Bioinform. (TCBB) 9 (4) (2012) 1106–1119.
[34] G.H. John, R. Kohavi, K. Pfleger, Irrelevant features and the subset selection problem, in: ICML'94, 1994, pp. 121–129.
[35] Q. Song, J. Ni, G. Wang, A fast clustering-based feature subset selection algorithm for high dimensional data, IEEE Trans. Knowledge Data Eng. 99 (2011) 1, doi:10.1109/TKDE.2011.181.
[36] R. Simon, M.D. Radmacher, K. Dobbin, L.M. McShane, Pitfalls in the use of dna microarray data for diagnostic and prognostic classification, Natl. Cancer Inst. 95 (1) (2003) 14–18.
[37] B. Duval, J.-K. Hao, Advances in metaheuristics for gene selection and classification of microarray data, Brief. Bioinform. (2009) 127–141, http://dx.doi.org/10.1093/bib/bbp035.
[38] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2005.
[39] Kent Ridge Biomedical Dataset Public Repository. URL http://datam.i2r.a-star.edu.sg/datasets/krbd/index.html.
[40] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. U.S.A. 96 (12) (1999) 6745–6750.
[41] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.
[42] G.J. Gordon, R.V. Jensen, L.-L. Hsiao, S.R. Gullans, J.E. Blumenstock, S. Ramaswamy, W.G. Richards, D.J. Sugarbaker, R. Bueno, Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma, Cancer Res. 62 (17) (2002) 4963–4967.
[43] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, et al., Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling, Nature 403 (6769) (2000) 503–511.
[44] E.F. Petricoin III, 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, et al., Use of proteomic patterns in serum to identify ovarian cancer, Lancet 359 (9306) (2002) 572–577.
[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, et al., Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2) (2002) 203–209.
[46] D. Koloseni, J. Lampinen, P. Luukka, Optimized distance metrics for differential evolution based nearest prototype classifier, Expert Syst. Appl. 39 (12) (2012) 10564–10570.
[47] J. Derrac, S. Garcí a, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Computat. 1 (1) (2011) 3–18.
[48] J.H. Zar, Biostatistical Analysis, Prentice Hall, 2009.
[49] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Am. Stat. Assoc. 32 (1937) 675701.
[50] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (1979) 6570.
[51] T. Fawcett, An introduction to ROC analysis, Pattern Recog. Lett. 27 (8) (2006) 861–874.
[52] J. García-Nieto, E. Alba, Parallel multi-swarm optimizer for gene selection in DNA microarrays, Appl. Intell. 37 (2) (2012) 255–266.
[53] 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 the 7th IEEE International Conference on Bioinformatics and Bioengineering, IEEE, 2007, pp. 1439–1444.
[54] S. Pang, I. Havukkala, Y. Hu, N. Kasabov, Classification consistency analysis for bootstrapping gene selection, Neural Comput. Appl. 16 (6) (2007) 527–539.
[55] J.C.H. Hernandez, B. Duval, J.-K. Hao, A genetic embedded approach for gene selection and classification of microarray data, in: Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer, 2007, pp. 90–101.
[56] A. Wang, N. An, G. Chen, J. Yang, L. Li, G. Alterovitz, Incremental wrapper based gene selection with markov blanket, in: IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2014, IEEE, 2014, pp. 74–79.
[57] V. Bolón-Canedo, N. Sánchez-Maro no, A. Alonso-Betanzos, Distributed feature selection: An application to microarray data classification, Appl. Soft Comput. 30 (2015) 136–150.
[58] A. Sharma, K.K. Paliwal, S. Imoto, S. Miyano, A feature selection method using improved regularized linear discriminant analysis, Mach. Vision Appl. 25 (3) (2014) 775–786.