RAIRO-Oper. Res. 53 (2019) 269–288
https://doi.org/10.1051/ro/2018059

RAIRO Operations Research
www.rairo-ro.org

GENE SELECTION *VIA* BPSO AND BACKWARD GENERATION FOR CANCER CLASSIFICATION

Ahmed Bir-jmel, Sidi Mohamed Douiri* and Souad Elbernoussi

Abstract. Gene expression data (DNA microarray) enable researchers to simultaneously measure the levels of expression of several thousand genes. These levels of expression are very important in the classification of different types of tumors. In this work, we are interested in gene selection, which is an essential step in the data pre-processing for cancer classification. This selection makes it possible to represent a small subset of genes from a large set, and to eliminate the redundant, irrelevant or noisy genes. The combinatorial nature of the selection problem requires the development of specific techniques such as filters and Wrappers, or hybrids combining several optimization processes. In this context, we propose two hybrid approaches (RBPSO-1NN and FBPSO-SVM) for the gene selection problem, based on the combination of the filter methods (the Fisher criterion and the ReliefF algorithm), the BPSO metaheuristic algorithms and the Backward algorithm using the classifiers (SVM and 1NN) for the evaluation of the relevance of the candidate subsets. In order to verify the performance of our methods, we have tested them on eight well-known microarray datasets of high dimensions varying from 2308 to 11225 genes. The experiments carried out on the different datasets show that our methods prove to be very competitive with the existing works.

Mathematics Subject Classification. 62F07; 62H30; 62P10; 68T20; 90C06.

Received April 30, 2017. Accepted July 7, 2018.

1. Introduction

The recent advance in DNA microarray technology [34] has allowed researchers to analyze the expression levels of several thousands of genes simultaneously. These levels are very important for cancer classification. The raw microarray data is transformed into gene expression matrices. Generally, a row in the matrix represents an experimental condition (sample), and a column represents a gene. The numeric value m_{ij} is the expression level of the jth gene in the ith sample see Table 1.

For most biological problems, each cell line contains information about the state (class) of a tissue (cancerous or not) under a given experimental condition. By means of the interesting class information, we can formulate the DNA microarray analysis as a supervised classification task [25].

Keywords. Gene selection, cancer classification, BPSO, backward generation, SVM, 1NN, ReliefF, Fisher criterion, DNA microarray.

Laboratory of Mathematics, Computing & Applications-Information Security, Faculty of Sciences, Mohammed V University in Rabat, BP1014 Rabat, Morocco.
* Corresponding author: douirisidimohamed@gmail.com

Article published by EDP Sciences

TABLE 1. A gene expression matrix.

Gene$_{id}$	Gene$_1$	Gene$_2$	$\cdots$	Gene$_N$
Sample$_1$	m$_{11}$	m$_{12}$	$\cdots$	m$_{1N}$
Sample$_2$	m$_{21}$	m$_{22}$	$\cdots$	m$_{2N}$
Sample$_3$	m$_{31}$	m$_{32}$	$\cdots$	m$_{3N}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Sample$_M$	m$_{M1}$	m$_{M2}$	$\cdots$	m$_{MN}$

However, gene expression data has characteristics of high dimension, high noise, and small sample size. This noisy data limits the performance of classifiers. As we have a small number of samples whereas each sample is described by a very large number of genes, one of the main challenges in gene expression analysis is to select a small subset of genes that contain the necessary information and relevant to a given cancer [29].

In this context, the feature (gene) selection has become the object that attracts the attention of many researchers over the last few years, this selection makes it possible to identify and remove unneeded, irrelevant and redundant attributes that penalize the performances of the classifiers. It also facilitates interpretation and understanding of medical and biological aspects, thus, it defies the curse of dimensionality to improve the performance of classification methods. Generally, there are two subclasses of feature selection algorithms namely: Filter methods and Wrapper methods [23]. In Filter Approaches the selection is independent of the classifier used, the principle of these methods consists in evaluating each feature individually, to assign it a score. The feature selection is done by choosing the best ranked features. These methods are easily implemented. The main negative point of these methods is that they evaluate the features individually by neglecting the possible interactions with the other features. In the literature, there are several individual gene ranking methods (Filter) methods such as Fischer score [16], *t*-test [19], signal-to-noise-ratio [31], information gain [40], and ReliefF [24], etc. While as in Wrapper methods, the features are selected using the classifier. Their general idea consists of generating candidate subsets and evaluating them using a specific classifier, these methods give a high performance (accuracy), and they are simple: just generate and evaluate. Although, they typically suffer from increased computational cost and possible overfit [23]. In Wrapper methods meta-heuristics are commonly used to generate high-quality subsets of features, the evaluation is carried out by using a specific classifier.

Examples of classifiers used for evaluating each candidate solution include K-nearest neighbor (KNN) and support vector machines (SVMs). They are listed in the top 10 most influential data-mining algorithms in the research community [42].

The first works on the classification of oncological data were published at the end of the 1990s [3,14]. In [14], for example, the data obtained from DNA microarray made it possible to discriminate two forms of leukemia. Moreover, among the 6817 genes tested, a small number of genes (about 50) appear to be very important for the recognition of the two forms of the disease.

The common goal of metaheuristic algorithms is to find near-optimal solutions for a given problem, as of example the PSO, ACO, and GA are considered as the most popular population-based metaheuristics that can be used for combinatorial optimization in which an optimal solution is sought over a discrete search-space. Particularly the gene selection problem, as such there is no best algorithm in data mining. Several researchers have utilized metaheuristics methods for solving gene selection problem, in order to facilitate recognition of cancer cells: PSO [1, 2, 6, 8, 9, 29, 32] genetic algorithm [2, 29, 33]; ACO [7, 28, 36, 39, 44], Binary Differential Evolution (BDE) algorithm [5], and incorporating imperialist competition algorithm (ICA) [41].

The Particle Swarm Optimization algorithm (abbreviated as PSO) is a Population-based metaheuristic [20]. Thanks to its efficiency, simple concepts and rapid convergence, it has been used to solve several optimization problems in different fields. In PSO each particle presents a candidate solution for the problem, the particles are assumed to "fly" over the search space, searching for the global optimum. The displacement of each particle is influenced by its velocity and the best particle position.

In the context of gene selection for cancer classification, there are thousands of genes, then it is inefficient to adopt an evolutionary algorithm directly in a microarray dataset without passing by a pre-treatment step. In this step, a subset of genes is selected by using a filter selection method in the majority of cases, then an evolutionary algorithm is employed to select the final subset [39]. As examples, Chi-square statistics and a GA are used in [27]. Information gain filter and a memetic algorithm in [46], information gain and improved simplified swarm optimization [26]. Zhao *et al.* proposed a hybrid approach by combining Fisher score with a GA and PSO [45]. ReliefF, mRMR (Minimum Redundancy Maximum Relevance) and GA [37].

Thus, metaheuristics have been successfully used for solving the gene selection problem which known to be NP-hard [4, 12]. In this study, we propose two hybrid approaches for solving the gene selection problem. The first proposed approach is based on the combination of a Filter type method using Fisher Score [16], and a wrapper method based on PSO and Local Search algorithm (hill climbing) guided by SVM classifier. In this approach, the role of SVM is to evaluate a candidate gene subset generated. The second approach is similar to the first, just we replace Fisher Score by ReliefF algorithm [24], and we use a 1NN classifier to evaluate candidate solutions. These proposed approaches have not been previously investigated. Moreover, this work investigates the performance of our methods, by applying them to eight gene expression data sets, including binary classes and multiclass data sets.

This paper is organized as follows: In the next section we present our new gene selection methods. The third section presents the experiments that we have carried out to evaluate our approaches. Finally, we conclude our paper.

2. Methods

2.1. Filter approaches

To apply the Filter approach, we need to define some tools to assign a score to a gene.

2.1.1. Fisher score F_i

Allows to measure the degree of separability of the classes, by means of a given gene g_i. It is defined by:

$$F_i = \frac{\sum_{k=1}^{c} n_k(\mu_k^i - \mu^i)^2}{\sum_{k=1}^{c} n_k(\sigma_k^i)^2}$$

where c, n_k, μ_i^k and σ_k^i represent respectively the number of classes, the effective, the mean and the standard deviation of kth class corresponding to the ith gene. μ_i is the global mean of the ith gene.

We could say that the measure is related to the interclass variance of the gene.

2.1.2. ReliefF

This algorithm was introduced under the name of Relief in [22] and then improved and adapted to the multi-class case by Kononenko [24], under the name of ReliefF. It does not just eliminate redundancy but also defines a criterion of relevance (weight). This weight measures the ability of each gene to group the data similarly labeled and discriminate those with different labels.

If the weight of a gene is large, it means that the data coming from the same class have close values, and the data coming from different classes are well separated.

Gene selection problem can be viewed as an optimization problem, where we try to find the best gene subset which maximizes an objective function f, then we need to define some search algorithms (Search Strategy).

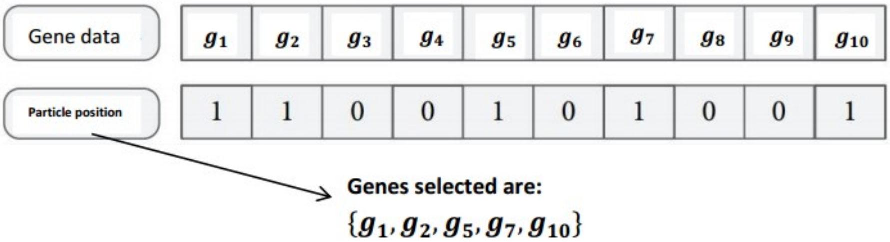

FIGURE 1. An illustrated example with generated subset and particle representation.

2.2. Binary particle swarm optimization

Binary Particle Swarm Optimization (BPSO) is a population-based metaheuristic proposed by Kennedy and Eberhart [21]. The BPSO algorithm is inspired by the social behavior of swarms, such as bird flocking or fish schooling. It is the adaptation of the PSO algorithm to solve binary optimization problems.

In gene selection, the position of each particle i, $X_i = [x_i^1, x_i^2, \ldots, x_i^p]$ (binary coded) presents a candidate solution (gene subset) in a p-dimensional space, where each bit of the particle represents a gene, if a bit has a value of "1", this means that gene has been selected. On the other hand, a value of "0" indicates that the gene is not selected in the subset (Fig. 1). The movement of a particle is influenced by its velocity $V_i = [v_i^1, v_i^2, \ldots, v_i^p]$, its best position already visited $P_i = [p_i^1, p_i^2, \ldots, p_i^p]$ and its best neighbour's position $P_{gi} = [p_{gi}^1, p_{gi}^2, \ldots, p_{gi}^p]$.

The update of X_i and V_i is done as follows:

$$v_i^d = w * v_i^d + c_1 * U(0,1) * (p_i^d - x_i^d) + c_2 * U(0,1) * (p_{gi}^d - x_i^d). \tag{2.1}$$

$$\mathrm{Sig}(t) = \frac{1}{1+e^{-t}}$$

$$x_i^d = \begin{cases} 1, & \text{if} \quad \mathrm{Sig}(v_i^d) > U(0,1) \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

where $i = 1 \ldots m$, and
$d = 1 \ldots p$.

Where U is a uniformly distributed random variable, c_1 and c_2 are positive constant, represent the acceleration coefficients, indicating the cognitive and the social learning factors, respectively. w is the inertia weight. The values of the velocities are between $v_{\min}$ and $v_{\max}$. The size of the swarms is m.

The function Sig is used to transform the velocity vector into a set of probabilities. Where $\mathrm{Sig}(v_i^d)$ represents the probability of bit x_i^d taking the value 1. The initial swarm is generated randomly.

2.3. Backward generation (Hill Climbing)

Backward generation is a local search method for gene selection (Hill Climbing), in this iterative algorithm, we start from a set of genes and we try to remove a not relevant gene at each iteration (the gene which does not influence the accuracy of the classification must be eliminated). The coding of the candidate solutions is the same of BPSO (Fig. 2).

Let f be a function that evaluates the quality of a subset of gene. The follow algorithm presents the backward generation.

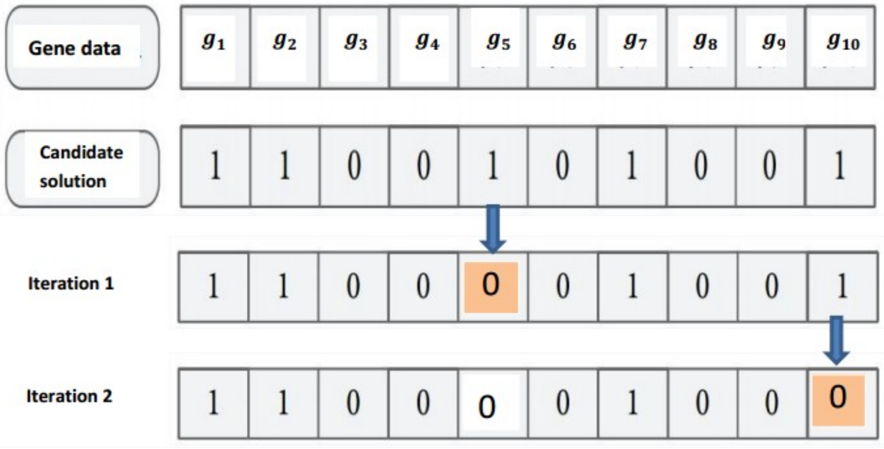

FIGURE 2. Example illustrates the simulation of two iterations of the backward approach; g_5 and g_{10} are not relevant.

Algorithm 1. Backward generation.

```
Input: candidate solution X (genes subset), maximum number of iterations $it_{max}$
Output: $X_{best}$
Begin
for ($i = 1 : it_{max}$) do
    Randomly choosing a gene ($g_i$) from X
    $X' = X \backslash g_i$
    if (f(X') > f(X)) then
        X=X'
    end if
end for
$X_{best} = X$
End
```

2.4. Selection criteria

We studied some search strategies for finding the optimal gene subset. We now need to define "what is a good gene subset?" Without defining a measure for the evaluation of each subset, we can't talk about the best or optimal gene subset [30].

This evaluation can be often measured by an objective function f which presents a compromise between the number of genes selected and the classifier's accuracy, in gene selection we seek to maximize predictive accuracy and minimize the number of genes used.

In order to estimate predictive accuracy, we need a classifier that can be tested on the test data after learning from the training data. In this study we use two classifiers: Linear SVM and KNN.

2.5. Linear SVM

Support Vector Machine (SVM) is specifically designed for two-class problems (binary classification) [10,35]. The basic principle of SVM is to find the optimal hyperplane $w^T x + b = 0$ (w denotes the normal vector of the hyperplane) that separates two classes in a training data of labeled samples. This hyperplane is the one that maximizes the distance between the two classes (the margin). When addressing nonlinear problems, SVM adopts the nonlinear kernels to map data to a higher dimensional space in order to easily find the boundary decision. However, when the dimension is already high; especially the number of features (gene expression data), the nonlinear mapping does not improve the performance. Using the linear kernel is good enough [17, 18, 43].

The following function allows us to classify the new examples:

$$\mathrm{Class}(x) = sgn(w^T x + b).$$

SVM can be adapted to handle multiclass classification problems [15]. In our methods, we use one-against-all strategy.

2.6. K-Nearest Neighbor (KNN)

The K-Nearest Neighbor method is a supervised learning algorithm was first introduced by Fix and Hodges in 1951 [13]. KNN is based on the notion of proximity (Neighbor) between examples, and on the idea of reasoning from similar cases to make a decision [11].

The class of a new example is given by a majority vote of its K neighbors. The K neighbors are determined by the Euclidean distance which is defined as follows:

$$D(X_p, X_q) = \sqrt{\sum_{i=1}^{N} (x_{pi} - x_{qi})^2}.$$

2.7. Fitness function

For each candidate solution X, the quality of X is evaluated by a fitness function f according to two criteria: the ability to obtain a good classification accuracy with this subset of genes and the number of genes contained in this subset. More formally, the fitness value is calculated as follows:

$$f(X) = w_1 * \mathrm{CA}_{\mathrm{Classifier}}(X) + (1 - w_1) * \left(1 - \frac{\#\mathrm{genes}}{p}\right),$$

where w_1 is a weight coefficient in $[0, 1]$ that controls the aggregation of both objectives, #genes the number of selected genes in X and p the dimension of the vector X.

The fitness function seeks to maximize the classification accuracy ($\mathrm{CA}_{\mathrm{Classifier}}$) obtained with a determined classifier constructed on the subset of genes X, calculated by using LOOCV (leave one out cross validation). The second term assures the fact of preferring subsets with less number of genes. our aim is to maximize the function f. And as our main objective is to find a subset of genes with a maximum accuracy, we choose $w_1 = 0.98$.

Mention that "$\mathrm{CA}_{\mathrm{Classifier}}$" is nothing but the average of the r accuracy calculated by the classifier, where accuracy is the percentage of samples correctly classified calculated as follows:

$$\mathrm{accuracy} = \frac{\#\,\text{of correctly predicted samples}}{\#\,\text{of samples}}.$$

2.8. Proposed methods

The originality of our methods stems from the combination of two different approaches Filters (Fisher score and ReliefF) and Wrappers (the BPSO metaheuristic combined with a local search algorithm) to select a small number of highly relevant genes responsible for cancer classification which facilitates early prognosis of the disease. These proposed methods enrich the state of the art of gene selection.

2.8.1. General structure of the proposed approaches

The general procedure of our approaches can be described by a three-step sequential process that uses complementary techniques to gradually reduce the search space and select a relevant subset of genes (Fig. 3).

Step 1: It is a pre-treatment step that has the objective of filtering the irrelevant genes, for example, genes whose expression levels are uniform regardless of the class. Further, there are thousands of genes, then it is inefficient to adopt an evolutionary algorithm such as a binary particle swarm optimization (BPSO) directly in a microarray dataset without filtration. The output of this pre-treatment is a set of genes classified in order of relevance according to a determined score. This is a preliminary step for reducing the size of DNA microarray data. The number of pre-selected genes p was set at 300.

Step 2: This step consists of a Wrapper method, where (BPSO) generates subsets of genes evaluated by a specific classifier, starting from the genes retained by the previous filtering.

Step 3: This step is based on a **Backward** type sequential generation method, applied to the set of genes selected in step 2. the aim of this step is to improve the solutions given by particles and provide good solutions within a reasonable time.

Algorithm 2 shows an outline of the proposed approaches.

Algorithm 2. Proposed Approach.

```
Input: DNA microarray data ; n$_{max}$ ;m ;p ;$it_{max}$; $v_{min}$; $v_{max}$, $w$, $c_1$; $c_2$
Output: $X_{best}$
Begin
Step 1:
Apply a Filter algorithm to choose the p best ranked genes.
Step 2:
k=0;
Randomly initialize the swarm
while $k < n_{max}$ do
    for i = 1 ; i ¡= m ; i++ do
        Evaluate the fitness of the particle $X_i$ by a specific Classifier.
        if (fitness value of $X_i$ is greater than that of $P_i$) then
            $P_i = X_i$
            if (fitness value of $X_i$ is greater than that of $P_{gi}$) then $P_{gi} = X_i$
            end if
        end if
        Calculate the velocity $V_{i+1}$ by using (3)
        for d = 1 ; d ¡ N ; d++ do
            if ($v_{i+1}^d > v_{max}$) then
                $v_{i+1}^d = v_{max}$
            end if
            if ($v_{i+1}^d < v_{min}$) then
                $v_{i+1}^d = v_{min}$
            end if
            if ($Sig(v_{i+1}^d) > random()$) then
                $X_{i+1}^d = 1$
            else$X_{i+1}^d = 0$
            end if
        end for
    end for
    k=k+1
end while
Step 3:
Apply the backward algorithm (Algorithm 1) to $P_{gi}$.
Return $X_{best}$.
```

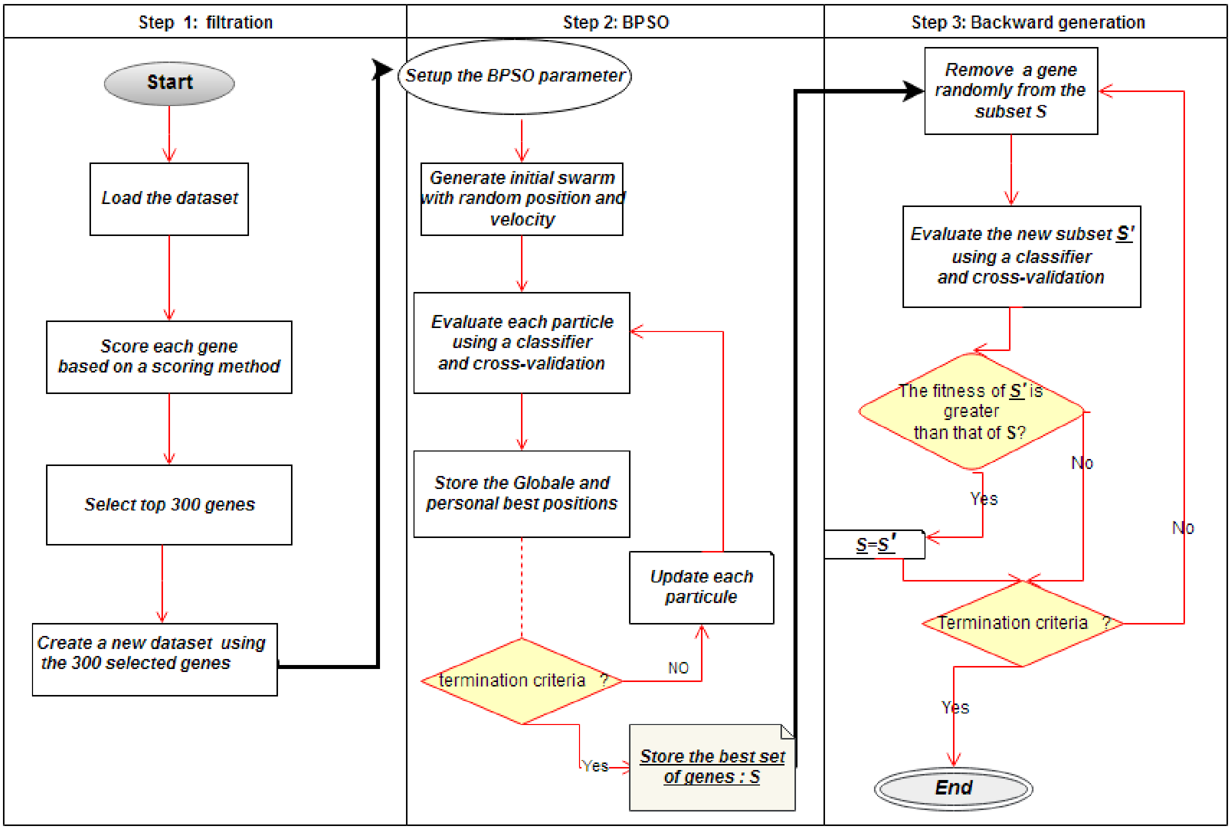
Step 1: filtration
Step 2: BPSO
Step 3: Backward generation
Start
Load the dataset
Score each gene based on a scoring method
Select top 300 genes
Create a new dataset using the 300 selected genes
Setup the BPSO parameter
Generate initial swarm with random position and velocity
Evaluate each particle using a classifier and cross-validation
Store the Globale and personal best positions
Update each particule
termination criteria ?
NO
Yes
Store the best set of genes : S
Remove a gene randomly from the subset S
Evaluate the new subset S' using a classifier and cross-validation
The fitness of S' is greater than that of S?
No
Yes
S=S'
No
Termination criteria ?
Yes
End

FIGURE 3. Flowchart of our proposed approaches for gene subset selection in DNA microarray data.

2.8.2. FBPSO-SVM

Our first proposed method for gene selection (FBPSO-SVM) based on the combination of two approaches for gene selection: the Filter approach through a generalized Fisher score for the multiclass and the approach Wrapper through a binary PSO combined with a heuristic local search algorithm called the backward generation. In this method, the quality of the candidate subset of genes is measured by using a multiclass SVM classifier with a Leave-one-out-cross-validation (LOOCV). The choice of the SVM classifier is justified by its proven robustness for a high-dimensional data.

The following diagram represents the general structure of our first Approach (see Fig. 4 below).

2.8.3. RBPSO-1NN

The second approach (RBPSO-1NN) is derived from the first one (FBPSO-SVM), the only two differences are:

- In the first step (Filter step), the Fisher criterion is replaced by the ReliefF algorithm.
- For the fitness function used in the wrapper approach, the 1NN classifier was chosen, after several experiments it was found that KNN is suitable for data filtered by ReliefF (since both use the notion of neighborhood and distance).

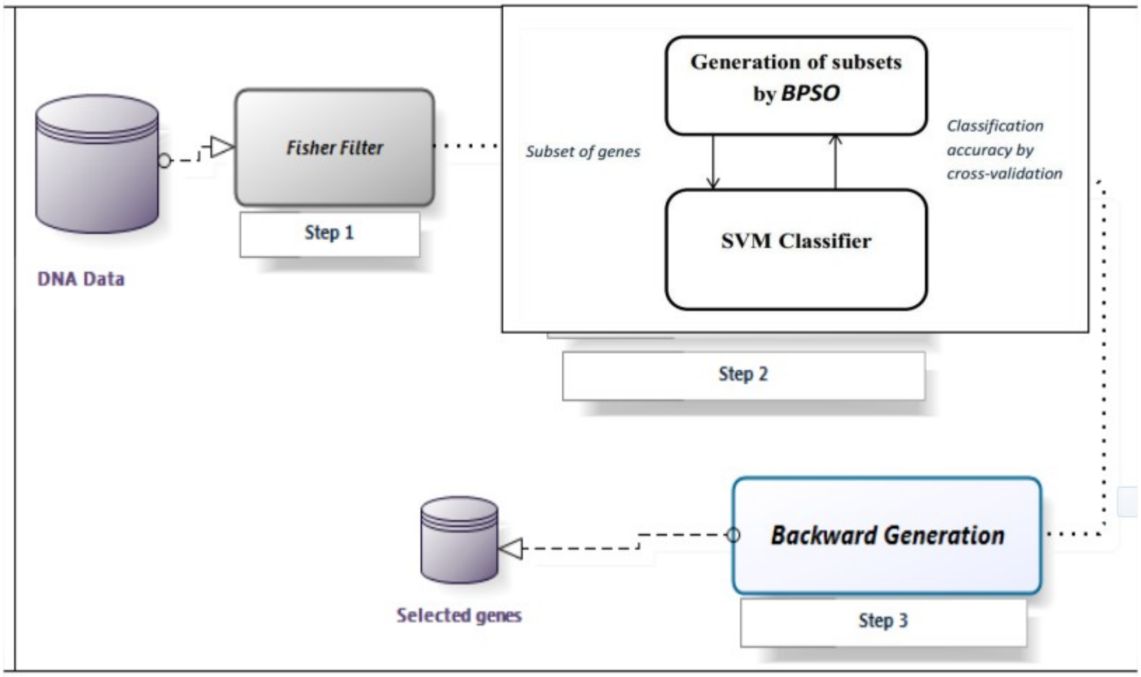

Figure 4. General diagram of FBPSO-SVM.

3. Experiments and results

The implementation of the proposed approaches (FBPSO-SVM) and (RBPSO-1NN) is done under Matlab. As far as the SVM classifier is concerned, we have chosen a predefined binary linear classifier.
In the case where the number of classes exceeds two, we have developed a SVM classifier based on the "One-against-all strategy".
For the KNN classifier and the ReliefF algorithm, we used predefined functions in Matlab.
We chose the datasets from (Gene Expression Model Selector [38]) for experiments.

The rest of this section is devoted to the presentation of the datasets used and the analysis of the experimental results.

3.1. Environment

To evaluate our approaches, we have chosen datasets (DNA microarray), all of which concern problems of recognition of cancers imported since (Gene Expression Model Selector [38]).
We have used publicly available, easily accessible datasets that are used in many data supervised classification work.
The details of the datasets are presented in the Table 2. The choice of the latter depends on a multitude of specificities (binary or multi-classes, number of genes, number of samples). Some datasets have binary classes (Prostate_Tumor, DLBCL) while others have multi-classes (Leukemia1, Lung Cancer ...). Some of them have a smaller number of samples (Brain_Tumor2, 9_Tumors ...), while others have a higher number (Prostate_Tumor ...).
As our approaches are designed to process data of high dimensions all these datasets are characterized by thousands of genes ranging from 2308 to 11225 (high-dimensional data).

Table 2. Description of the datasets (DNA microarray) used.

Dataset name	Diagnostic task	Number		
		of samples	of genes	of classes
9_Tumors	9 various human tumor types	60	5726	9
Brain_Tumor1	5 human brain tumor types	90	5920	5
Brain_Tumor2	4 malignant glioma types	50	10367	4
Leukemia1	Acute myelogenous leukemia (AML), acute lympboblastic leukemia (ALL) B-cell, and ALL T-cell	72	5327	3
Leukemia 2	AML, ALL, and mixed-lineage leukemia (MLL)	72	11225	3
SRBCT	Small, round blue cell tumors (SRBCT) of childhood	83	2308	4
Prostate_Tumor	Prostate tumor and normal tissues	102	10509	2
DLBCL	Diffuse large b-cell lymphomas (DLBCL) and follicular lymphomas	77	5469	2

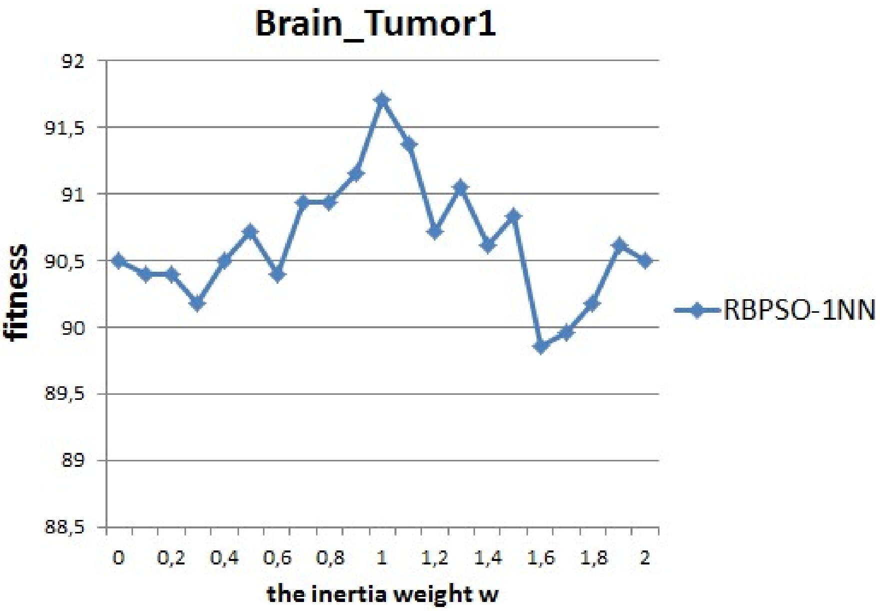

Figure 5. The parametric study for the inertia weight w.

3.2. Parameters

We note that our approaches have been run on an Acer Aspire 7750 g laptop with Intel Core I5 2.30 GHz processor and 8 GB RAM, under system running Windows 7 (64 bit).

As part of the study of our approach, several tests were carried out in order to adopt the best parameterization. More exactly, we selected the best parametrization by setting first, a set of initial values for the parameters, then we adjust the value of one parameter until the solutions could not be improved. We repeat the process until the adjustment of all parameters did not provide a better solution. This process has been tested on two datasets of cancer classification (Brain_tumor1 and Brain_tumor2). As an example for the inertia weight w, the below figure shows that $w = 1$ is the best value of w (Fig. 5):

The number of genes to be selected by the filter approach (step 1) is set at 300.

Table 3. Parameters used for our methods.

Parameters	Value
Swarm size m	30
Number of generations $n_{\max}$	100
c_1, c_2	2
w	1
$v_{\max}$	4
$v_{\min}$	-4
Backward generation, it_{max}	200

The number of neighbors for ReliefF is set to 3. On the other hand, the number of neighbors for the KNN classifier is set to $K = 1$.

The parameters we have set for exploring the wrapper method are shown in the Table 3.

3.3. Results and comparisons

The objects of the experiments carried out on the eight datasets of (DNA microarray) are, on the one hand, to test the effect of gene selection on the improvement of (the classification accuracy and the computational time of classification) and, on the other hand, to show the performances of our approaches.

Given the non-deterministic nature of our approaches, ten independent runs (to compare run times) were performed for each dataset and method.

The Table 4 illustrates the classification rates (LOOCV) for the eight datasets, these rates are obtained from the SVM and 1NN algorithms without selection and then with selection by using our approaches, this table emphasizes the interest of selection in a classification problem.

We analyze our results in four ways:

1. The number of genes selected by the 2 methods.
2. The classification accuracy.
3. The execution time.
4. The computational time of classification before and after the selection.

3.3.1. Discussion

The proposed approaches derive their effectiveness from the remarkable improvement in the classification accuracy (shown in bold in Tab. 4), in all datasets (Fig. 6).

The "RBPSO-1NN" and "FBPSO-SVM" approaches reduce the number of genes compared to the original number of genes in the datasets, we talk about a selection that ensures better accuracy.

In addition, for the "Brain Tumor2" dataset, the "FBPSO-SVM" approach has reached a classification accuracy of 100% using 0.116% of the original genes only (12 genes from 10367). Similarly, for the "Leukimia2" dataset the "RBPSO-1NN" approach gave a perfect accuracy using just 0.044% of the genes (5 genes from 11225).

The results show first that "FBPSO-SVM" and "RBPSO-1NN" perform similar performances for the majority of datasets, with an advantage of FBPSO-SVM at the level of accuracy (in bold Tab. 4), except for "Leukimia2 and SRBCT" where "RBPSO-1NN" is the best according to the number of selected genes and execution time. However the major difference between our two approaches is in "execution time", in fact, "RBPSO-1NN" gives good performance in a minimal time. This is due to the use of 1NN as a classifier (Fig. 7).

In the figures (Figs. 9, 11 and 13), the abscissa axis expresses the number of generations while the ordinate axis represents the classification accuracy of the best particle of the swarm. This is done for the average solution found for the datasets "9_Tumors, Brain_Tumor1 and Brain_Tumor2". These figures clearly show that the use of BPSO plays a crucial role in raising the classification accuracy rate.

Table 4. Comparison of SVM, 1NN, RBPSO-1NN and FBPSO-SVM (LOOCV).

Data sets	Performance	SVM	1NN	RBPSO-1NN		FBPSO-SVM	
				Best	Avg	Best	Avg
9_Tumors	Accuracy (%)	38.33	53.33	83.33	81.83	**95**	92.999
	Genes	5726	5726	20	29.1	71	45
	Time(h)	–	–	0.68	0.69	4.17	4.22
	Class Time(s)	26.38	2.21	0.79	0.9	4.89	5.27
Brain_Tumor1	Accuracy (%)	88.89	86.67	94.44	94.00	**97**,78	97.22
	Genes	5920	5920	11	24.7	21	22.4
	Time(h)	–	–	1.47	1.48	4,09	4,20
	Class Time(s)	23.32	3.32	1.10	1.04	4.81	5.38
Brain_Tumor2	Accuracy (%)	70.00	70.00	96.00	92.80	**100.00**	**100.00**
	Genes	10367	10367	15	24.5	**12**	14.3
	Time(h)	–	–	0.50	0.52	1.84	1.91
	Class Time(s)	14.28	2.29	0.63	0.59	1.72	1.55
Leukimia1	Accuracy (%)	97.22	87.50	**100.00**	99.72	**100.00**	**100.00**
	Genes	5327	5327	8	11.7	**6**	8.4
	Time(h)	–	–	0.76	0.76	1.58	1.60
	Class Time(s)	10.81	1.96	0.9	0.83	1.68	1.54
Leukemia2	Accuracy (%)	97.22	93.05	**100.00**	**100.00**	**100.00**	**100.00**
	Genes	11225	11225	**5**	13.1	6	8.6
	Time(h)	–	–	**0.74**	0.84	1.92	1.99
	Class Time(s)	18.89	3.06	**0.85**	0.87	1.68	1.71
SRBCT	Accuracy%)	**100.00**	91.57	**100.00**	**100.00**	**100.00**	**100.00**
	Genes	2308	2308	**7**	11.7	10	12.4
	Time(h)	–	–	0.87	0.85	2.89	3.02
	Class Time(s)	10.86	2.21	**1.02**	1.06	3.29	3.01
Prostate_Tumor	Accuracy(%)	92.16	76.47	99.02	98.24	**100.00**	**100.00**
	Genes	10509	10509	9	11.2	**6**	8.3
	Time(h)	–	–	1.02	1.01	3.04	3.46
	Class Time(s)	24.48	4.73	1.40	1.29	2.87	2.97
DLBCL	Accuracy (%)	97.40	87.01	**100.00**	**100.00**	**100.00**	**100.00**
	Genes	5469	5469	6	12.5	**4**	6.7
	Time(h)	–	–	1.34	1.30	1.43	1.49
	Class Time(s)	8.34	2.48	0.97	0.97	1.11	1.20

Note: The best results are shown in bold.
Accuracy: The classification accuracy rate using LOOCV (leave one out cross validation).
Genes: The number of genes used in the classification.
Time: The execution time in hours.
Class Time: The computational time of classification in seconds.
Best: The best result found in all ten runs.
Avg: The average of the ten experiments.

In the figures (Figs. 8, 10 and 12), we show the evolution of the number of genes selected on the ordinate axis relative to the number of generations (the abscissa axis) for the "9_Tumors, Brain_Tumor1 and Brain_Tumor2" datasets. These figures illustrate the role of the Backward algorithm in reducing the number of genes.

Moreover, the wrapper method based on the BPSO and the Backward generation algorithm plays a key role in increasing the classification accuracy rate and reducing the number of genes used. Indeed, the BPSO aims to identify the best subset of candidate genes that maximizes the objective function (a compromise between the classification rate and the number of genes used) (Figs. 9, 11 and 13). Once the subset in question is found,

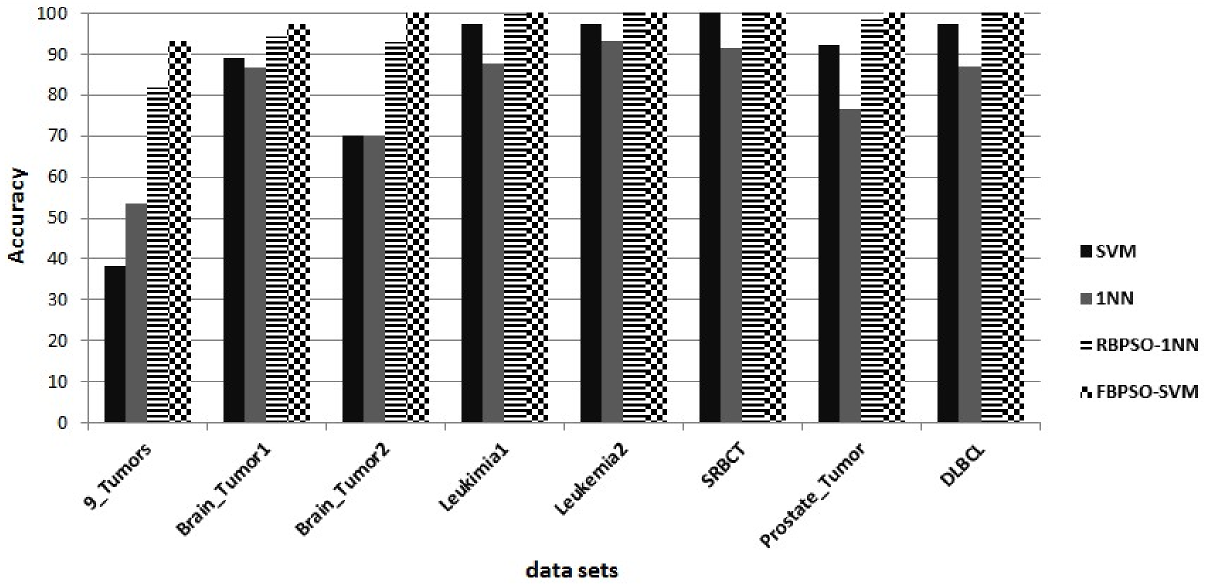

FIGURE 6. Comparison of the average Classification accuracy between the four methods (Best).

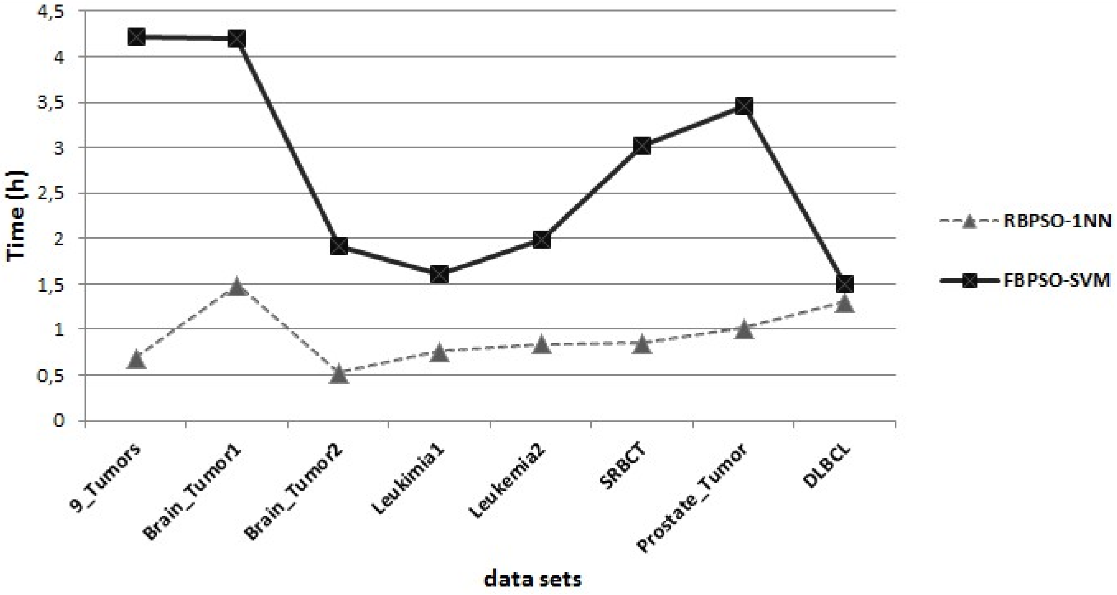

FIGURE 7. Comparison of average execution times in hours.

the Backward algorithm is called to reduce the number of genes used while retaining the classification rate previously fixed (Figs. 8, 10 and 12).

We tried to evaluate our results statistically using the kruskal-wallis statistical test, in order to test the significance of the difference in the results (Classification accuracy) obtained by our approaches.

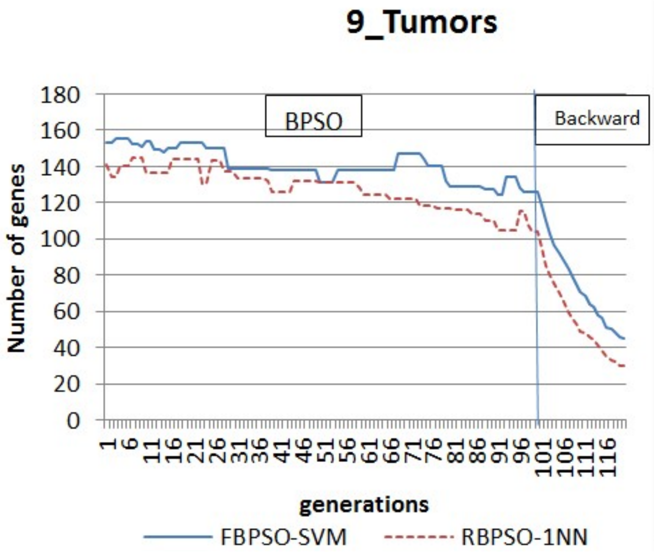

FIGURE 8. Comparison of the evolution of the number of genes used for "9_Tumor1".

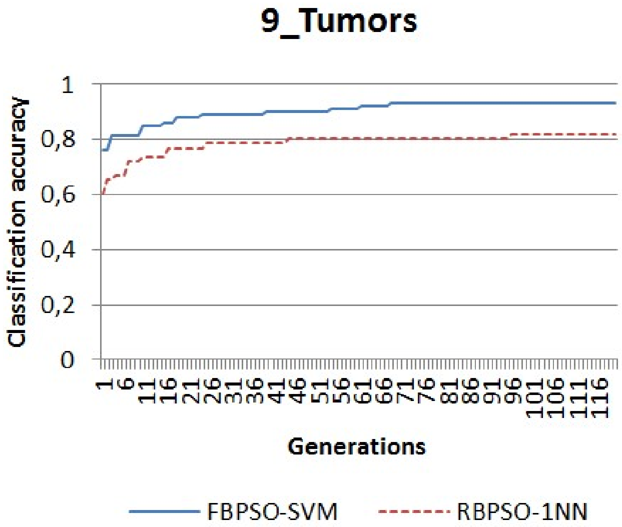

FIGURE 9. Comparison of the evolution of the Classification accuracy rate for "9_Tumors".

The kruskal-wallis statistical test presented in the figure (Fig. 14) shows a comparison of the results obtained by our proposed algorithms, the SVM classifier and the 1NN. According to the figure (Fig. 14), the performance of our proposed approaches exceeds that of SVM and 1NN classifiers. In terms of statistical significance of the results (Classification rate), the kruskal-wallis test shows that the difference between the classification rate in "SVM" and "FBPSO-SVM" is statistically significant (large and remarkable).

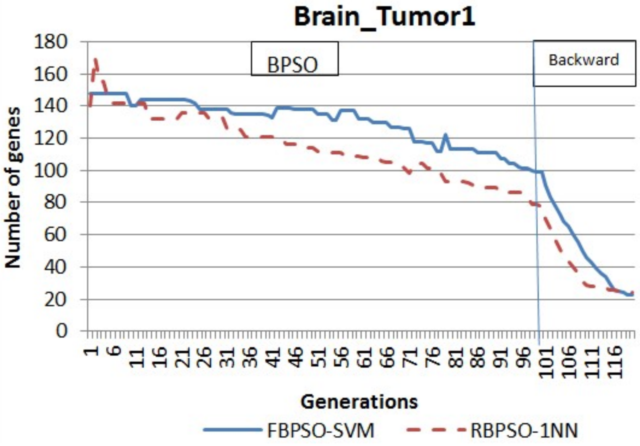

FIGURE 10. Comparison of the evolution of the number of genes used for "Brain_Tumor1".

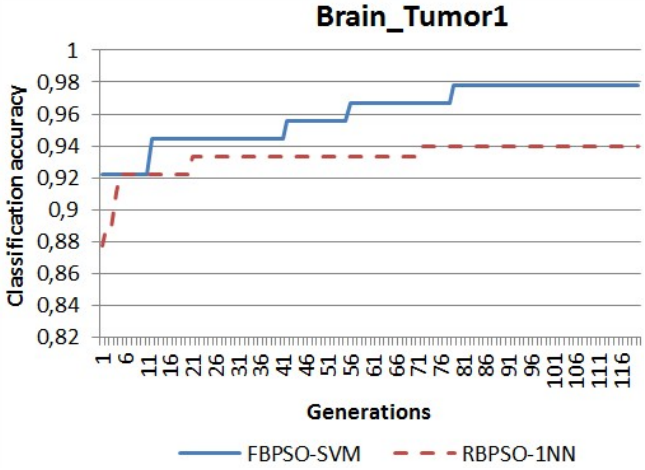

FIGURE 11. Comparison of the evolution of the Classification accuracy rate for "Brain_Tumor1".

Based on these results from the experiments we carried out, we can say that our approaches (FBPSO-SVM and RBPSO-1NN) of gene selection are very well-founded. Indeed, of the eight datasets used, our selection approaches have succeeded in improving the quality of classification. In addition, the FBPSO-SVM approach yielded a classification rate equal to or greater than 95.00% for all datasets, with a perfect 100% classification for (Brain_Tumor2, Leukimia1, Leukimia2, SRBCT, Prostate_Tumor, DLBCL) using less than 12 genes.

In practice, the choice of an optimal subset of genes has been the subject of growing interest in the biological domain, in the sense that it allows biologists to gain time, since they can refer directly to selected genes that have more be useful for the diagnosis of cancer. Moreover, a small subset of genes can reduce the computational time of classification as shown in (Figs. 15 and 16).

In the following, we propose to make a comparison with other optimization algorithms on several benchmark classification datasets.

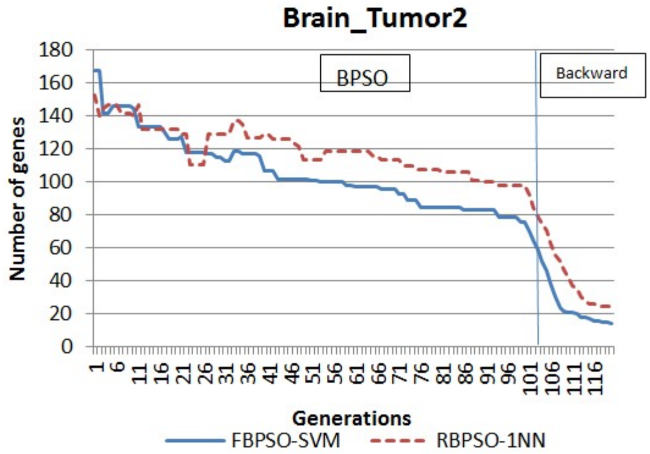

Figure 12. Comparison of the evolution of the number of genes used for "Brain_Tumor2".

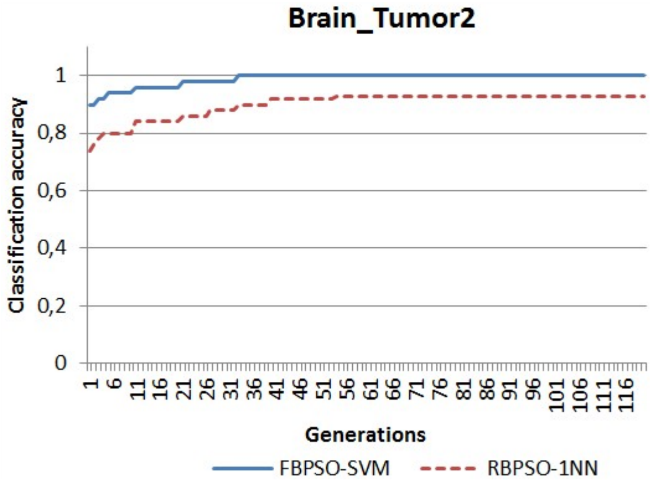

Figure 13. Comparison of the evolution of the Classification accuracy rate for "Brain_Tumor2".

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

In this subsection, we compare our results with those of five reference algorithms [1,8,9,32,41]. To do this, we perform our experiments using the same experimental conditions used by each reference algorithm. Specifically, we execute our approaches ten times, and then we choose the average and the best gene subset found.

Let us take into account that the five reference algorithms use LOOCV to calculate the classification accuracy.

We indicate that the authors of [8,9] report the best results found over 10 runs.

The Table 5 illustrates the classification accuracy for the different approaches as well as the number of selected genes. We use the (–) symbol to designate that a result is not reported in the previous related work. We note that the results obtained by our approaches are very competitive compared to the most representative methods of the last years.

First, for the dataset (9_Tumors) we have a classification accuracy of 95% with only 71 genes. We find that the best performance for this dataset is the performance of our approach (FBPSO-SVM). We note that the

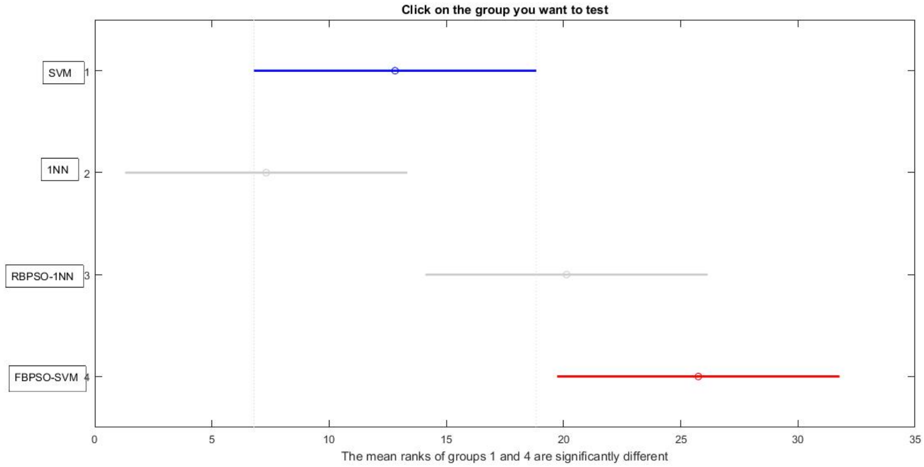

FIGURE 14. The result of the kruskal-wallis test between the proposed approaches, SVM and 1NN on the datasets (classification rate).

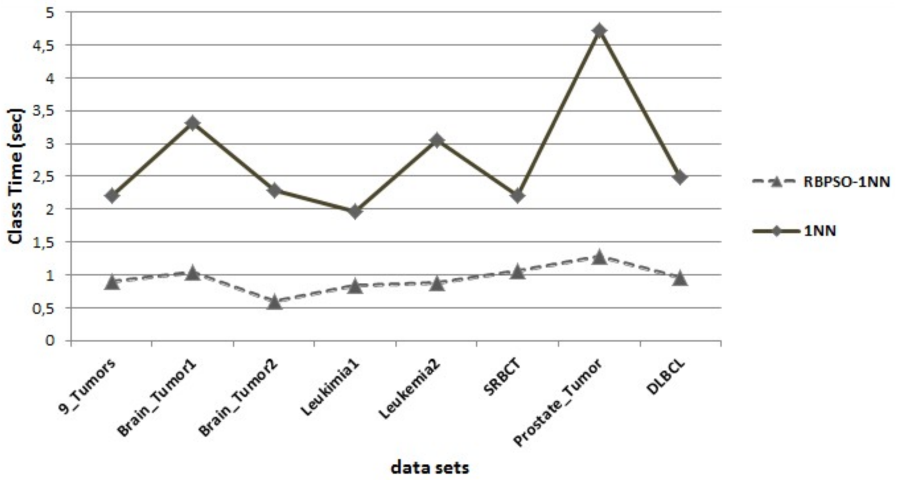

FIGURE 15. Comparison of the computational time of classification for the 1NN classifier before and after the selection using RBPSO.

number of genes reported in other works is at least 251 genes to have a good accuracy and they fail to provide high performance.

Similarly for the datasets (Brain_Tumor1, Brain_Tumor2 and Prostate_Tumor), we get the best performance. Using the approach (FBPSO-SVM), we have a perfect classification (100%) for (Brain_Tumor2 and Prostate_Tumor) with less than 12 genes, and accuracy of 97.78% for (Brain_Tumor1) using only 21 genes. Moreover, our method (FBPSO-SVM) resulted in higher classification accuracies on all datasets compared to the other methods.

The results of this comparative analysis with other proposed methods for the gene selection and the classification of DNA microarray data have enabled us to know just how competitive our approaches are.

Table 5. A comparison between our methods (RBPSO-1NN and FBPSO-SVM) and methods of state-of-the-art.

Datasets	Method	RBPSO-1NN	FBPSO-SVM	FRBPSO	HICATS	EPSO	TS-BPSO	IBPSO
		(This work)	(This work)	2017	(2016)	(2013)	(2009)	(2008)
	Performances			[1]	[41]	[32]	[9]	[8]
9_Tumors	Best #Acc (%)	83.33	**95.00**	–	83.33	76.67	81.63	78.33
	Best #Genes	**20**	71	–	259	251	2941	1280
	Average #Acc (%)	81.83	92.222	–	78.33	75.00	–	–
	Average #Genes	29.1	45	–	248.5	247.10	–	–
Brain_Tumor1	Best #Acc (%)	94.44	**97.78**	–	94.44	93.33	95.89	94.44
	Best #Genes	11	21	–	**6**	8	2913	754
	Average #Acc (%)	94.00	97.22	90.67	93.10	92.11	–	–
	Average #Genes	24.7	22.4	803	8.9	7.5	–	–
Brain_Tumor2	Best #Acc (%)	96.00	**100.00**	–	94.00	94.00	92.65	94.00
	Best #Genes	15	12	–	**3**	4	5086	1197
	Average #Acc (%)	92.80	100.00	87.6	92.60	92.4	–	–
	Average #Genes	24.5	14.3	662	5.8	6.0	–	–
Leukimia1	Best #Acc (%)	**100**	**100.00**	–	**100.00**	**100.00**	**100.00**	**100.00**
	Best #Genes	8	6	–	**3**	2	2577	1034
	Average #Acc (%)	99.72	100.00	98.89	100.00	100.00	–	–
	Average #Genes	11.7	8.4	825	3	3.2	–	–
Leukemia2	Best #Acc (%)	**100**	**100.00**	–	**100.00**	**100.00**	**100.00**	**100.00**
	Best #Genes	5	6	–	5	**4**	5609	1292
	Average #Acc (%)	100.00	100.00	97.50	100.00	100.00	–	–
	Average #Genes	13.1	8.6	1028	6.80	6.8	–	–
SRBCT	Best #Acc (%)	**100.00**	**100.00**	–	**100.00**	**100.00**	**100.00**	**100.00**
	Best #Genes	**7**	10	–	9	**7**	1084	431
	Average #Acc (%)	100.00	100.00	98.19	100.00	99.64	–	–
	Average #Genes	11.7	12.4	213	11.7	14.90	–	–
Prostate_Tumor	Best #Acc (%)	99.02	**100.00**	–	98.04	99.02	95.45	92.61
	Best #Genes	9	6	–	**5**	**5**	5320	1294
	Average #Acc (%)	98.24	100.00	92.43	97.75	97.84	–	–
	Average #Genes	11.2	8.3	418	7.2	6.6	–	–
DLBCL	Best #Acc (%)	**100.00**	**100.00**	–	**100.00**	**100.00**	**100.00**	**100.00**
	Best #Genes	6	4	–	**3**	**3**	2671	1042
	Average #Acc (%)	100.00	100.00	96.49	100.00	100.00	–	–
	Average #Genes	12.5	6.7	105	4.10	4.70	–	–

Note: The best results are shown in bold.
#Acc: Denotes the classification accuracy.
#Genes: Represents the number of selected genes.
Average: Is the average of the ten experiments.
FRBPSO = A Fuzzy Rule Based Binary PSO.
HICATS = Hybrid Binary Imperialist Competition Algorithm and Tabu Search.
EPSO = An enhancement of binary particle swarm optimization.
TS-BPSO = A combination of tabu search and BPSO.
IBPSO = An improved binary PSO.

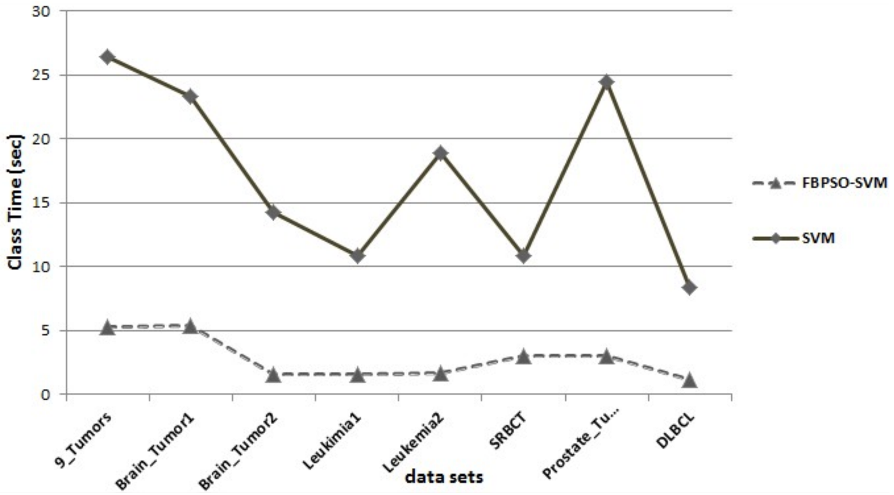

FIGURE 16. Comparison of the computational time of classification for the SVM classifier before and after the selection by FBPSO.

4. Conclusion

In this work, we have presented two hybrid approaches for the gene selection in DNA microarray data. The two proposed approaches consist of a pre-selection phase carried out using the Fisher and ReliefF filter methods for FBPSO-SVM and RBPSO-1NN respectively, and a search phase that determines a good subset of genes for the classification. The latter is based on two metaheuristics, BPSO and a local search method (backward generation). These approaches aim to select an optimal subset of relevant genes from a dataset which contains redundant, noisy or irrelevant data.

The experimental results show that our approaches compare very favorably with the reference methods in terms of the classification accuracy and the number of selected genes.

References

[1] S. Agarwal, R. Rajesh and P. Ranjan, FRBPSO: a Fuzzy rule based binary PSO for feature selection. *Proc. Nat. Acad. Sci. India Sec. A: Phys. Sci.* **87** (2017) 221–233.

[2] E. Alba, J. Garcia-Nieto, L. Jourdan and E.G. Talbi, Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms. In: IEEE Congress on Evolutionary Computation, 2007. CEC 2007. IEEE (2007, September) 284–290.

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

[4] E. Amaldi and V. Kann, On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theor. Comput. Sci.* **209** (1998) 237–260.

[5] J. Apolloni, G. Leguizamón and E. Alba, Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Appl. Soft Comput.* **38** (2016) 922–932.

[6] K.H. Chen, K.J. Wang, K.M. Wang and M.A. Angelia, Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data. *Appl. Soft Comput.* **24** (2014) 773–780.

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

[8] L.Y. Chuang, H.W. Chang, C.J. Tu and C.H. Yang, Improved binary PSO for feature selection using gene expression data. *Comput. Biol. Chem.* **32** (2008) 29–38.

[9] L.Y. Chuang, C.H. Yang and C.H. Yang, Tabu search and binary particle swarm optimization for feature selection using microarray data. *J. Comput. Biol.* **16** (2009) 1689–1703.

[10] C. Cortes and V. Vapnik, Support-vector networks. *Mach. Learn.* **20** (1995) 273–297.

[11] T. Cover and P. Hart, Nearest neighbor pattern classification. *IEEE Trans. Info. Theory* **13** (1967) 21–27.

[12] M. Dashtban, M. Balafar and P. Suravajhala, Gene selection for tumor classification using a novel bio-inspired multi-objective approach. *Genomics* **110** (2018) 10–17.

[13] E. Fix and J.L. Hodges Jr, Discriminatory Analysis-Nonparametric Discrimination: Consistency Properties. California Univ Berkeley, Berkeley (1951).

[14] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, *et al.*, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286** (1999) 531–537.

[15] Y. Guermeur, SVM multiclasses, théorie et applications. Habilitation à diriger des recherches. UHP (2007).

[16] Q. Gu, Z. Li and J. Han, Generalized fisher score for feature selection. Preprint arXiv: 1202.3725 (2012).

[17] C.W. Hsu, C.C. Chang and C.J. Lin, A practical guide to support vector classification. Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf (2003).

[18] H.Y. Huang and C.J. Lin, Linear and kernel classification: when to use which? In: *Proc. of the 2016 SIAM International Conference on Data Mining.* Society for Industrial and Applied Mathematics (2016) 216–224.

[19] P. Jafari and F. Azuaje, An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors. *BMC Med. Info. Decis. Mak.* **6** (2006) 27.

[20] J. Kennedy and R. Eberhart, PSO optimization. In: *Proc. IEEE Int. Conf. Neural Networks.* IEEE Service Center, Piscataway, NJ **4** (1995) 1941–1948.

[21] J. Kennedy and R.C. Eberhart, A discrete binary version of the particle swarm algorithm. In: Systems, Man, and Cybernetics, 1997. *IEEE International Conference on Computational Cybernetics and Simulation.* IEEE **5** (1997) 4104–4108.

[22] K. Kira and L.A. Rendell, A practical approach to feature selection. In: *Proc. of the Ninth International Workshop on Machine Learning.* Morgan Kaufmann Publishers Inc., San Francisco (1992) 249–256.

[23] R. Kohavi and G.H. John, Wrappers for features subset selection. *Artif. Intell.* **97** (1997) 273–324.

[24] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF. In: *European Conference on Machine Learning.* Springer, Berlin, Heidelberg (1994) 171–182.

[25] B. Kumari and T. Swarnkar, Filter versus wrapper feature subset selection in large dimensionality micro array: a review. *Int. J. Comput. Sci. Inf. Technol.* **2** (2011) 1048–1053.

[26] C.M. Lai, W.C. Yeh and C.Y. Chang, Gene selection using information gain and improved simplified swarm optimization. *Neurocomputing* **218** (2016) 331–338.

[27] C.P. Lee and Y. Leu, A novel hybrid feature selection method for microarray data analysis. *Appl. Soft Comput.* **11** (2011) 208–213.

[28] Y. Li, G. Wang, H. Chen, L. Shi and L. Qin, An ant colony optimization based dimension reduction method for high-dimensional datasets. *J. Bionic Eng.* **10** (2013) 231–241.

[29] S. Li, X. Wu and M. Tan, Gene selection using hybrid particle swarm optimization and genetic algorithm. *Soft Comput.* **12** (2008) 1039–1048.

[30] H. Liu and H. Motoda, Feature selection for knowledge discovery and data mining. In Vol. 454. Springer Science Business Media (2012).

[31] D. Mishra and B. Sahu, Feature selection for cancer classification: a signal-to-noise ratio approach. *Int. J. Sci. Eng. Res.* **2** (2011) 1–7.

[32] M.S. Mohamad, S. Omatu, S. Deris, M. Yoshioka, A. Abdullah and Z. Ibrahim, An enhancement of binary particle swarm optimization for gene selection in classifying cancer classes. *Algorithm Mol. Biol.* **8** (2013) 15.

[33] S.K. Pati, A.K. Das and A. Ghosh, Gene selection using multi-objective genetic algorithm integrating cellular automata and rough set theory. In: *International Conference on Swarm, Evolutionary, and Memetic Computing.* Springer, Cham (2013) 144–155.

[34] A.C. Pease, D. Solas, E.J. Sullivan, M.T. Cronin, C.P. Holmes and S.P. Fodor, Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proc. Nat. Acad. Sci.* **91** (1994) 5022–5026.

[35] J.C. Platt, N. Cristianini and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In *Proc. of Advances in neural information processing systems* (2000) 547–553.

[36] F.V. Sharbaf, S. Mosafer and M.H. Moattar, A hybrid gene selection approach for microarray data classification using cellular learning automata and ant colony optimization. *Genomics* **107** (2016) 231–238.

[37] S.S. Shreem, S. Abdullah, M.Z.A. Nazri and M. Alzaqebah, Hybridizing ReliefF, MRMR filters and GA wrapper approaches for gene selection. *J. Theor. Appl. Inf. Technol.* **46** (2012) 1034–1039.

[38] A. Statnikov, C. Aliferis and I. Tsamardinos, Gems: Gene Expression Model Selector. Available at: http://www.gems-system.org (2005).

[39] S. Tabakhi, A. Najafi, R. Ranjbar and P. Moradi, Gene selection for microarray data classification using a novel ant colony optimization. *Neurocomputing* **168** (2015) 1024–1036.

[40] Z. Wang, Neuro-fuzzy modeling for microarray cancer gene expression data. First year transfer report. University of Oxford (2005).

[41] S. Wang, W. Kong, W. Zeng and X. Hong, Hybrid binary imperialist competition algorithm and tabu search approach for feature selection using gene expression data. *BioMed Res. Int.* **2016** (2016). 9721713.

[42] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, *et al.*, Top 10 algorithms in data mining. *Knowl. Info. Syst.* **14** (2008) 1–37.

[43] G.X. Yuan, C.H. Ho and C.J. Lin, Recent advances of large-scale linear classification. *Proc. IEEE* **100** (2012) 2584–2603.

[44] H. Yu, G. Gu, H. Liu, J. Shen and J. Zhao, A modified ant colony optimization algorithm for tumor marker gene selection. *Genomics Proteomics Bioinf.* **7** (2009) 200–208.

[45] W. Zhao, G. Wang, H.B. Wang, H.L. Chen, H. Dong and Z.D. Zhao, A novel framework for gene selection. *Int. J. Adv. Comput. Technol.* **3** (2011) 184–191.

[46] A. Zibakhsh and M.S. Abadeh, Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function. *Eng. App. Artif. Intell.* **26** (2013) 1274–1281.