Neurocomputing 168 (2015) 1024–1036

Contents lists available at ScienceDirect

Neurocomputing

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

Gene selection for microarray data classification using a novel ant colony optimization

Sina Tabakhi [a], Ali Najafi [b,*], Reza Ranjbar [b], Parham Moradi [a]

[a] Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran
[b] Molecular Biology Research Center, Baqiyatallah University of Medical Sciences, Tehran, Iran

ARTICLE INFO

Article history:
Received 1 November 2014
Received in revised form
27 February 2015
Accepted 7 May 2015
Communicated by A.M. Alimi
Available online 14 May 2015

Keywords:
Gene selection
Feature selection
Microarray data classification
Filter approach
Ant colony optimization
Pattern recognition

ABSTRACT

The high-dimensionality of microarray data with small number of samples has presented a difficult challenge for the microarray data classification task. The aim of gene selection is to reduce the dimensionality of microarray data in order to enhance the accuracy of the classification task. Existing gene selection methods generally use class labels of the data while due to availability of mislabels or unreliable labels of samples in the microarray data, unsupervised methods could be more essential to the gene selection process. In this paper, we propose an unsupervised gene selection method called MGSACO, which incorporates the ant colony optimization algorithm into the filter approach, by minimizing the redundancy between genes and maximizing the relevance of genes. Moreover, a new fitness function is applied in the proposed method which does not need any learning model to evaluate the subsets of selected genes. Thus, it is classified into the filter approach. The performance of the proposed method is extensively tested upon five publicly available microarray datasets, and it is compared to those of the seven well-known unsupervised and supervised gene selection methods in terms of classification error rates of the three frequently used classifiers including support vector machine, naïve Bayes, and decision tree. Experimental results show that MGSACO is significantly superior to the existing methods over different classifiers and datasets.

1. Introduction

With the rapid advances of the DNA microarray technology in recent decades, biology researchers are now able to monitor thousands of gene expression levels simultaneously in a single experiment that could be useful for detection or classification of the specific tumor type. Microarray data classification is concerned with the extraction of valuable information through the use of machine learning and data mining techniques in which a model is built to classify the samples into different categories. The high-dimensional nature of microarray data with small number of samples presents a well-known phenomenon named “curse of dimensionality” [1]. However, several studies show that most genes do not contain the relevant information for the classification task; thus, data preprocessing is an essential effort to attain efficient, accurate, and reliable performance in the classification of microarray data [2–9]. Gene selection is a common technique in microarray data preprocessing. This technique is a process of identifying a subset of informative genes from the original gene set, which decreases the computational costs as well as improves the classification performance [10–14].

The gene selection methods broadly fall into four categories including filter, wrapper, embedded, and hybrid approaches [4,7,11,15,16]. The filter approach assesses the relevance of genes based on statistical properties of the data without using any learning model. Up to now, different strategies have been proposed to evaluate the relevance of genes including univariate and multivariate strategies [3,13,15,17]. The univariate strategy, at first, evaluates and ranks the genes individually using a given criterion, and then the subset of genes with the highest rank values is selected as the final subset. Several criteria have been used in the literature such as Laplacian score [2,18], term variance [1], Signal-to-noise ratio [19], mutual information [14], and information gain [20]. The univariate-based methods are fast and efficient but may lead to lower classification accuracy due to ignorance of dependencies between genes. On the other hand, the multivariate strategy attempts to tackle the problem of univariate strategy by considering the correlation between genes. Therefore, the performance of the multivariate-based methods is better than that of the univariate-based methods. Several methods based on multivariate strategy have been proposed in the literature including mRMR [5,21], FCBF [22,23], RRFS [24,25], UFSACO [13], RSM [26–28], and mutual correlation [29,30]. Often, the search strategies applied in

* Corresponding author. Tel.: +98 21 82482548.
E-mail addresses: sina.tabakhi@ieee.org (S. Tabakhi), najafi74@ibb.ut.ac.ir (A. Najafi), ranjbar@bmsu.ac.ir (R. Ranjbar), p.moradi@uok.ac.ir (P. Moradi).

http://dx.doi.org/10.1016/j.neucom.2015.05.022
0925-2312/

the multivariate-based methods seek a subset of genes in a single iteration; thus, these methods can easily be trapped into the local optimum.

The wrapper approach applies a specific learning model in the gene selection process to evaluate a subset of selected genes iteratively, and then the accuracy of the learning model is used to guild the search process. This approach can be classified into greedy and stochastic search strategy [15,31]. The greedy search strategy is a single-track search and thus can easily be trapped in the local optimum. Sequential forward selection and sequential backward selection are two basic methods used in the greedy search strategy [16,32]. On the other hand, the stochastic search strategy uses the randomness nature in its gene selection process. Some of the methods based on the stochastic search strategy include ant colony optimization (ACO) [11,33,34], particle swarm optimization (PSO) [35,36], genetic algorithm (GA) [37,38] and the firefly algorithm [39]. Since the wrapper approach contains a given learning model and also considers the interaction between subsets of genes, their performance is better than that of the filter approach, but this approach suffers from high computational cost especially for high-dimensionality of microarray datasets.

In the embedded approach a specific learning model is trained using an initial gene set to establish a criterion to measure the rank values of genes. Examples of the embedded based methods include support vector machine based on recursive feature elimination (SVM-RFE) [40], random forest [41], and the first order inductive learner (FOIL) rule based feature subset selection algorithm [42]. The main advantage of the embedded approach is the interaction with the learning model, but training a given classifier with the full gene set is time-consuming especially due to the high-dimensionality of microarray datasets.

The hybrid approach has been proposed to take the advantages of the filter and the wrapper approaches. In the hybrid approach at first a subset of genes is selected based on the filter approach and then the wrapper approach is employed to choose the final gene set. Consequently, the wrapper approach encounters reduced size genes, and its computation requirement becomes acceptable. Chi-square statistics and a GA [6], information gain and a memetic algorithm [43], Fisher score with a GA and PSO [44], and the multiple-filter-multiple-wrapper (MFMW) method [7] have been presented based on the hybrid approach. The major disadvantage of the hybrid approach is that the filter and the wrapper approaches are not truly integrated with each other, which may lead to lower classification performance.

Swarm intelligence-based methods such as ACO and PSO are multi-agent systems with the collective behavior of a population of artificial agents. These methods have been successfully applied for solving the feature selection problem in many fields such as face recognition [45], text classification [46], and financial domains [47]. Although most of these methods are classifier-based approaches (*i.e.*, wrapper and hybrid approaches) and have obtained acceptable performance for feature selection in different fields, they have not been frequently used in the DNA microarray area due to high computation consumption. Consequently, the filter approach has attracted a lot of attention in the microarray sample classification problem.

Since class labels of the microarray data are available, most of the proposed gene selection methods in the literature consider these labels in their search processes which are known as supervised gene selection methods [5,14,19,20,22,42]. However, in some of the microarray data, there are samples that are incorrectly labeled or whose class labels may be unreliable [4,10]. Therefore, these problems show the significance of using the unsupervised gene selection methods that have been neglected in the DNA microarray field.

Thus, the aim of the current study is to design a framework to combine the computational efficiency of the filter approach and the good performance of the ACO algorithm, in which the learning model and the class labels of the sample are not needed in the gene selection process. In this paper, we propose a new unsupervised filter based gene selection method for microarray data classification called microarray gene selection based on ant colony optimization (MGSACO). The proposed method is an iterative improvement process where at each iteration a population of agents selects a subset of genes. Then, the performance of the found subsets of genes is evaluated using a new proposed fitness function without using any learning model. Finally, the best subset of genes in all iterations is chosen as the final gene set. The proposed method also uses the relevance and redundancy analyses in the gene selection process to identify the irrelevant and redundant genes.

The remainder of the paper is organized as follows. Section 2 briefly reviews the ant colony optimization algorithm. Section 3 presents the proposed gene selection method using the ACO algorithm. Section 4 provides the experimental results on five microarray datasets. Section 5 presents the statistical analysis and discussion. Finally, Section 6 demonstrates the conclusion and directions for future research.

2. Background

This section briefly introduces the ant colony optimization (ACO) and its strengths. Then, the existing feature selection methods based on the ACO will be reviewed.

2.1. Ant colony optimization

ACO is one of the newly developed forms of swarm intelligence methods which were initially introduced by Dorigo et al. to solve various hard combinatorial optimization problems [48]. ACO was biologically inspired by the food-seeking behavior of real ants. While the ants are walking, they deposit a chemical material, called pheromone, on the ground. The intensity of the deposited pheromone depends on the distance between the nest and the food source. The lower paths get larger amounts of pheromone. When the new ants come into the system, they prefer in probability to choose the path with the greater amount of pheromone. After a period of time, with a positive feedback effect, all ants will be choosing the lower paths [49,50]. This structure can be applied to solve different problems. The main characteristics of the ACO algorithm include [13,48–52]: (1) the use of a colony of ants increases the robustness of the algorithm (*i.e.*, it is a population-based method), (2) the collective interaction between the ants can efficiently solve a problem (*i.e.*, it is a multi-agent system), (3) the greedy and stochastic natures of the algorithm increase the local and global search abilities, (4) a structure similar to the reinforcement learning scheme, and (5) distributed computing due to the inherent parallelism.

2.2. Ant colony optimization for feature selection

Recently, the ACO algorithm has obtained more attention for solving the feature selection problem in many areas. The feature selection for text classification using ACO was developed in many literature [46,53,54]. These studies have modeled the states of the problem as a graph and iteratively employed a specific classifier to evaluate the quality of subsets of selected features and used its performance to guide the search process. Kashef and Nezamabadi-pour [55] used binary ACO to select subsets of features for

classification problems. They constructed a graph model from features in which each node had two sub-nodes for selecting or deselecting features. Their method visited all features and could decide to select or deselect a feature. Kanan and Faez [45] proposed an improved feature selection method based on the ACO for face recognition system. Their proposed method used classifier performance and the length of the selected feature in the ACO algorithm. Chen et al. [56] presented an ACO based feature selection method for image classification. Nemati and Basiri [57] developed a feature selection method for text-independent speaker verification system based on the ACO. Marinakis et al. [47] introduced a framework for selection of a set of appropriate features based on a combination of ACO and PSO algorithms in financial classification problems. They used three different classifiers for the classification problem based on the K-nearest neighbor. Vieira et al. [58] proposed a hybridized method for feature selection using the ACO algorithm and fuzzy classifiers for minimizing the number of features and the classification error.

Several methods have been proposed for gene selection via the ACO algorithm in DNA microarray data classification. Li et al. [11] presented a two-stage dimensional reduction method based on the ACO algorithm in which at the first stage, irrelevant genes are removed from a gene set using a modified ant system, and in the second stage, the final gene set is selected using an improved ant colony system. Yu et al. [34] proposed a modified ACO algorithm for extracted tumor-related marker genes and applied the SVM classifier in the search process of ACO to evaluate the performance of the subset of selected genes. Nemati et al. [59] developed a hybrid method for feature selection in prediction of the postsynaptic activity of proteins using the ACO and GA algorithms. Tabakhi et al. [13] introduced a new unsupervised feature selection method based on the ACO algorithm. The hybrid of the ACO algorithm and the neural network classifier is proposed by Kabir et al. to select a subset of salient features [33].

3. Proposed method

In this section, we present a new gene selection method based on ant colony optimization, called MGSACO, for microarray data classification. Section 3.1 describes the details of the proposed method and its computational complexity has been analyzed in Section 3.2.

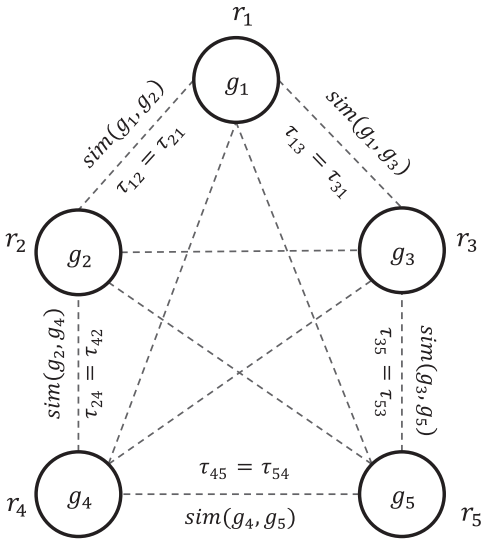

Fig. 1. The graph representation for the gene selection problem. $sim(g_i,g_j)$ is a similarity value assigned to the edge (g_i,g_j) in which $sim(g_i,g_j)=sim(g_j,g_i)$, r_i indicates the relevance of gene g_i, and τ_{ij} denotes the pheromone value on the edge (g_i,g_j).

3.1. Microarray gene selection based on ant colony optimization

In this section, the representation of the search space, the mechanism of the MGSACO method, state transition and pheromone updating rules, and the fitness function have been presented.

3.1.1. Representation of the search space

In the proposed method, the search space is represented as a fully connected weighted graph where the nodes in the graph denote the original genes set, and the graph edges indicate the relationship between the genes. Additionally, the weight of the edge between genes g_i and g_j, $\forall i,j=1\ldots n$ is set to the similarity value between them, that is defined as follows:

$$sim(g_i,g_j)=\left|\frac{g_i g_j}{\|g_i\|\,\|g_j\|}\right|=\left|\frac{\sum_{s=1}^{p} g_{is}g_{js}}{\sqrt{\sum_{s=1}^{p} g_{is}^2}\sqrt{\sum_{s=1}^{p} g_{js}^2}}\right| \quad (1)$$

where p is the number of samples and g_{is} denotes the value of gene i for sample s. According to Eq. (1) it can be concluded that $0\le sim(g_i,g_j)\le 1$, where the value 0 means that the two genes are completely non-similar while the value 1 means that the two genes are completely similar.

Moreover, the proposed method employs the search strategy of the ACO algorithm to solve the gene selection problem. Thus, "*heuristic information*" and "*desirability*" must be defined. The heuristic information is referred to as the prior knowledge about the problem which can guide the ants toward the promising solutions and the desirability, which is also known as pheromone, reflects the information obtained from the past experiences of the ants. In the proposed method, the inverse of the similarity between genes is used as the heuristic information which is assigned to the graph edges. Besides, the pheromone values τ_{ij}, $\forall i,j=1\ldots n$ are associated with the edges in the graph and will be updated by the ants during the search process. Fig. 1 illustrates this representation of the search space.

3.1.2. Mechanism of the proposed gene selection method

The framework of the proposed gene selection method based on ACO is shown in Algorithm 1.

Algorithm 1. Microarray gene selection method based on Ant Colony Optimization (MGSACO)

Input

$D: p\times n$ matrix, p patterns of a n dimensional training set.
$m(\le n)$: the number of genes for the final reduced gene set.
I: the maximum number of allowed iterations.
A: the number of ants.
NG: the number of genes selected by ants in each iteration.

Output

$\tilde{D}: p\times m$ matrix, reduced dimensional training set.

Begin

1. Compute the similarity $sim(g_i,g_j)$ between pairs of genes, $\forall i,j=1\ldots n$.
2. Compute the relevance r_i of genes, $\forall i=1\ldots n$.
3. Set the initial intensity of pheromone $\tau_{ij}(1)$ on the edges to constant value c, $\forall i,j=1\ldots n$.
4. **for** $t=1$ to I **do**
5. Initialize the edge counter $EC[i,j]$ to zero, $\forall i,j=1\ldots n$.
6. Place the ants randomly on the nodes in the graph.
7. **for** $i=1$ to NG **do**

8. **for** $k=1$ to A **do**
9. Choose the next gene among unvisited genes according to the state transition rule.
10. Move the kth ant to the new selected gene.
11. Increment edge counter corresponding to the visited edge.
12. **end for**
13. **end for**
14. Evaluate the candidate subsets of selected genes using fitness function.
15. Find global best solution.
16. Update pheromone values by applying the pheromone updating rule.
17. **end for**
18. Keep the global best solution in all iterations.
19. Build $\tilde{D}$ from D based on the global best solution.

End

The proposed method is composed of two main parts including the initialization part and the gene selection part. In the initialization part (lines 1–3), the similarity values between genes are calculated and assigned to the edges in the graph. Then, the initial intensity of pheromone values on the edges is set to a constant value c. Finally, the relevance of each gene is simply evaluated using the term variance criterion [1].

The gene selection part (lines 4–19) is an iterative improvement process. Each iteration of this part consists of several steps as follows:

Step 1: The edge counter (EC) matrix is defined to count the number of times that a specific edge between two genes is visited by ants, and their initial values are set to zero. Additionally, A ants are randomly placed on the graph nodes as their starting nodes.

Step 2: Each ant constructs a candidate solution by iteratively adding a gene to the current selected gene subset according to a "*state transition rule*" which is a combination of the heuristic information and the pheromone values (see Section 3.1.3 for details). An ant prefers to visit an edge with low similarity to its previously selected gene as well as high intensity of pheromone values. When a given edge is visited by the ant, its corresponding edge counter (*i.e.*, $EC[i, j]$) is increased. This step continues until a given number of genes are selected by each ant.

Step 3: The candidate subsets of genes are evaluated using a new proposed fitness function (see Section 3.1.5 for details). Then, the subset of genes with the better fitness value is kept as the best result in the current iteration.

Step 4: The intensity of pheromone values on the edges are updated according to a "*pheromone updating rule*". In other words, a fraction of pheromone values on each edge is evaporated; the edges with greater EC values get the greater amount of pheromone; and all ants deposit an amount of pheromone on the edges which belong to the fitness of their selected gene subsets (see Section 3.1.4 for details).

This process continues until the maximum number of iterations I is reached. Finally, the global best subset of genes in all iterations is chosen as the final subset of genes.

3.1.3. State transition rule

In the proposed method, each ant chooses the next node by applying either greedy or probabilistic rules that is a combination of the heuristic information and the intensity of pheromone on the edges.

In the greedy rule, the kth ant positioned on gene i selects the next gene j according to the following formula:

$$j = \underset{u \in J_i^k}{\arg\max}\left\{[\tau_{iu}][\eta(g_i,g_u)]^{\beta}\right\}, \quad \text{if } q \le q_0 \tag{2}$$

where J_i^k is the unvisited gene set, τ_{iu} is the pheromone value on the edge (g_i, g_u), $\eta(g_i, g_u) = 1/sim(g_i, g_u)$ is heuristic information which was chosen to be the inverse of the similarity value between genes g_i and g_u, parameter β controls the relative influence of the pheromone value versus heuristic information ($\beta > 0$), q is a random number uniformly distributed in [0. 1], and q_0 is a parameter ($0 \le q_0 \le 1$).

In the probabilistic rule, the kth ant selects the next gene j with a probability $P_k(i,j)$ as follows:

$$P_k(i,j) = \begin{cases} \frac{[\tau_{ij}][\eta(g_i,g_j)]^{\beta}}{\sum_{u \in J_i^k}[\tau_{iu}][\eta(g_i,g_u)]^{\beta}} & \text{if } j \in J_i^k \quad \text{if } q > q_0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The probabilistic rule (Eq. (3)) allows the ants to build a variety of different solutions in order to explore a larger solution space, while the greedy rule (Eq. (2)) has the strong local search ability.

3.1.4. Pheromone updating rule

The pheromone updating rule is performed on all edges after each ant has completed its traverse. The pheromone values are updated by applying the following formula:

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \frac{EC[i,j]}{\sum_{u,v=1,\ldots,n} EC[u,v]} + \sum_{k=1}^{A} \Delta\tau_{ij}^k(t) \tag{4}$$

where

$$\Delta\tau_{ij}^k(t) = \begin{cases} fitness(k) & \text{if } (i,j) \in subset(k) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

ρ is an evaporation rate parameter ($0 < \rho < 1$), $\tau_{ij}(t)$ and $\tau_{ij}(t+1)$ show the pheromone values on edge (g_i, g_j) at time t and $t+1$, respectively, n is the number of original genes, $EC[i,j]$ denotes the number of times that edge (g_i, g_j) is visited by the ants, and A is the number of ants. In Eq. (5), $fitness(k)$ is a fitness function which determines the quality of the solution found by ant k, and $subset(k)$ indicates the subset of genes selected by ant k.

There are two strategies in the pheromone updating rule that are used to make the components of the good solutions (*i.e.*, the best gene subsets) more attractive to the ants in the next iterations. The first strategy is pheromone evaporation in which the intensity of pheromone deposited by the previous ants decreases over time. The aim is to avoid a too rapid convergence of the algorithm toward local optimum areas in the search space. The second strategy is pheromone deposit in which the amount of pheromone increases so that the components belonging to the good solutions will receive the greater amount of pheromone. The aim is to make the components of these solutions more desirable for the ants to explore the optimal areas in the search space in the further iterations.

3.1.5. Fitness function

The main goal of the proposed method is to use the relevance and redundancy analyses in the gene selection process. To this end, the ants seek gene subsets with minimum redundancy between genes. Thereafter, subsets with maximum relevance should get a greater fitness value.

The fitness value of solution k is computed as follows:

$$fitness(k) = \frac{1}{|subset(k)|} \sum_{i=1}^{|subset(k)|} relevance(g_i^k) \quad (6)$$

where $subset(k)$ is the subset of genes selected by ant k, $|subset(k)|$ is the size of $subset(k)$, g_i^k is the ith gene in $subset(k)$, and *relevance* is a function that evaluates the relevance of each gene. In this paper the term variance [1] is used as a relevance function, which is defined as follows:

$$TV(g_i) = \frac{1}{p} \sum_{s=1}^{p} (g_{is} - \bar{g}_i)^2 \quad (7)$$

where p is the number of samples, g_{is} denotes the value of gene i for sample s, and $\bar{g}_i$ is the average value of all the samples corresponding to gene g_i. Also, the relevance value of each gene is normalized in the interval [0..1] using the softmax scaling function [1]. Note that the number of selected genes by ants in each iterations is equal to a constant value NG. It can be seen from Eq. (6) that this specific kind of fitness function is independent of any learning model.

3.2. Computational complexity analysis

In this section, the computational complexity of MGSACO as time complexity and space complexity has been analyzed.

Time complexity can be estimated based on the pseudo code of the proposed method (*i.e.*, Algorithm 1) as follows:

- Evaluating the similarity values between all pairs of genes (line 1) has a computational cost of $O(n^2p)$, where n is the number of genes and p is the number of samples.
- Computing the relevance values of all genes using term variance (line 2) needs $O(np)$.
- Finding the best subset of genes (lines 4–17) is an iterative improvement process. Each iterations of this part consists of four steps. In the first step, construction of candidate solutions by ants (lines 6–13) requires $O(NGnA)$, where NG is the number of genes selected by ants in each iteration and A is the number of ants. Moreover, in the second step (line 14), the candidate solutions are evaluated using the fitness function, which has a complexity of $O(A\,NG)$. In the third step, finding the best solution needs $O(A)$, and finally, in the fourth step, updating the pheromone values has $O(n^2A)$. The overall computational complexity of this part is $O(I(NGnA+(A\,NG\ +A+n^2A)))=O(In^2A)$, where I is the number of iterations.

Consequently, the total time complexity of the proposed method is $O(n^2p+np+In^2A)=O(n^2p+In^2A)$. Since the number of samples in microarray data is small (*i.e.*, IA≫F), the time complexity can be reduced approximately to $O(In^2A)$.

It should be noted that finding the best subset of genes in the proposed method is performed in the iteration improvement process. Therefore, the time complexity of the proposed method is slower than those of the filter-based methods. On the other hand, the proposed method does not need any classifiers in the gene selection process. Thus, its time complexity is faster than those of the wrapper-based methods.

Space complexity can be estimated as follows:

- The similarity values between all pairs of genes are kept in a matrix which needs the space complexity $S(n^2)$.
- The relevance values need the space complexity $S(n)$.
- Candidate solutions of the ants have a complexity of $S(A\,NG)$.
- The pheromone values are saved in a matrix with a cost of $S(n^2)$.
- For the edge counter matrix (*i.e.*, EC), space complexity is $S(n^2)$.
- The best subset of genes is kept in an array with a cost of $S(NG)$.

Therefore, the overall space complexity of the proposed method is $S(n^2+n+A\,NG+n^2+n^2+NG)=S(n^2)$.

4. Results

In this section, we empirically evaluate the performance of the MGSACO method upon five well-known microarray datasets. Seven frequently used gene/feature selection methods were selected to be compared with the proposed method. Term variance (TV) [1] and Laplacian score (LS) [2,18] are the univariate filter methods that can effectively remove irrelevant genes. Relevance-redundancy feature selection (RRFS) [24,25], random subspace method (RSM) [26,28], and mutual correlation (MC) [29,30] are multivariate filter methods which are proposed to address the irrelevant and redundant genes. Moreover, since the proposed method is an ACO-based gene selection method, unsupervised feature selection based on the ACO method (UFSACO) [13] is selected as the benchmark method which is the latest ACO-based gene selection method. Finally, the minimal-redundancy-maximal-relevance method (mRMR) [5,21] is selected to be applied in this study as a well-known and widely used supervised multivariate gene selection method by researchers in the literatures.

Since the proposed method is a filter-based gene selection method without using any classifiers in the gene selection process, it should have good performance on different types of classifiers. Therefore, to evaluate the adequacy of the gene selection method over the datasets, three frequently used classifiers including support vector machine (SVM) [40], naïve Bayes (NB) [1], and decision tree (DT) [60] were considered. The WEKA machine learning software library [61] was applied to the implementation of the presented classifiers. SMO with the polykernel was selected as the SVM classifier and it was applied with the one-against-rest strategy for the multiclass problems. Also, in SMO classifier, the complexity parameter c was set to 1 and the tolerance parameter was set to 0.001. Additionally, NaïveBayes was used as the NB classifier. Moreover, J48 was adopted as the DT classifier, in which the post-pruning technique was used in the pruning phase, its confidence factor was set to 0.25, and the minimum number of samples per leaf was set to 2.

To evaluate the performance of the methods, the average classification error rate of 5 independent runs with random train/test partitions was considered, so in each run, the datasets were randomly distributed into the training set (2/3 of the datasets) and test set. The experiments were performed on a 2.13 GHz Intel Core-i3 CPU with 4 GB of RAM using Java.

The description of the datasets used in the experiment, the parameter settings, and the experimental results are presented in the following subsections.

4.1. Datasets

The comparison is performed on the five well-known and extensively used microarray datasets with a wide range of cancer types. The two datasets, *Colon* and *Leukemia*, are publically available at the Bioinformatics Research Group from Universidad Pablo de Olavide [62] and the three datasets, *SRBCT*, *Prostate Tumor*, and *Lung Cancer*, are publically available at Gene Expression Model Selector from Vanderbilt University [63]. Colon, Leukemia, and Prostate Tumor datasets are the binary classification problems which deal with the problem of cancer detection. Moreover, SRBCT and Lung Cancer datasets represent multi-class problems the task

of which is to classify the tumors into different types. A brief description of these datasets is summarized in Table 1.

4.2. Parameter settings

In the proposed MGSACO method, several parameters have to be initialized. The maximum number of cycles is set to 50 ($I=50$), the number of ants is set to 100 ($A=100$), parameter q_0 in Eqs. (2) and (3) is set to 0.7, the initial pheromone values on each edge are set to 0.2 ($\tau_{ij}=0.2,\ \forall i, j=1\ldots n$), parameter β is set to 1, that shows the equal importance of the pheromone and heuristic information, and the evaporation rate parameter is set to 0.2 ($\rho=0.2$).

For the rest of the methods, there are parameters to be set. To make a fair comparison, the parameters of UFSACO are set to $NC_{max}=50$, $NAnt=100$, $q_0=0.7$, $\beta=1$, $\rho=0.2$, and $\tau_i=0.2$ as reported in [13]. Moreover, for the RRFS method, the maximum allowed similarity between pairs of features is set in the range of $[0.5, 1)$. Finally, for the RSM method, the number of iterations is set to 50, and the size of the subspace in each iterations is set to 200.

4.3. Experimental results

In the first set of experiments, the performance of the proposed method has been evaluated over different datasets using various types of classifiers. Tables 2–4 show the results of the comparison between the proposed MGSACO method and unsupervised filter-based methods including UFSACO, RSM, MC, RRFS, TV, and LS, in terms of average classification error rates (in %) over 5 different runs, by applying the SVM, NB, and DT classifiers, respectively. Note that, the last rows of these tables show the average classification error rates over all of the datasets.

It can be seen from Table 2 that the MGSACO method obtains the lowest classification error rate compared to the other methods on all of the datasets, expect for the Prostate Tumor dataset, that gets the second lowest error rates. For example, for the Leukemia dataset, the MGSACO acquired a 17.94% classification error rate while for the UFSACO, RSM, MC, RRFS, TV, and LS methods, this value was reported 41.02%, 37.64%, 38.23%, 23.52%, 20.58%, and 35.29%, respectively. Additionally, the average classification error rates over all of the datasets show that the proposed method with an error rate of 21.28% outperforms UFSACO by 8.48%, RSM by 10.45%, MC by 15.67%, RRFS by 4.67%, TV by 6.2%, and LS by 13.01%. Therefore, the proposed method gets the lowest average error rate compared to the other unsupervised filter-based methods.

The results of Table 3 illustrate that MGSACO outperforms the other methods in terms of classification error rate for the NB classifier on the Colon, SRBCT, Leukemia, and Lung Cancer datasets. The average values on all of the datasets, in the last row of Table 3, show that the MGSACO is superior to all the other methods. It outperforms UFSACO by 12.73%, RSM by 11.96%, MC by 18.24%, RRFS by 9.74%, TV by 15.44%, and LS by 10.07%.

From Table 4 it can be observed that the classification error rates of the DT classifier based on the MGSACO method is superior to those of the unsupervised filter-based methods when the Colon, SRBCT, Prostate Tumor, and Lung Cancer datasets are used. When MGSACO is used on the Leukemia dataset, the classification error rate of the DT classifier gets the second lowest, only inferior to that of the RRFS and TV methods. The average values in the last row of Table 4 show that the proposed method performs the best on all the datasets in terms of classification error rate. In other words, MGSACO outperforms UFSACO by 5.2%, RSM by 14.18%, MC by 11.68%, RRFS by 4.58%, TV by 3.82%, and LS by 12.06%.

It can be concluded from Tables 2–4 that MGSACO is the best one among the mentioned unsupervised filter-based methods (*i.e.*, UFSACO, RSM, MC, RRFS, TV, and LS) in terms of classification error rate for each of the three classifiers over different datasets.

In the second set of experiments, the execution time of the proposed method has been evaluated over different numbers of selected genes using all of the datasets. Fig. 2 shows the average execution times (in seconds) taken to the gene selection process of the MGSACO method. The results demonstrate that the run times of the proposed method, over all of the datasets, constantly increases as the size of selected genes is increased. Additionally, it can be observed that when the number of original genes in the

Table 1
Characteristics of the datasets used in the experiments.

Dataset	Genes	Classes	Patterns
Colon	2000	2	62
SRBCT	2308	4	83
Leukemia	7129	2	72
Prostate Tumor	10509	2	102
Lung Cancer	12600	5	203

Table 2
Average classification error rate over 5 runs of the unsupervised gene selection methods using SVM classifier. The best result is shown in bold face and underlined and second best is in bold face.

Datasets	#Selected genes	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
Colon	20	**21.81**	**21.81**	**24.54**	38.18	**24.54**	**21.81**	33.63
SRBCT	20	**25.51**	**28.27**	37.93	45.51	31.72	39.31	36.55
Leukemia	20	**17.94**	41.02	37.64	38.23	23.52	**20.58**	35.29
Prostate Tumor	20	**26.85**	40.57	**22.85**	34.28	30.85	28.00	48.00
Lung Cancer	20	**14.28**	**17.14**	35.71	28.57	19.14	27.71	18.00
Average		**21.28**	29.76	31.73	36.95	**25.95**	27.48	34.29

Table 3
Average classification error rate over 5 runs of the unsupervised gene selection methods using NB classifier. The best result is shown in bold face and underlined and second best is in bold face.

Datasets	#Selected genes	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
Colon	20	**20.00**	28.18	**26.36**	31.81	32.72	41.81	47.27
SRBCT	20	**15.86**	**20.00**	37.92	37.93	28.27	38.62	32.41
Leukemia	20	**7.69**	41.02	42.35	29.41	35.29	32.35	**8.82**
Prostate Tumor	20	37.14	39.42	**30.28**	33.71	**31.42**	33.14	32.57
Lung Cancer	20	**20.00**	35.71	23.57	59.04	**21.71**	31.99	29.99
Average		**20.14**	32.87	32.10	38.38	**29.88**	35.58	30.21

Table 4
Average classification error rate over 5 runs of the unsupervised gene selection methods using DT classifier. The best result is shown in bold face and underlined and second best is in bold face.

Datasets	#Selected genes	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
Colon	20	**23.63**	**24.54**	28.18	33.63	34.54	31.81	39.09
SRBCT	20	**22.75**	**27.58**	58.62	44.13	28.96	**22.75**	45.51
Leukemia	20	**23.07**	30.76	38.82	32.35	**20.58**	**20.58**	29.41
Prostate Tumor	20	**29.71**	**33.71**	**33.71**	36.00	37.71	38.85	43.99
Lung Cancer	20	**20.00**	28.57	30.71	31.42	**20.28**	24.28	21.43
Average		**23.83**	29.03	38.01	35.51	28.41	**27.65**	35.89

datasets is increased, finding the subset of genes requires a larger execution time.

In the third set of experiments, the performance of the proposed method has been evaluated over different numbers of selected genes using various types of classifiers. Tables 5–7 report the average classification error rates, over 5 independent runs of the proposed MGSACO method and those of unsupervised filter-based methods on the SRBCT dataset using the SVM, NB, and DT classifiers, correspondingly.

The results reported from Table 5 show that the proposed method has the best performance compared to those of unsupervised filter-based methods using the SVM classifier when the number of selected genes is 10, 20, 30, 40, 60, 70, 80, and 90. Additionally, MGSACO achieves the second lowest classification error rate only inferior to that of the UFSACO and RRFS when the number of selected genes is 50 and 100.

From Table 6 it is clear that the classification error rate of the proposed method is much lower than those of the other methods in most cases. For example, the classification error rate of the MGSACO is 15.86% when 20 genes are selected, while this value for UFSACO, RSM, MC, RRFS, TV, and LS was reported 20%, 37.92%, 37.93%, 28.27%, 38.62%, and 32.41%, correspondingly. Additionally, it can be observed that MGSACO acquires the lowest classification error rate 3.45% for NB classifier when 70 genes are selected.

The results of Table 7 show that in most cases the performance of the proposed method is significantly superior to all the other methods. For example, when the number of selected genes was 10, MGSACO outperforms UFSACO by 9.66%, RSM by 31.73%, MC by 33.8%, RRFS by 15.18%, TV by 20%, and LS by 28.97%. Furthermore, the worst classification error rate of the MGSACO was 22.75%, while for UFSACO, RSM, MC, RRFS, TV, and LS, the worst classification error rates were 31.03%, 58.62%, 55.17%, 36.55%, 41.37%, and 50.34%, respectively.

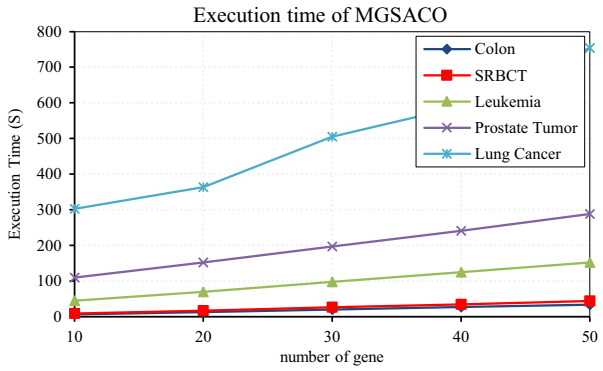

Fig. 2. Execution times (in seconds) of the proposed method over different datasets.

Additionally, Tables 8–10 summarize the average classification error rates, over 5 different runs of the proposed MGSACO method and those of the unsupervised filter-based methods on the Colon dataset using the SVM, NB, and DT classifiers, correspondingly.

It can be seen from Table 8 that the performance of MGSACO is much better than that of the other methods when the number of selected genes is 20, 40, 50, 60, 70, 90, and 100. On the other hand, the proposed method attains the second lowest error rate when 10, 30, and 80 genes are selected. Furthermore, it can be seen that the lowest classification error rate of the proposed method was 7.27% when 70 genes are selected, while this value for UFSACO, RSM, MC, RRFS, TV, and LS was reported 9.09% with 80 genes, 16.36% with 50 genes, 19.09% with 60 genes, 12.72% with 50 genes, 12.73% with 50 genes, and 30.90% with 100 genes, correspondingly.

From the results of Table 9 it can be observed that the proposed method gets the first place in most cases and it achieves the second place only inferior to the UFSACO and MC methods when 60 and 100 subsets of genes are selected. For example, when 50 genes are selected, MGSACO with an error rate of 15.45% lies on the first place among the gene selection methods and it outperforms UFSACO by 9.09%, RSM by 12.73%, MC by 14.55%, RRFS by 14.54%, TV by 20%, and LS by 31.82%.

Table 10 illustrates that when the number of selected gene was 20 and 30, the classification error rate of the proposed method was 23.63% and 19.09%, respectively, which shows that the MGSACO is superior to the other methods. Moreover, the MGSACO gets the second best result over 10, 40, 70, 80, and 90 genes selected. On the other hand, the performance of the proposed method is better than MC, RRFS, and LS methods when the number of selected genes was 50. Additionally, MGSACO is superior to the RSM, RRFS, TV, and LS methods when 60 genes are selected.

From Tables 5–10, we observe that the performance of the proposed method in terms of classification error rates was superior to those of the other methods on the various subsets of genes by applying different classifiers. This is because the proposed method uses relevance and redundancy analyses in the gene selection procedure to address the irrelevant and redundant genes in the datasets. Also, the proposed method selects the subsets of genes without using any classifiers in an iterative improvement process, which can cause good performance over different classifiers.

In the fourth set of experiments, the performance of the proposed method has been compared to supervised gene selection method in terms of classification error rates over different numbers of selected genes. Figs. 3–7 display the results of the comparison between the MGSACO and mRMR methods in terms of classification error rates (average over 5 different runs) of the SVM, NB, and DT classifiers on all of the dataset in Table 1. Note that, the *x*-axis denotes the number of selected genes, while the *y*-axis shows the classification error rates (in %).

Table 5
Average classification error rates (in %) over 5 runs of the gene selection methods on SRBCT dataset using SVM classifier. The best result is indicated in bold face and underlined and the second best is in bold face.

Methods	Number of selected genes									
	10	20	30	40	50	60	70	80	90	100
MGSACO	**39.30**	**25.51**	**14.48**	**7.58**	**7.58**	**4.14**	**3.45**	**2.07**	**0.69**	**2.07**
UFSACO	51.72	**28.27**	**14.48**	**9.65**	**4.14**	**4.14**	**4.14**	3.45	5.52	**0.69**
RSM	59.99	37.93	35.86	39.30	18.62	18.62	16.55	15.86	13.79	8.96
MC	64.13	45.51	41.38	34.48	17.23	17.93	14.48	15.17	11.03	9.65
RRFS	45.51	31.72	**15.86**	13.79	**4.14**	**4.14**	8.96	**2.76**	4.14	**2.07**
TV	46.89	39.31	28.96	25.51	19.31	**6.90**	**3.45**	5.52	**3.45**	3.45
LS	**44.13**	36.55	28.96	13.79	17.24	13.10	18.62	10.34	9.65	6.89

Table 6
Average classification error rates (in %) over 5 runs of the gene selection methods on SRBCT dataset using NB classifier. The best result is indicated in bold face and underlined and the second best is in bold face.

Methods	Number of selected genes									
	10	20	30	40	50	60	70	80	90	100
MGSACO	**18.62**	**15.86**	**11.72**	**9.65**	**15.17**	**7.58**	**3.45**	**4.14**	**8.27**	**5.52**
UFSACO	**31.03**	**20.00**	**10.34**	**11.72**	**12.41**	**6.21**	**11.03**	13.10	**3.45**	**13.10**
RSM	46.20	37.92	39.31	25.51	22.75	22.75	20.68	25.51	20.69	22.06
MC	35.17	37.93	20.69	26.89	23.44	21.37	17.24	23.44	23.44	16.55
RRFS	37.24	28.27	22.06	22.75	23.45	15.86	11.72	**5.52**	10.34	22.75
TV	47.58	38.62	27.58	24.13	18.62	17.93	19.31	8.96	8.96	17.24
LS	42.06	32.41	27.58	27.58	26.20	27.58	12.41	22.76	26.89	24.13

Table 7
Average classification error rates (in %) over 5 runs of the gene selection methods on SRBCT dataset using DT classifier. The best result is indicated in bold face and underlined and the second best is in bold face.

Methods	Number of selected genes									
	10	20	30	40	50	60	70	80	90	100
MGSACO	**21.37**	**22.75**	**19.30**	22.07	**16.55**	**17.93**	**13.79**	**17.24**	22.06	**15.86**
UFSACO	**31.03**	**27.58**	26.89	**14.48**	25.51	**21.38**	26.20	20.00	**18.62**	**22.75**
RSM	53.10	58.62	41.37	45.51	34.48	37.24	40.00	37.93	37.93	31.72
MC	55.17	44.13	39.31	35.16	41.37	29.65	28.96	36.55	40.68	38.62
RRFS	36.55	28.96	**24.82**	22.07	26.20	22.75	**20.69**	**17.93**	22.06	23.44
TV	41.37	**22.75**	32.41	27.58	**24.13**	23.44	24.13	**17.24**	**20.68**	31.03
LS	50.34	45.51	24.83	**20.00**	25.51	22.75	21.37	21.37	26.20	28.96

Table 8
Average classification error rates (in %) over 5 runs of the gene selection methods on Colon dataset using SVM classifier. The best result is indicated in bold face and underlined and the second best is in bold face.

Methods	Number of selected genes									
	10	20	30	40	50	60	70	80	90	100
MGSACO	**23.63**	**21.81**	**18.18**	**14.54**	**12.72**	**12.73**	**7.27**	**13.63**	**11.82**	**16.36**
UFSACO	**23.63**	**21.81**	**16.36**	**15.45**	**12.72**	22.72	**16.36**	**9.09**	16.36	18.18
RSM	35.45	**24.54**	20.90	18.18	16.36	24.54	20.00	19.09	19.09	**16.36**
MC	32.72	38.18	31.81	29.09	25.45	19.09	24.54	19.09	19.09	19.09
RRFS	**12.73**	**24.54**	20.90	18.18	**12.72**	**18.18**	**16.36**	18.18	18.18	24.54
TV	24.54	**21.81**	**18.18**	26.36	**12.73**	20.00	21.81	17.27	**15.45**	**17.27**
LS	39.09	33.63	37.27	40.00	31.81	35.45	33.63	30.91	33.63	30.90

Table 9
Average classification error rates (in %) over 5 runs of the gene selection methods on Colon dataset using NB classifier. The best result is indicated in bold face and underlined and the second best is in bold face.

Methods	Number of selected genes									
	10	20	30	40	50	60	70	80	90	100
MGSACO	35.45	**20.00**	**22.72**	**19.99**	**15.45**	**18.18**	**18.18**	**17.27**	**16.36**	**20.91**
UFSACO	30.00	28.18	**26.36**	26.36	**24.54**	**12.72**	**18.18**	**19.09**	**16.36**	26.36
RSM	**29.09**	**26.36**	34.54	**22.72**	28.18	29.99	**19.09**	30.90	26.36	30.90
MC	41.81	31.81	33.63	**22.72**	30.00	39.08	22.72	25.45	**24.54**	**18.18**
RRFS	**27.27**	32.72	31.81	30.90	29.99	25.45	29.09	**19.09**	34.54	27.27
TV	36.36	41.81	39.09	38.18	35.45	35.45	28.18	29.08	35.45	24.54
LS	58.18	47.27	48.18	39.09	47.27	50.00	39.08	38.18	40.90	50.90

Fig. 3(a) shows that there is no method that performs consistently better than the others and the overall performance of MGSACO and mRMR is similar. MGSACO gets slightly lower classification error rates when the number of genes is in the range between 10 and 25. From Fig. 3(b), it can be concluded that when the number of genes is greater than 10, the classification error rate curve of MGSACO is significantly superior to that of the mRMR method. Especially, when 36 genes were selected, the classification error rates were around 11% and 38% for MGSACO and mRMR, respectively. As seen in Fig. 3(c), the lowest classification error rate of the proposed method was 12.12% when 26 genes were selected, while for the mRMR this value was reported 21.21% when 25 genes were selected.

Fig. 4(a)–(c) show the respective comparison results for SRBCT dataset. The different classification error rate curves of MGSACO and mRMR can be seen more prominently for this dataset. Fig. 4(a) indicates that the classification error rates of the proposed method are much lower than those of mRMR in most cases. Fig. 4(b) shows that MGSACO acquires significantly lower classification error

Table 10
Average classification error rates (in %) over 5 runs of the gene selection methods on Colon dataset using DT classifier. The best result is indicated in bold face and underlined and the second best is in bold face.

Methods	Number of selected genes									
	10	20	30	40	50	60	70	80	90	100
MGSACO	**24.54**	**23.63**	**19.09**	**20.00**	29.08	21.81	**22.72**	**20.00**	**24.54**	27.27
UFSACO	**14.54**	**24.54**	25.45	25.45	24.54	**19.08**	**20.00**	24.54	26.36	**19.99**
RSM	38.18	28.18	27.27	23.63	**19.99**	27.27	25.45	22.72	33.63	26.36
MC	40.00	33.63	41.81	**19.09**	32.72	**16.36**	25.45	**18.18**	30.90	**22.72**
RRFS	**24.54**	34.54	**24.54**	21.81	36.36	29.09	26.36	31.81	27.27	**22.72**
TV	41.81	31.81	27.27	34.54	**23.63**	28.18	**22.72**	20.90	**23.63**	29.09
LS	41.81	39.09	40.00	33.63	30.90	32.72	41.81	29.99	36.36	34.54

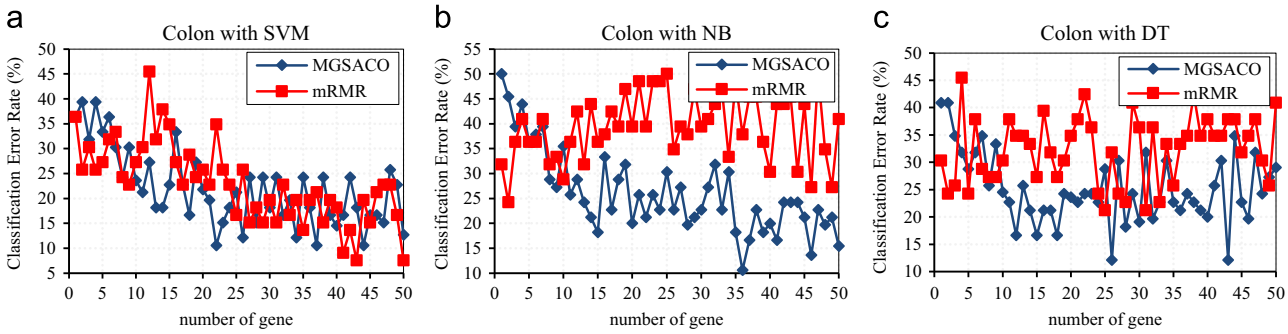

Fig. 3. Classification error rates (average over 5 different runs) with respect to the number of selected genes on the Colon dataset with (a) the support vector machine classifier, (b) the naïve Bayes classifier, and (c) the decision tree classifier.

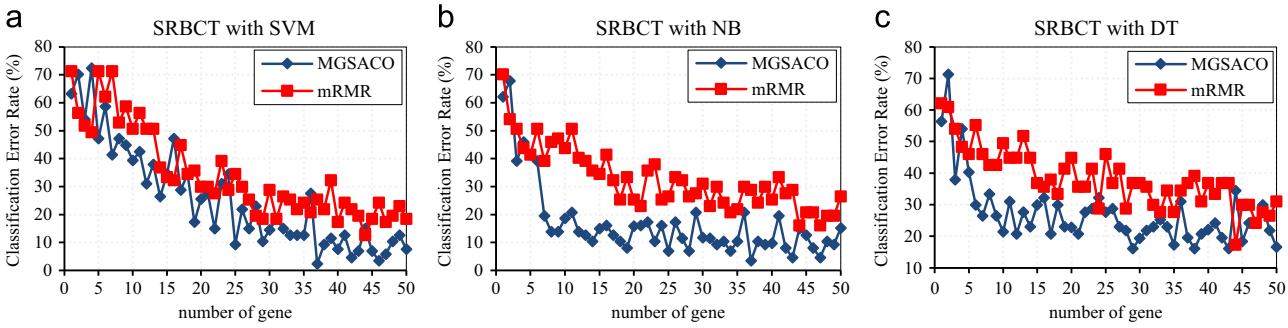

Fig. 4. Classification error rates (average over 5 different runs) with respect to the number of selected genes on the SRBCT dataset with (a) the support vector machine classifier, (b) the naïve Bayes classifier, and (c) the decision tree classifier.

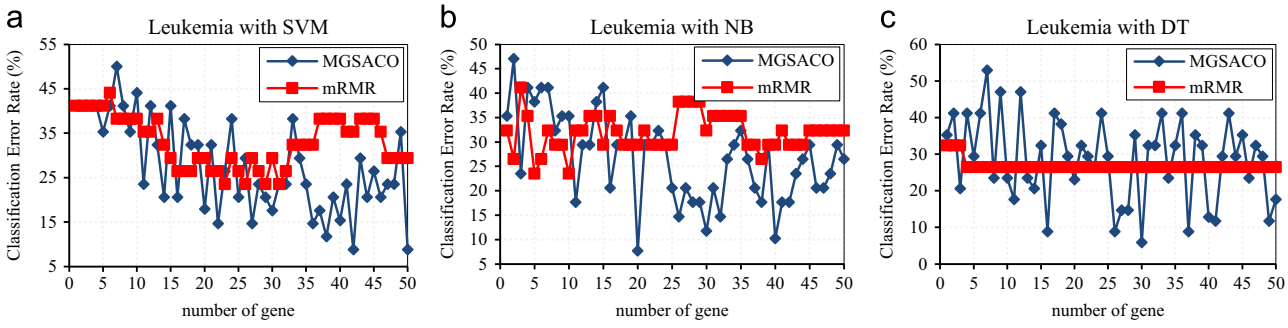

Fig. 5. Classification error rates (average over 5 different runs) with respect to the number of selected genes on the Leukemia dataset with (a) the support vector machine classifier, (b) the naïve Bayes classifier, and (c) the decision tree classifier.

rates than mRMR when the number of selected genes is greater than 5. For example, when the number of selected genes was 9, the error rates of MGSACO and mRMR were around 14% and 47%, correspondingly. Fig. 4(c) shows that similarly to those of the NB classifier, the classification error rates of the MGSACO significantly and consistently outperformed those of mRMR when the number of selected genes was in the range between 5 and 43.

Fig. 5(a) demonstrates that the overall performance of MGSACO is superior to that of the mRMR method when the SVM classifier is applied. Especially, when 40 genes were selected, the classification error rate of the proposed method was 15.38%, while for the mRMR this value was reported 38.23%. Moreover, it can be seen from Fig. 5(b) that the performance of the proposed method is much better than that of the mRMR when the number of selected

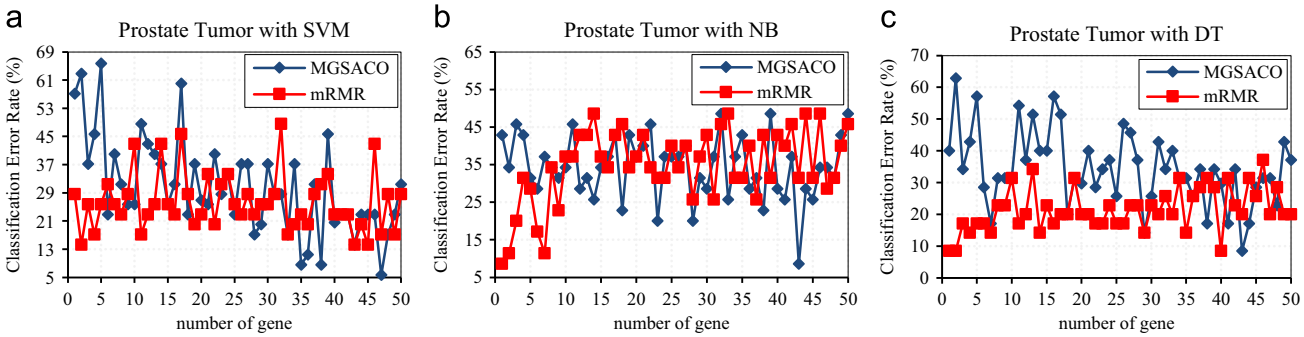

Fig. 6. Classification error rates (average over 5 different runs) with respect to the number of selected genes on the Prostate Tumor dataset with (a) the support vector machine classifier, (b) the naïve Bayes classifier, and (c) the decision tree classifier.

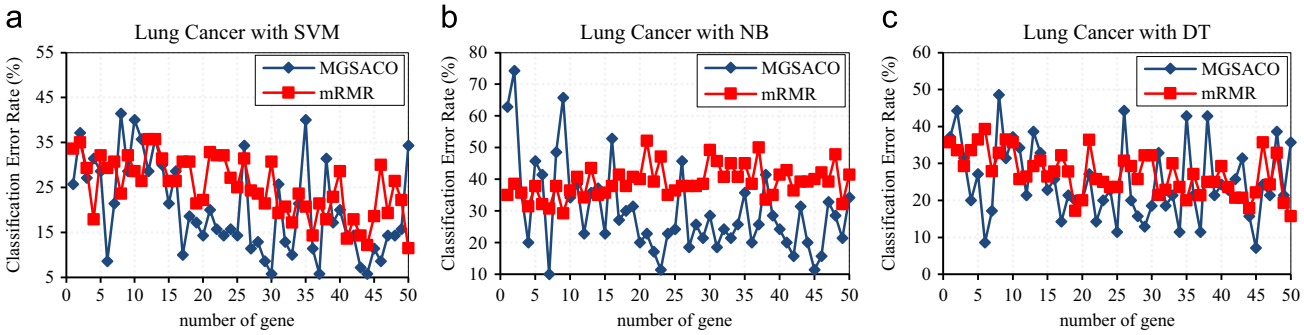

Fig. 7. Classification error rates (average over 5 different runs) with respect to the number of selected genes on the Lung Cancer dataset with (a) the support vector machine classifier, (b) the naïve Bayes classifier, and (c) the decision tree classifier.

genes is greater than 24. Especially, when the number of selected genes was 20, 30, and 40, MGSACO got classification error rates of 7.69%, 11.76%, and 10.25%, correspondingly, while in these cases the classification error rates of mRMR were reported 29.41%, 32.35%, and 29.41%, respectively. Furthermore, similar results have been reported when the DT classifier is used. As seen in Fig. 5(c), the performance curve of mRMR is almost constant. On the other hand, the lowest error rate of the proposed method was 5.88%, while this value for mRMR was 26.47%.

Fig. 6(a) illustrates that the overall performance of MGSACO and mRMR is close. However, the proposed method attained the lowest classification error rate, 5.71%, compared to the mRMR method. Also, similar results have been reported when the NB classifier is used. Fig. 6(b) shows that no method performs consistently better than the others. Moreover, Fig. 6(c) shows that the results of MGSACO are inferior to those of mRMR.

Fig. 7(a) indicates that the best performance of the proposed method was achieved (with error rate 5.71%) when the number of selected genes were 30, 37, and 44, while for mRMR method this value was 11.43% when 50 genes were selected. From Fig. 7(b) it is clear that the classification error rate of the proposed method is much lower than that of mRMR method in most cases. For example, when 23 genes were selected, the classification error rates of MGSACO and mRMR methods were 11.42% and 47.14%, correspondingly. The results of Fig. 7(c) demonstrate that the overall performance of the proposed method is better than that of mRMR method. The best result of the MGSACO was 7.14% when the number of selected genes was 45, while this value for mRMR was reported 15.71% when 50 genes were selected.

It can be concluded from Figs. 3–7 that although the MGSACO is an unsupervised method and does not need class labels of the samples, it can be much better than the mRMR method, which is a supervised method, in terms of classification error rate. This is because the mRMR selects the subset of genes in a single-track process and also it starts to search from a specific point. But, the MGSACO is a population-based method which simultaneously explores the search space from different points. Additionally, the proposed method uses an iterative improvement process to select the subsets of genes.

5. Statistical analysis and discussion

5.1. Statistical analysis

In order to illustrate that the experimental results are statistically significant, the Friedman test [64] has been performed on the results. The Friedman test is a non-parametric test used to measure the statistical differences of methods over multiple datasets. For each dataset, the methods are ranked separately based on the classification error rates. The method with the lowest classification error rate gets rank 1, the second lowest result gets rank 2, and so on. When several methods have the same classification error rate, their average rank is assigned to each method. The Friedman test is distributed according to the Fisher distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom which is defined as follows:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{8}$$

where

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{9}$$

N is the number of datasets, k is the number of methods, and R_j is the average rank of the jth method over all datasets. The null hypothesis in Friedman test means that all methods perform equivalently at the significance level α. The null hypothesis is accepted when F_F is less than the critical value; otherwise it is rejected. In the experiments, the significance level was set to $\alpha = 0.05$.

Table 11 reports the average ranks of the unsupervised filter-based methods using SVM, NB, and DT classifiers. It should be noted that these average ranks are computed according to the values in Tables 2–4.

Table 12 presents the results of Friedman test for comparison between the proposed method and the unsupervised filter-based methods. In the experiments, $N=5$, $k=7$, and the critical value of Fisher distribution with $7-1=6$ and $(7-1)(5-1)=24$ degrees of freedom is equals to $F(6,24)=2.51$ for $\alpha=0.05$. It can be seen that when the gene selection methods are incorporated with SVM and DT classifiers, the value of F_F is greater than 2.51. Therefore, the null hypothesis will be rejected and it can be concluded that these results are statistically significant. Moreover, the value of F_F is less than 2.51 when NB classifier is used. Therefore, the null hypothesis is accepted and it is clear that MGSACO method performs equivalently with the other methods.

5.2. Discussion

Microarray datasets contain high-dimensional genes most of which are irrelevant and redundant. A good gene selection method must be able to handle both irrelevant and redundant genes. The LS and TV methods are univariate ones that can only handle irrelevant genes. Moreover, LS is sample-based method and it has poor performance over microarray datasets due to the very small samples. On the other hand, UFSACO, RSM, MC, and RRFS are multivariate methods which are able to eliminate both irrelevant and redundant genes. However, MC and RRFS are based on a single-track process and they also start to explore the search space from a specific point; thus, they can be trapped in a local optimum. Furthermore, RSM is suitable when a given classifier is applied as a multivariate method in its search process.

The MGSACO uses relevance and redundancy analyses in the gene selection procedure to address the irrelevant and redundant genes. This scheme provides better performance than the LS and TV methods. Moreover, MGSACO is a population-based method which simultaneously explores the search space from different points in an iterative improvement process. Therefore, these mechanisms lead to achievement of better performance than that of MC, RRFS and RSM methods. Moreover, the proposed method is essentially different from UFSACO in several aspects:

(1) In MGSACO both relevance and redundancy analyses are used in the gene selection process, while UFSACO only uses the redundancy analysis. Therefore, the performance of UFSACO can be reduced on the datasets with lower redundancy between genes.
(2) The pheromone values in MGSACO are associated with the edges in the graph. Therefore, the number of times that a specific edge (i, j) is visited by ants indicates the importance of considering these two genes (*i.e.*, genes i and j) together. However, the pheromone values in UFSACO are associated with the nodes in the graph. Therefore, the relevance of each gene is evaluated based on its similarity values to the other genes.
(3) In MGSACO an ant constructs its candidate solution in each step by selecting the edge with the highest pheromone value and lowest similarity in an iterative way. This strategy leads to effective and significative redundancy analyses. But in UFSACO an ant constructs a subset of genes by selecting the best gene in an iterative way. Therefore, the state transition rule will be different due to the use of different pheromone values.
(4) MGSACO keeps the best subset of genes in each iteration, and finally the global best one in all iterations is chosen as a final result. But UFSACO is a ranking method in which the relevance of genes is computed in different iterations, and finally a subset of genes with the highest pheromone values is selected as a final result. Therefore, the redundancy analysis of MGSACO is much better than that of UFSACO.
(5) MGSACO determines the quality of the found solutions with a specific fitness function, and also the fitness values are considered in the pheromone updating rule. The fitness function increases the pheromone level of the solution in order to make the solution components more attractive to ants in the next iterations. But UFSACO does not use any fitness functions at the end of iterations.

It is demonstrated in Tables 2–4 that the proposed method significantly obtained better performance than the other methods over all datasets using different classifiers (*i.e.*, SVM, NB, and DT classifiers). Moreover, as can be seen in Tables 5–10, MGSACO with SVM classifier achieved the lowest classification error rate over Colon dataset. Also, the MGSACO attained the lowest classification error rate over SRBCT dataset when it was combined with SVM classifier. It can be inferred from the obtained results that MGSACO is more suitable when SVM classifier is used. The main reason is that MGSACO seeks the best edges in each iteration and considers the interaction between genes. Furthermore, it can be concluded that MGSACO is effective to find minimal subsets of genes with lower associated classification error rates than those of the other methods.

As mentioned in the introduction section, the aim of the proposed method is to design a framework to combine the filter approach and ACO search strategy without using any learning models. The results of Tables 2–10 and Fig. 2 show that the proposed method attained better classification accuracy than the unsupervised filter methods within an acceptable computational time. However, it is suggested that one perform a simple filter method in order to reduce the search space before starting the search process of the MGSACO for the datasets with very large numbers of genes (*i.e.*, more than 20,000 genes).

The MGSACO is an unsupervised method in which the class labels are not considered in the gene selection process. However, the results of Figs. 3–7 show that the performance of the proposed method is much better than that of the mRMR method, which is a supervised approach. This is due to the fact that the mRMR selects the first gene with maximal relevance to the class label based on the mutual information measure. However, in some of the microarray datasets, there are samples that are incorrectly labeled or whose class labels may be unreliable. Also, the values of genes are real numbers and the numbers of samples in these datasets are

Table 11
Average ranks of the unsupervised methods considered over five datasets, based on the classification error rate of SVM, NB, and DT classifiers.

Datasets	MGSACO	UFSACO	RSM	MC	RRFS	TV	LS
SVM	1.4	3.8	4.5	6.2	3.7	3.6	4.8
NB	2.0	4.8	3.6	5.0	3.4	5.2	4.0
DT	1.5	3.5	5.1	5.4	3.7	3.4	5.4

Table 12
The results of Friedman test for the comparisons between unsupervised methods.

Classifier	χ_F^2	F_F	$F(6,24)$	Significant
SVM	13.693	3.359	2.51	+
NB	8.14	1.49	2.51	=
DT	12.94	3.03	2.51	+

very small. Therefore, mRMR method cannot correctly identify the gene with maximal relevance. Furthermore, the search process of the mRMR is started at a given point and the final subset of genes is obtained in a single-track process. On the other hand, MGSACO computes the relevance of genes based on the term variance measure which is an unsupervised approach and does not need the class labels. Also, this measure is more suitable for the datasets with real values. Moreover, the iterative improvement process, the population-based mechanism, and the greedy and stochastic natures of the proposed method leads to increase in its efficiency compared to mRMR method.

6. Conclusion

In this paper, an unsupervised filter method based on the ACO algorithm, called MGSACO, was proposed for the gene selection in the microarray data. The computational efficiency of the filter approach and good performance of the ACO search strategy were combined to improve the performance of the proposed method. In addition we employed a new fitness function to evaluate the subsets of selected genes without using any learning model to enhance the efficiency of the proposed method.

The performance of the proposed method was examined on the five microarray datasets using three different classifiers including support vector machine, naïve Bayes, and decision tree. Furthermore, the proposed method was compared to the state-of-the-art unsupervised filter-based gene selection methods including unsupervised feature selection based on ACO (UFSACO), relevance-redundancy feature selection (RRFS), random subspace method (RSM), mutual correlation (MC), term variance (TV), and Laplacian score (LS) and also compared to the well-known and frequently used supervised filter-based method: the minimal-redundancy-maximal-relevance (mRMR) method. The experimental results show that the MGSACO was able to select a subset of genes with minimum redundancy and maximum relevance. Moreover, the results show that the classification accuracy of the proposed method is much superior to that of the other unsupervised methods for various subsets of genes over all the three classifiers. Furthermore, the results indicate that the MGSACO was significantly better than the supervised mRMR method due to population-based and iterative improvement natures of the proposed method. We also confirm that the proposed method obtained good performance over different classifiers.

Future works will focus on development of a metric to define the number of selected genes for each ant. Also, another extension would be to define new fitness functions to enhance the efficiency of the gene selection procedure of the proposed method.

References

[1] S. Theodoridis, K. Koutroumbas, Pattern Recognition, fourth ed., Academic Press, Oxford, 2008.

[2] B. Liao, Y. Jiang, W. Liang, W. Zhu, L. Cai, Z. Cao, Gene selection using locality sensitive Laplacian score, IEEE/ACM Trans. Comput. Biol. Bioinform. 11 (2014) 1146–1156. http://dx.doi.org/10.1109/TCBB.2014.2328334.

[3] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. d. Schaetzen, R. Duque, H. Bersini, A. Nowe, A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (2012) 1106–1119.

[4] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J.M. Benítez, F. Herrera, A review of microarray datasets and applied feature selection methods, Inf. Sci. 282 (2014) 111–135 http://dx.doi.org/10.1016/j.ins.2014.05.042.

[5] C. DING, H. PENG, Minimum redundancy feature selection from microarray gene expression data, J. Bioinform. Comput. Biol. 03 (2005) 185–205.

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

[7] Y. Leung, Y. Hung, A Multiple-Filter-Multiple-Wrapper, Approach to gene selection and microarray data classification, IEEE/ACM Trans. Comput. Biol. Bioinform. 7 (2010) 108–117.

[8] A. Najafi, G. Bidkhori, J.H. Bozorgmehr, I. Koch, A. Masoudi-Nejad, Genome scale modeling in systems biology: algorithms and resources, Curr. Genomics 15 (2014) 130–159.

[9] M. Mirzaei, A. Najafi, M. Arababadi, M. Asadi, S. Mowla, Altered expression of apoptotic genes in response to OCT4B1 suppression in human tumor cell lines, Tumor Biol. 35 (2014) 9999–10009.

[10] S. Niijima, Y. Okuno, Laplacian linear discriminant analysis approach to unsupervised feature selection, IEEE/ACM Trans. Comput. Biol. Bioinform. 6 (2009) 605–614.

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

[12] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (2005) 491–502.

[13] S. Tabakhi, P. Moradi, F. Akhlaghian, An unsupervised feature selection algorithm based on ant colony optimization, Eng. Appl. Artif. Intell. 32 (2014) 112–123.

[14] R. Cai, Z. Hao, X. Yang, W. Wen, An efficient gene selection algorithm based on mutual information, Neurocomputing 72 (2009) 991–999.

[15] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (2007) 2507–2517.

[16] I. Inza, P. Larrañaga, R. Blanco, A.J. Cerrolaza, Filter versus wrapper gene selection approaches in DNA microarray domains, Artif. Intell. Med. 31 (2004) 91–103.

[17] C. Lai, M. Reinders, L. van't Veer, L. Wessels, A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets, BMC Bioinform. 7 (2006) 235.

[18] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, Adv. Neural Inf. Process. Syst. 18 (2005) 507–514.

[19] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.

[20] L.E. Raileanu, K. Stoffel, Theoretical comparison between the Gini index and information gain criteria, Ann. Math. Artif. Intell. 41 (2004) 77–93.

[21] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1226–1238.

[22] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.

[23] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, in: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 856–863.

[24] A.J. Ferreira, M.A.T. Figueiredo, An unsupervised approach to feature discretization and selection, Pattern Recognit. 45 (2012) 3048–3060.

[25] A.J. Ferreira, M.A.T. Figueiredo, Efficient feature selection filters for high-dimensional data, Pattern Recognit. Lett. 33 (2012) 1794–1804.

[26] C. Lai, M.J.T. Reinders, L. Wessels, Random subspace method for multivariate feature selection, Pattern Recognit. Lett. 27 (2006) 1067–1076.

[27] A. Bertoni, R. Folgieri, G. Valentini, Bio-molecular cancer prediction with random subspace ensembles of support vector machines, Neurocomputing 63 (2005) 535–539.

[28] X. Li, H. Zhao, Weighted random subspace method for high dimensional data classification, Stat. Interface 2 (2009) 153–159.

[29] M. Haindl, P. Somol, D. Ververidis, C. Kotropoulos, Feature selection based on mutual correlation, in: Pattern Recognition, Image Analysis and Applications, Springer, Berlin, Heidelberg (2006) 569–577.

[30] S.N. Ghazavi, T.W. Liao, Medical data mining by fuzzy modeling with selected features, Artif. Intell. Med. 43 (2008) 195–206.

[31] I.A. Gheyas, L.S. Smith, Feature subset selection in large dimensionality domains, Pattern Recognit. 43 (2010) 5–13.

[32] I. Inza, B. Sierra, R. Blanco, P. Larrañaga, Gene selection by sequential search wrapper approaches in microarray cancer class prediction, J. Intell. Fuzzy Syst. 12 (2002) 25–33.

[33] M.M. Kabir, M. Shahjahan, K. Murase, A new hybrid ant colony optimization algorithm for feature selection, Expert Syst. Appl. 39 (2012) 3747–3763.

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

[35] B. Sahu, D. Mishra, A Novel, Feature selection algorithm using particle swarm optimization for cancer microarray data, Procedia Eng. 38 (2012) 27–31.

[36] E. Martinez, M.M. Alvarez, V. Trevino, Compact cancer biomarkers discovery using a swarm intelligence feature selection algorithm, Comput. Biol. Chem. 34 (2010) 244–250.

[37] J.J. Liu, G. Cutler, W. Li, Z. Pan, S. Peng, T. Hoey, L. Chen, X.B. Ling, Multiclass cancer classification and biomarker discovery using GA-based algorithms, Bioinformatics 21 (2005) 2691–2697.

[38] C.H. Ooi, P. Tan, Genetic algorithms applied to multi-class prediction for the analysis of gene expression data, Bioinformatics 19 (2003) 37–44.

[39] A. Srivastava, S. Chakrabarti, S. Das, S. Ghosh, V.K. Jayaraman, Hybrid firefly based simultaneous gene selection and cancer classification using support vector machines and random forests, in: Proceedings of the Seventh

International Conference on Bio-inspired Computing: Theories and Applications, Springer, India, 2013, pp. 485–494.
[40] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.
[41] R. Diaz-Uriarte, S. Alvarez de Andres, Gene selection and classification of microarray data using random forest, BMC Bioinform. 7 (2006) 3.
[42] G. Wang, Q. Song, B. Xu, Y. Zhou, Selecting feature subset for high dimensional data via the propositional FOIL rules, Pattern Recognit. 46 (2013) 199–214.
[43] A. Zibakhsh, M.S. Abadeh, Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function, Eng. Appl. Artif. Intell. 26 (2013) 1274–1281.
[44] W. Zhao, G. Wang, H.-b. Wang, H.-l. Chen, H. Dong, Z.-d. Zhao, A Novel, Framework for gene selection, Int. J. Adv. Comput. Technol. 3 (2011) 184–191.
[45] H.R. Kanan, K. Faez, An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system, Appl. Math. Comput. 205 (2008) 716–725.
[46] M.H. Aghdam, N. Ghasem-Aghaee, M.E. Basiri, Text feature selection using ant colony optimization, Expert Syst. Appl. 36 (2009) 6843–6853.
[47] Y. Marinakis, M. Marinaki, M. Doumpos, C. Zopounidis, Ant colony and particle swarm optimization for financial classification problems, Expert Syst. Appl. 36 (2009) 10604–10611.
[48] M. Dorigo, G. Di Caro, Ant colony optimization: a new meta-heuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999, pp. 1470–1477.
[49] M. Dorigo, T. Stützle, Ant colony optimization: overview and recent advances, in: Handbook of Metaheuristics, Springer, US (2010) 227–263.
[50] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man, Cybern. B: Cybern. 26 (1996) 29–41.
[51] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1997) 53–66.
[52] M. Dorigo, L.M. Gambardella, Ant colonies for the travelling salesman problem, Biosystems 43 (1997) 73–81.
[53] A.M.d. Mesleh, G. Kanaan, Support vector machine text classification system: using ant colony optimization based feature subset selection, in: Proceedings of the International Conference on Computer Engineering & Systems, 2008, pp. 143–148.
[54] M.H. Aghdam, N. Ghasem-Aghaee, M.E. Basiri, Application of ant colony optimization for feature selection in text categorization, in: Proceedings of the IEEE Congress on Evolutionary Computation, 2008), pp. 2867–2873.
[55] S. Kashef, H. Nezamabadi-pour, An advanced ACO algorithm for feature subset selection, Neurocomputing 147 (2015) 271–279 http://dx.doi.org/10.1016/j.neucom.2014.06.067.
[56] B. Chen, L. Chen, Y. Chen, Efficient ant colony optimization for image feature selection, Signal Process. 93 (2013) 1566–1576.
[57] S. Nemati, M.E. Basiri, Text-independent speaker verification using ant colony optimization-based selected features, Expert Syst. Appl. 38 (2011) 620–630.
[58] S.M. Vieira, J.M.C. Sousa, T.A. Runkler, Two cooperative ant colonies for feature selection using fuzzy models, Expert Syst. Appl. 37 (2010) 2714–2723.
[59] S. Nemati, M.E. Basiri, N. Ghasem-Aghaee, M.H. Aghdam, A novel ACO–GA hybrid algorithm for feature selection in protein function prediction, Expert Syst. Appl. 36 (2009) 12086–12094.
[60] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.
[61] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten, The WEKA data mining software, available: ⟨http://www.cs.waikato.ac.nz/ml/weka⟩.
[62] Dataset Repository, Bioinformatics Research Group, available: ⟨http://www.upo.es/eps/bigs/datasets.html⟩, (2014).
[63] A. Statnikov, C.F. Aliferis, I. Tsamardinos, Gems: Gene Expression Model Selector, Available: ⟨http://www.gems-system.org/⟩, (2005).
[64] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (1937) 675–701.

Sina Tabakhi received the B.Sc. degree with honors in information technology from Azad University, Sanandaj Branch, Iran, in 2011, and the M.Sc. degree with honors in computer engineering from University of Kurdistan, Iran, in 2013. He is currently a lecturer in the department of computer engineering, University of Kurdistan, Iran. His research interests include pattern recognition, machine learning, feature selection, data mining, and bioinformatics.

Ali Najafi is a bioinformatist. He completed masters in molecular biology and did Ph.D. in bioinformatics and systems biology from Tehran University, Tehran, Iran. At present, he is a researcher in Molecular Biology Research Center, Baqiyatallah University of Medical Sciences, Tehran, Iran. His research areas are: microarray data analysis, bioinformatics software development, network and pathway reconstruction, and computational complex disease modeling. To date, he has published more than 20 scientific papers.

Reza Ranjbar is a microbiologist. He completed masters in medical microbiology from Shahid Beheshti University of Medical Sciences, Tehran, Iran and did Ph.D. in medical bacteriology from Tehran University of Medical Sciences, Tehran, Iran. At present, he is an associate professor and chairman of Molecular Biology Research Center, Baqiyatallah University of Medical Sciences, Tehran, Iran. His research areas are: molecular epidemiology of bacteria, molecular detection of bacteria, and Molecular epidemiology of antimicrobial resistance. To date, he has published more than 200 scientific papers.

Parham Moradi received the M.Sc. and Ph.D. degree in computer science from Amirkabir University of Technology, Iran, in 2005 and 2011, respectively. He conducted a part of his Ph.D. research work in the Laboratory of Nonlinear Systems, Ecole Polytechnique Federal de Lausanne (EPFL), Lausanne, Switzerland, from September 2009 to March 2010. He is currently an assistant professor in the department of computer engineering, University of Kurdistan, Iran. His research focuses on feature selection, recommender systems, reinforcement learning and social network analysis.