Journal of Universal Computer Science, vol. 25, no. 4 (2019), 418-443
submitted: 17/12/18, accepted: 21/3/19, appeared: 28/4/19

A Probabilistic Multi-Objective Artificial Bee Colony Algorithm for Gene Selection

Zeynep Banu Ozger
(Yildiz Technical University, Turkey
ozgerzeynep@gmail.com)

Bulent Bolat
(Yildiz Technical University, Turkey
bbolat@yildiz.edu.tr)

Banu Diri
(Yildiz Technical University, Turkey
diri@yildiz.edu.tr)

Abstract: Microarray technology is widely used to report gene expression data. The inclusion of many features and few samples is one of the characteristic features of this platform. In order to define significant genes for a particular disease, the problem of high-dimensionality microarray data should be overcome. The Artificial Bee Colony (ABC) Algorithm is a successful meta-heuristic algorithm that solves optimization problems effectively. In this paper, we propose a hybrid gene selection method for discriminatively selecting genes. We propose a new probabilistic binary Artificial Bee Colony Algorithm, namely PrBABC, that is hybridized with three different filter methods. The proposed method is applied to nine microarray datasets in order to detect distinctive genes for classifying cancer data. Results are compared with other well-known meta-heuristic algorithms: Binary Differential Evolution Algorithm (BinDE), Binary Particle Swarm Optimization Algorithm (BinPSO), and Genetic Algorithm (GA), as well as with other methods in the literature. Experimental results show that the probabilistic self-adaptive learning strategy integrated into the employed-bee phase can boost classification accuracy with a minimal number of genes.

Key Words: microarray, normalization, gene selection, machine learning, artificial bee colony
Category: I.2 I.2.6 H.3.2 L.3.2

1 Introduction

DNA (Deoxyribo nucleic acid) microarray technology involves microscopic DNA spots that are formed in sequence by observing thousands of genomic expression levels at the same time and are attached to a solid surface such as glass, plastic, or silicon chip. Measurement of gene expression using microarrays is feasible in many areas of biology and medicine. For example, microarrays can be used to identify disease-related genes by assessing gene expression in diseased and normal cells [Govindarajan et al. 2012]. Gene expression analysis includes two important steps: identifying disease-related genes and developing a classification model for unseen data, because each gene in microarray analysis allows evaluation of different tissue types [San Segundo-Val and Sanz-Lozano 2016]. When microarray data images are digitized, this reveals noisy data consisting of thousands of features. However, because they generally include few samples and many features, the process of clustering or classifying genes is difficult. To identify significant genes and analyze gene expression, it is critical to eliminate irrelevant features from microarray data.

Feature (gene) selection is a pre-processing step aims to improve performance and to facilitate classification and clustering processes [Guyon and Elisseeff 2003]. Feature selection is a kind of multi-objective optimization problem. Its goal is to minimize feature size and maximize classification or clustering performance. Feature selection methods are generally classified into three categories: filters, wrappers, and hybrid methods. In filter approaches, features are ranked according to a specific criterion like Chi squared test, information gain etc. and those with high rank are selected. Wrapper methods evaluate the performance of feature subsets using a learning algorithm. The features in subsets can be selected sequentially. In this way, all possible combinations of features are evaluated. Another strategy involves selecting feature subsets heuristically: without testing all of the combinations, the features that are likely to be successful are combined. Hybrid methods are formed by the combination of filter and wrapper methods [Tsamardinos and Aliferis, 2003].

Feature selection has a critical role in a wide range of applications such as gene expression analysis, image processing, and text mining etc. In microarray analysis, it is at challenge to, select the most important genes from a set of thousands. Statistical methods are inadequate because the number of samples is small.This situation led researchers to try different methods such as traditional feature selection methods and hybridized optimization algorithms. For a simple and fast solution, the feature selection process can be realized with filter and feature extraction methods. [Bolón et al. 2017], applied Mutual Information Maximization and Minimum Redundancy and Maximum Relevance (mRMR) algorithms in a distributed environment. [Aziz et al. 2016], combined Fuzzy Backward Feature Elimination and Independent Component Anaylsis (ICA). The hybrid model proposed by [Kalaviani and Kumar 2017], used Gaussian kernel approximation and constructed a fuzzy rough set model for selecting significant genes. [Sun et al. 2016] proposed a Lagrange Multiplier-based feature selection method with Support Vector Machines (SVM) classifier and compared their results with traditional filter methods. [Mortazavi et al. 2016] introduced a robust filter method that used qualitative mutual information.

Because of the high dimensionality of microarray data, hybrid methods that include a filter and a wrapper method are generally preferred. [Guo et al. 2016] applied Linear Discriminant Analysis (LDA) with Logistic Regression to data.

The class centroid is determined with kernel-based expectation. Logistic Regression is used as an optimization algorithm, and the class separability measure is used as the fitness function. The authors applied Partial Least Squares (PLS) for feature extraction after logistic regression in their next study [Guo et al. 2017]. The Discrete Bacterial Algorithm was also applied to the feature selection problem. The feature subset that was selected via individuals was restricted; therefore, no filter method was needed. The results were compared with some traditional methods and three evolutionary algorithms [Wang et al. 2017]. From the study of [Yang et al. 2008], datasets filtered with Information Gain (IG) and Correlation-Based Feature Selection (CFS), and subsequently feature subsets, were optimized with an Improved Binary Particle Swarm Optimization algorithm. The feature subset accuracies were evaluated with K Nearest Neighbor (K-NN) and SVM classifiers. [Abdi et al. 2012], used Particle Swarm Optimization (PSO) for two purposes: optimizing kernel parameters in SVM and weighting gene subsets after selecting the top n genes with the mRMR filter method. BinPSO was hybridized with rough set theory for the gene selection problem, as reported in [Dara et al. 2017]. As an improvement of Univariate Marginal Distibution Algorithm, which is an Estimation of Distribution-based algorithm, MOEDA, when combined with an mRMR filter, yielded promising results [Lv et al. 2016]. The Genetic Algorithm is a widely used evolutionary method, also employed for selecting important genes after filtering datasets with the mRMR filter [El Akadi et al. 2011]. GA has also been applied with the IG filter method [Yang et al. 2010]. Optimization methods can also be applied in a multi-objective way. There are two fitness values: classification accuracy and feature subset size. The former involves maximization, and the latter is a minimization problem. Microarrays are normalized with min-max normalization, and gene size is reduced by a correlation coefficient-based filter method. For the optimization process, a multi-objective GA is used at [Hasnat and Molla 2016]. [Tabakhi et al. 2015] utilized Ant Colony Optimization as a filter for unlabeled data; they did not employ a learning algorithm. The sum of mRMR values of the genes in the subset were normalized and used as the fitness function. In the ABC Algorithm proposed by [Alshamlan et al. 2015a] for finding discriminative genes, the authors improved upon their previous studies and proposed a method that combined GA with ABC [Alshamlan et al. 2015b]. They comparatively applied ABC, PSO, and Cuckoo Search (CS) with three fitness functions [Mohamed et al. 2017]. [Apolloni et al. 2016] proposed a Differential Evolution (DE) Algorithm with IG filter. MCSO [Mohapatra et al. 2016] is an improved version of the Cat Swarm Optimization (CSO) Algorithm that is used for the microarray gene selection process.

In this paper, we focus on improving an effective hybrid gene selection method for microarray data. Before the gene selection process, datasets are normalized using three microarray normalization methods: Robust Multi-Array Average (RMA) [Irizarry et al. 2003], Guanine Cytosine RMA (GcRMA) [Wu et al. 2004], and Micro Array Suite 5.0 (MAS5) [Hubbell et al. 2002]. According to purity and accuracy values, RMA yielded better results. Then, three filter methods in combination are applied to the normalized data. The top n features are selected for each filtered result, and these three sets are combined. Finally, the proposed binary ABC Algorithm is applied to the gene set selected by the filters to find the optimized feature subset. Results are compared to well-known meta-heuristic optimization methods. Additionally, we compared PrBABC results with some other gene selection methods in the literature. Consequently, we demonstrate that using the genes selected by three different filter algorithms increases the performance of the pre-processing step. Furthermore, the proposed binary version of the ABC Algorithm, because of its self-learning strategy, can easily adapt to the problem.

The remainder of this paper is structured as follows: Section 2 gives brief information about methods that used in this paper. Section 3 describes normalization phase, filtering phase and optimization phase of the proposed method. Datasets description, parameter settings and simulation results are given in Section 4. Finally, conclusion is drawn in Section 5.

2 Related Methods

2.1 Normalization

DNA microarray is a technology used to examine global changes in gene expression profiles in cells and tissues. Affymetrix GeneChips are a commonly used technology for expression profiling [Dalma et al. 2006]. The DNA sequences in microarrays are called 'probes'. Thousands of density values are associated with oligonucleotide probes, and these are grouped into probesets. Probe pairs include Perfect Match (PM) and Mismatch (MM) oligonucleotides. A Perfect Match is an exactly match with a particular gene; therefore, it measures the expression of this gene. A Mismatch contains a difference at the center of the sequence. MM probes find the correct transcript levels for genes that are expressed at low levels [Liu et al. 2010].

Transforming intensity values to expression values is accomplished by normalization. Normalization for microarray data aims to remove or minimize non-biological data at measured signal intensity levels. In this way, biological differences in gene expression can be appropriately detected [Quackenbush 2002]. Three common normalization methods are used for microarray data normalization: MAS5, RMA, and GcRMA. In MAS5 normalization, each probe is independently and sequentially normalized. It depends on the differences between perfect match and mismatch. The value of a probe-set summary is normalized using linear scaling based on a reference array in MAS5. RMA is a multichip linear model and uses only perfect matches; i.e., it ignores the mismatches. RMA values are in log2 units. The value of each probe is normalized using quantile normalization in multiple arrays. MAS5 results are effective for correlation analysis, whereas RMA results are effective for detecting differentially expressed genes. Additional removal of mismatches can cause the loss of important signals of many probes [Do and Choi 2006]. The difference between GC-RMA and RMA is that RMA uses a convolutional model for background correction, but GcRMA uses the Guanine Cytosine (GC) content of the probes. In this way, it is intended to reduce the variance in the MM probe levels. According to position-dependant base effects, probe affinity is calculated. After MMs are adjusted based on probe affinity, they are subtracted from PM so that the MM values are not lost. GcRMA does not keep the probe-level information, and reports one value for each probe set [Wu et al. 2004].

2.2 Information Gain

Information Gain is an entropy-based feature ranking method. Entropy is a measure of purity of a sample. The information gain is the difference of the information using to recognize to feature Y (Eq. (1)) and the information about feature Y after observing X (Eq. (1)). As a feature selection method, IG measures the distinctiveness of a feature for a given class. The value of IG is the range of 0-1. The more independent the feature is from the class information, the closer the value of IG is to 0 [Hall 1999].

$$H(Y) = -\sum_{y \in Y} p(y) log(p(y)) \tag{1}$$

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) log(p(y|x)) \tag{2}$$

$$IG = H(Y) - H(Y|X) \tag{3}$$

where p(y) is the marginal probability density function for the random variable Y. p(y|x) is the conditional probability of y given x.

2.3 Correlation-Based Feature Selection

Correlation-Based Feature Selection is based on the hypothesis that good feature subsets are composed of features that are high in relation to the related class, and have low correlation with each other. CFS uses a search algorithm with a function that measures the information values of the feature subsets. The ‘correlation’ term refers to ‘measure of feature similarity’. The aim is to select the features that are high in relation to a particular class and low in relation to other features. The correlation value is a range between -1 and 1. The '-1' value indicates that there is a full negative linear relation between features, whereas the '1' value indicates a full positive linear relation between features. If the value is '0', there is no relation [Hall 1999].

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{4}$$

where k is number of features at subset, $\overline{r_{cf}}$ is the mean feature-class correlation, $\overline{r_{ff}}$ is the average feature-feature inter-correlation.

2.4 ReliefF

ReliefF is a statistical-based filter method. It takes a sample from a dataset and creates a model according to its closeness to the samples in the same class and distance from the samples in other classes. Therefore, ReliefF aims to maximize the margins that separate the classes [Robnik-Sikonja and Kononenko 2003]. At the beginning of the algorithm, weights for all attributes are 0. The algorithm selects a random observation and iteratively updates the weights according to Eq. (5).

$$S_i = \frac{\sum_{j=1}^{m} -diff(A_{ij}, H_{ij}) + diff(A_{ij}, C_{ij})}{m} \tag{5}$$

where i is the corresponding feature, m is the number of samples in dataset, $diff(A_{ij}, H_{ij})$ is the distance between the sample A_{ij} and the nearest sample H_{ij} in same class. $diff(A_{ij}, C_{ij})$ is the distance between the sample A_{ij} and the nearest sample C_{ij} in different class.

2.5 Artificial Bee Colony

[Karaboga and Akay, 2009] introduced at 2009, the Artificial Bee Colony Algorithm, which is a swarm-based heuristic method. ABC is a multi-dimensional optimization algorithm that imitates the foraging behaviour of bees. According to this algorithm, the purpose of the bees is to maximize the quantity of nectar sources and minimize the distance of the sources. For optimization problems, sources are represented by vectors. The vector dimension is the parameter number of the problem. Each source is a possible solution of the problem, and the source quality is represented by the amount of nectar; this is called the fitness value. Each source has a trial value and is set to 0 in the initialization phase. When a source is improved, this value remains the same; otherwise, it is incremented by 1. A colony has three kinds of bees: employed bees, onlooker bees, and scout bees. There are an equal number of employed and onlooker bees, whereas there is always a single scout bee. The algorithm includes four steps:

1. Initialization Phase: Sources are randomly initialized using Eq. (6).

$$x_{ij} = x_j^{min} + U(0,1)(x_j^{max} - x_j^{min}) \tag{6}$$

The current source number is represented by i, and j is the dimension of the vector. U(0,1) is a uniformly distributed random number.

2. Employed Bee Phase:Each employed bee selects a random neighbour, then using Eq. (7) generates a new solution. The fitness values of the new and current solutions are compared. If the former is better, it becomes the current solution; otherwise the trial value is incremented by 1.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{7}$$

x_i and x_k are current and randomly selected neighbour sources respectively. Ø is uniformly distributed random value between [-1,1].

3. Onlooker Bee Phase: The duty of onlooker bees in a swarm is optimization. After the employed bees exploitation phase, they return to the hive and provide information about their own sources. Each onlooker bee selects a source according to this information using the roulette-wheel scheme given in Eq. 8.

$$p_i = \frac{fitness_i}{\sum_{j=1}^{n} fitness_j} \tag{8}$$

p_i is the probability of i^{th} source. If this probability is greater than a random value this source is selected otherwise the trial number of i^{th} source is incremented by 1.

4. Scout Bee Phase: At the end of the employed bee and onlooker bee phases, the scout bee checks if the trial value has exceeded the 'limit' value by any source. If so, it is assumed that this source fell into a local optimum; therefore, it is abandoned and the scout bee generates a new source using Eq. (6).

3 Proposed Method

3.1 Normalization

Normalization is one of the key processes used in data mining. If the mean and variance of the variables are significantly different from each other, the variables with large mean and variance have a higher pressure on the others, and their role is significantly reduced. Therefore, data normalization should be performed. In order to reduce the effect of these factors, we firstly normalized microarray datasets by using MAS5, RMA, and GcRMA normalization methods for microarray data. The normalization process is applied using the R statistical programming language, employing the Bioconductor package, which is a commonly used tool for genomic data. In order to compare datasets that normalized using three different methods, purity, Silhouette coefficient, and accuracy measures are used. Purity and Silhouette coefficient are criteria that indicate the quality of clusters. Purity is calculated as the ratio of the number of samples in the correctly assigned cluster to the total number of samples. The Silhouette coefficient measures the similarity of each sample to its cluster and is calculated with the mean nearest-cluster distance and mean of the intra-cluster distance. Normalized datasets are clustered using K-Means and Hierarchical Clustering methods. Clustering and Classification algorithms were applied with MATLAB R2017a.

3.2 Filtering

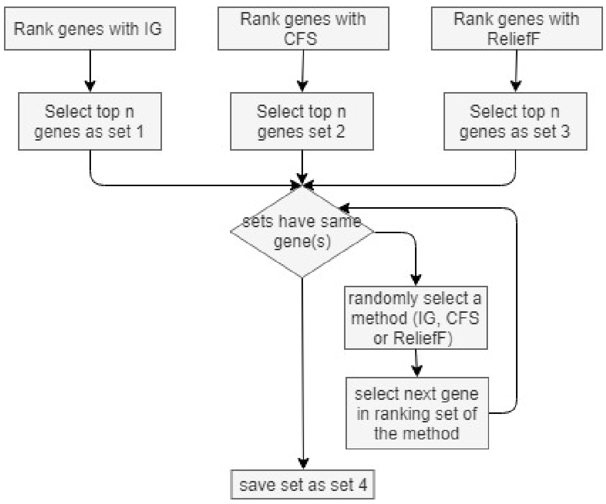


Figure 1: Flow Chart of Filtering Process

One of the most typical characteristic properties of microarray is that it has thousands of features and very few samples. Filtering methods that work by reducing feature size provide a preliminary process for determining the distinguishing features. For this purpose, we applied three filter methods to datasets before the optimization algorithm was applied. These are entropy-based Information Gain, Correlation Based Feature Selection, and distance-based ReliefF algorithms. Subsequently, we combined the results of all three methods. Filtering process is given in Figure 1. Genes are ranked according to these filter methods separately. Top n genes are selected from each ranked sets. Subsequently, it is checked whether there is/are same gene(s) in the clusters. If any the next gene (n+1) is taken from randomly selected ranked set. These filtering algorithms ranks genes according to different criteria. In this way, after the filtering process is complete there are 3n features that have different levels of discrimination in datasets obtained by three different filter methods. Filter algorithms were applied with MATLAB R2017a.

3.3 Probabilistic Binary Artificial Bee Colony Optimization

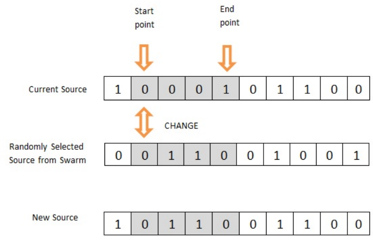
The ABC Algorithm is proposed for continous optimization problems. However, gene selection requires a binary solution space. The binarization process is accomplished by two basic approaches: to use binary vectors and to edit the new source generation equation to work with binary vectors or transforming continous values to binary space with a transformation function. Similarity measures (Jackard etc.) [Kasha et al. 2012], bitwise operators (AND, OR, XOR etc.) [Jia et al. 2014, Kiran and Gündüz 2013], insertion and/or swap operators [Zhang and Gu, 2015], substitution and/or shift operations [Ozmen et al. 2018] and genetic operators (cross-over, etc.) [Ozturk et al. 2015, Ozturk et al. 2014, Yurtkuran and Emel 2016] are used for the former approach. The round function [Wei and Hanning, 2012], sigmoid function [Tran and Wu 2014], mod operator [Kiran 2015] and, tangent function [Mandala and Gupta 2014] are used for transformation of continuous values to binary space.

Feature selection is a binary optimization problem. Certain assumptions are necessary for binary ABC variations: Food sources are binary vectors and their size is equal to the number of features of the dataset. If a feature is included in a feature subset, the corresponding value of the vector is 1; otherwise, it is 0. The fitness value is the classification/clustering performance of the selected subset.

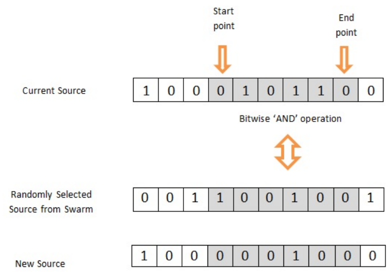
For the ABC algorithm, sources are real vectors and are initialized by using Eq. (6). We adopted an initialization phase that is suitable for binary space, and sources are initialized with the following rule:

$$x_{ij} = \begin{cases} 1 & \text{if } rand(0,1) \geq threshold \\ 0 & \text{if } rand(0,1) < threshold \end{cases} \tag{9}$$

The learning strategy is a critical issue for an optimization algorithm; it is highly dependent on the problem or data used. In ABC, the learning strategy is a new source generation formula that is given by Eq. (7). Researchers have proposed various novel source generation formulas for the employed bee phase in binary ABC variations. Feature selection can be defined as a multi-objective optimization problem. It aims to reduce the number of features while increasing the accuracy. At the stage of the binarization of the ABC algorithm, we tested



(a)



(b)

Figure 2: New source generation procedures (a) cross-over (b) bitwise AND

various learning methods. Some of them achieved improved classification accuracy and some of them yielded reduced feature size. Therefore, inspired by the study [Qin et al. 2005], we used two learning strategies in a probabilistic manner; one is intended to increase accuracy, and the other is intended to reduce feature size.

We applied some learning strategies to fit binary ABC and saw from our experiments that cross over and bitwise operators produce promising results, and have widespread use for the binarization process of ABC. In ABC, each current source contributes to the knowledge of other sources. Therefore, we constructed our learning strategies according to this information sharing. To increase accuracy, we used a partially-mapped crossover operator in the employed bee phase. According to this, a random source from a swarm is selected, and the applied partial-mapped crossover operator is shown in Figure 2. In the partial-mapped crossover operation, two points are selected randomly, then sub-strings between these points are exchanged. So, a new source is generated for the employed bee phase. If new source has a better fitness value than the current one, the new source becomes the current source. This means that the current source is improved. The advantage of the crossover operation is that the sub-strings are combined with different sub-strings in this way. These combinations can produce better classification results because in optimization algorithms, superior solutions has a better chance of survival. Additionally, crossover provides diversity and prevents falling into a local optimum.

We used bitwise operators as a second learning strategy. Similarly, a random source is selected from the swarm and the bitwise 'AND' operator is applied to sub-strings of sources, as shown in Figure 2b.

One of the these two learning strategies is probabilistically selected by the algorithm according to performance. When the algorithm is initialized, both of them have the same probability of selection: 0.5. As the iterations progress, these probabilities are updated according to the rate of improvement of fitness value using Eq. (10).

$$p_1 = \frac{imS_1}{\#totalImprovement}, p_2 = 1 - p_1 \tag{10}$$

p_1 and p_2 are the probability of the first and second learning strategy respectively. imS_1 is the number of successful new sources that generated by the first learning strategy. For each source in the employed bee phase and for some sources at the onlooker bee phase of ABC, new sources are generated. If it is better than the previous one, they are exchanged. imS_1 is incremented by 1 if it improves the source when it selected as the learning strategy. Similarly, if the second strategy is selected and improves the source, imS_2 is incremented by 1. $\#totalImprovement$ is the total success number of both strategies. The total probability of the two strategies is 1.

imS_1, imS_2, $\#totalImprovement$, p_1 and p_2 values are updated at the end of each iteration.

A uniformly distributed random number between 0 and max(p_1,p_2) is generated in each new resource generation stage. If this random number is smaller than p_1 a new source is generated by the first learning strategy; if it is smaller than p_2 a new source is generated by the second learning strategy. If the random number is smaller than both of the probabilities, the algorithm randomly selects one of the two methods. We experimentally saw that, if a strategy improved sources several times in an iteration, in the next iteration its probability is very high, according to the other strategy. In this way, we prevented such obvious differences in probabilities.

In the employed bee phase, we used the improvement capability of learning strategies. Similarly to the onlooker bee phase, we used failure rates of learning strategies. Unlike before, the probabilities were calculated using Eq. (11), and a lower probability value, indicated less failure. Namely, the selection chance of a strategy is increased, according to the magnitude of the random value. Thus, the algorithm takes into account both improvement and failure rates. We call our method "Probabilistic Binary Artificial Bee Colony (PrBABC)", and to demonstrate its effectiveness we applied it to the gene selection problem.

$$p_3 = \frac{\#failure_1}{\#totalFailure}, p_4 = 1 - p_3 \tag{11}$$

4 Experiments

We used nine differently sized microarray datasets listed in Table 1, in our experiments to measure the effectiveness of the proposed methods. All datasets are high-dimensional data, and the feature size is much larger than the sample size.

Microarray normalization methods were compared according to purity, accuracy, and silhouette coefficient values. Purity values were measured according to

Table 1: Datasets that used in experiments

MA	# Ins	# Cls	# Feat	Description	MA	# Ins	# Cls	# Feat	Description
Cll-Sub	111	3	12626	Subgroups of B-cell chronic lymphocytic leukemia [NCBI]	Stjude	327	7	12625	Sub type of pediatric lymphoblastic leukemia [St.Jude]
Dlbcl	77	2	7129	Diffuse Large B-Cell Lymphoma [Broad Institute]	Lung Can	235	5	12625	Human Lung Carcinomas [Broad Institute]
Glim.	50	2	12625	Brain Tumor [NCBI]	Leuk2	72	3	12626	Sub types of Leukemia [Broad Institute]
Pros Can.	102	2	12625	Prostate tumor versus normal genes [Broad Institute]	WBC	97	2	24482	Breast cancer relapse free survival [NCBI]
CNS	60	2	7129	Central nervous system embryonel tumour outcome [Broad Institute]					

K-Means and Hierarchical clustering methods. In order to establish a baseline to assess the clustering results, the clustering process was applied 10 times, and mean purity values were taken into account for K-Means. Silhouette coefficients were calculated according to clusters produced by the K-Means Algorithm. The class number of datasets is set as the cluster number. To obtain the accuracy values, the 3-NN algorithm was applied with 10-fold cross-validation. Clustering and Classification algorithms were applied in the MATLAB R2017a environment. The obtained results are given in Table 2.

Table 2: Normalization methods comparison

		Purity		Accur.	Sil. Co-eff.		Purity		Accur.	Sil. Co-eff.
Dataset	Method	K-Means	Hierar	3NN	K-Means	Dataset	K-Means	Hierar	3NN	K-Means
Cll-Sub	MAS5	0.252	**0.513**	0.829	**0.491**	Lung	0.302	0.715	0.962	0.154
	RMA	0.423	0.468	**0.892**	0.217		**0.506**	0.715	**0.983**	0.168
	GcRMA	**0.577**	0.459	0.883	0.251		0.451	**0.723**	0.974	**0.187**
CNS	MAS5	0.617	0.633	0.783	0.161	WBC	0.507	0.622	**0.787**	0.172
	RMA	0.617	0.633	**0.817**	**0.252**		**0.535**	0.622	**0.78**	**0.183**
	GcRMA	**0.683**	0.633	0.783	0.208		0.528	**0.629**	0.773	0.111
DLBCL	MAS5	0.61	0.74	0.948	0.174	Leuk2	0.444	0.389	0.944	**0.193**
	RMA	**0.714**	0.74	**0.974**	**0.182**		**0.667**	0.389	**0.986**	0.153
	GcRMA	**0.714**	0.74	0.948	0.131		0.639	0.389	0.972	0.109
Stjude	MAS5	0.177	0.231	0.865	**0.125**	Prostate	**0.627**	0.5	0.892	0.373
	RMA	**0.217**	**0.24**	0,939	0.056		0.569	0.5	**0.931**	0.358
	GcRMA	**0.217**	**0.24**	**0.96**	0.066		0.598	0.52	0.912	**0.391**
Glioma	MAS5	0.6	0.54	0.88	0.12					
	RMA	**0.72**	0.54	0.88	0.147					
	GcRMA	**0.72**	0.54	0.88	**0.29**					

According to the results in Table 2, the purity values obtained by hierarchical clustering are generally equal to each other. For some datasets, the purity values obtained by K-Means clustering are also equal. Generally, for this, measure RMA and GcRMA normalization methods yielded good clustering results. With respect to accuracy, RMA normalization is better than the other methods. When the silhouette coefficients are taken into account, the methods show equivalent performance. Overall, we saw that, RMA produced generally good results for these datasets. Therefore, we used datasets normalized with RMA method in our experiments.

In the optimization phase, datasets were classified with the 5-NN algorithm. Datasets were randomly divided into 3 sections: 60% of the data was the training set, 20% of the data was the validation set, and the remaining data is the test set. It is guaranteed that each set has at least one sample of each class. During the optimization phase, the selected subset is trained with the training set, and its accuracy value is measured with the test set. The samples in the validation and test sets were not used in the training phase. After all iterations had completed, the performance of the best feature subset was measured by the test set.

PrBABC was compared with three well-known algorithms: Genetic Algorithm [Sivanandam S. and Deepa 2008], Binary Particle Swarm Optimization Algorithm [Kennedy and Eberhart 1997], and the binary Differential Evolution Algorithm (BinDE) [Engelbrecht and Pampara 2007]. To ensure a fair comparison, 3 filter methods were applied to datasets as in PrBABC before the algorithms were run. The results were compared in terms of test set misclassification errors, number of genes in the subset, and CPU time. Algorithms are applied according suggestions; for GA in [Sivanandam S. and Deepa 2008] for BPSO in [Kennedy and Eberhart 1997] and for BinDE in [Engelbrecht and Pampara 2007] using MATLAB r2017a. The source code available for all algorithms at `github.com/ZBaOz/MicroarrayGeneSelection`.

The threshold value in the initialization phase was set as 0.85; this value represents the probability of a gene in the subset. The limit value that was used in the scout bee phase was 100. The number of employed bees was 25, and the population size for GA, BinPSO, and BinDE was 50 (2 x #employed-Bee). For GA, crossover rate was 0.8 and mutation rate was 0.2, as reported in [Sivanandam S. and Deepa 2008]. As recommended in [Mirjalili and Lewis 2012], the learning factors c1 and c2 were set to 2, initial weight (w) was set to 0.9, and the maximum speed (V_{max}) was set to 0.6 for BinPSO. For BinDE the scaling factor (F) was set to 1, perturbation parameter (P) was set to 0.25, and cross-over rate (CR) was set to 0.1 as recommended in [Engelbrecht and Pampara 2007]. Maximum iteration number for all algorithms was 100. Algorithms ran 20 times with different random seeds and were averaged to verify the results statistically. The algorithms were executed on a PC with 16 GB Ram and an Intel (R) Core(TM) i7-3630QM 2.4 GHz CPU by using MATLAB R2017a. The source code available at `github.com/ZBaOz/MicroarrayGeneSelection`.

Exploration in ABC is done in the scout bee phase and is controlled by the 'limit' value; therefore this value directly affects the performance of the ABC algorithm. [Karaboga and Basturk 2008] proposed to determine the limit value according to the following formula:

$$limit = n_e * D \tag{12}$$

n_e is the number of employed bees and D is the dimension of the problem. However, in [Veček et al. 2017], the authors tried different n_e and D value pairs, and they demonstrated that there does not exist a linear relationship between n_e and D as in Eq. (12), especially in high-dimensional problems. In the gene selection process, problem dimension is equal to the gene number of dataset. For example, for a microarray with 10,000 genes, the problem dimension is equal to 10,000. Determining the 'limit' value according to Eq. (12), will significantly reduce exploration. 'Limit' is a problem-dependent parameter; therefore, we set it to 100 by trial and error.

The fitness value is calculated according to accuracy and the number of gene values of the subset. The constant 'c' in Eq. (13) is used for regulating the weights of accuracy and gene number. In our study, since increasing accuracy is more important than reducing the number of genes, c was set as 0.9995. Thus, it is guaranteed that the algorithm always selects a subset that has high accuracy, but if any two subsets have identical accuracy, PrBABC selects the subset that has fewer genes.

$$fitness = (c * accuracy) + ((1 - c) * \frac{\#selectedGene}{\#totalGene}) \tag{13}$$

$$accuracy = \frac{\#correctlyClassifiedInstances}{\#totalInstances} \tag{14}$$

The maximum iteration number is an important parameter of an iterative algorithm, and it is a problem-dependent parameter. The difficulty of the problem is directly related to the size of the search space. If this value is small, the algorithm cannot converge to the optimal solution. A characteristic of microarrays is that they have many features and a few samples. Therefore, the validation and test sets have few samples for each class. When the maximum iteration number was too large, we saw that, initially, the error decreased, but subsequently, the error began to increase because of over-training. Taking this into consideration, the maximum iteration number was experimentally chosen as 100. Figure 3 shows the convergence graph related to test set misclassification error of PrBABC-3F and iteration number, from a randomly selected run with the top 1,500 features. According to these graphs, 100 iterations are sufficient, and the algorithm is terminated without over-training.

The IG, CFS, and ReliefF filter methods were applied to the microarrays and the top 1500, 1000, 750, 500, 300, and 100 genes were selected. PrBABC

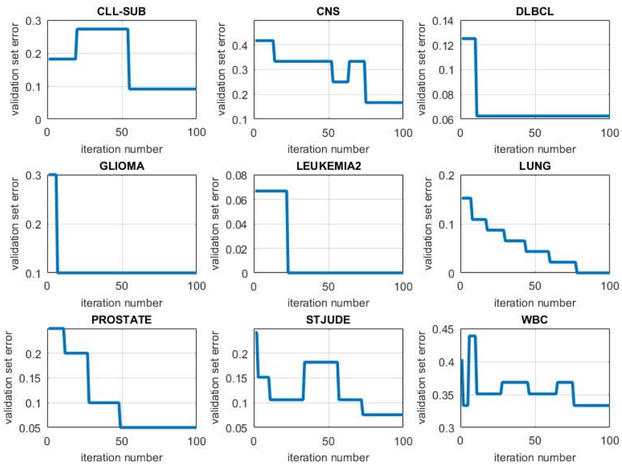


Figure 3: Convergence Graph

was applied after the datasets were filtered by three filter methods both combined and separately. PrBABC-3F represents filtering process performed with all three filtering methods. For example, as mentioned above, the top 1500 genes are composed of the top 500 genes obtained by each filter method. PrBABC-IG, PrBABC-CFS, and PrBABC-RF represent IG, CFS, and ReliefF filters with PrBABC, respectively. The test set misclassification error results and their standard deviations of methods are given in Table 3. The best results for each dataset are shown in bold text.

According to the results shown in Table 3, applying three filter methods together is better than the other strategies for most of the datasets. For DLBCL and Leukemia2 datasets, test set misclassification errors were generally smaller than 0.1, and other methods yielded better results for some conditions; however the error differences are very small and not meaningful. When 3 filter methods were compared with each other, ReliefF produced better results than IG and CFS, and it obtained the closest results to the PrBABC-3F method. ReliefF aims to maximize the margins that separate classes. The results clearly show that applying three filtering methods together selects the gene subset with the fewest errors. In this way, we took advantage of these 3 different methods.

To show the effectiveness of PrBABC, the results are compared with the GA, Binary PSO, and Binary DE algorithms. As in PrBABC, the datasets were filtered by all 3 methods, and the top 1500, 1000, 750, 500, 300, and 100 genes were selected.

GA-3F represents Genetic Algorithm with 3 filters, BinPSO-3F represents

Table 3: Misclassification Error Comparison for Filter Methods

	Method	CLL-SUB	CNS	DLBCL	Glioma	Leuk2	Lung	Prostate	Stjude	WBC
1500	PrBABC-3F	**0.259± 0.1**	**0.242± 0.08**	**0.031± 0.04**	**0.14± 0.1**	**0.03± 0.04**	**0.042± 0.03**	**0.1± 0.07**	**0.125± 0.04**	**0.325± 0.05**
	PrBABC-CFS	0.355± 0.12	0.367± 0.1	0.081± 0.06	0.255± 0.14	0.083± 0.09	0.076± 0.03	0.165± 0.07	0.174± 0.04	0.383± 0.06
	PrBABC-IG	0.289± 0.09	0.383± 0.14	0.044± 0.05	0.205± 0.14	0.053± 0.05	0.053± 0.03	0.153± 0.05	0.131± 0.04	0.372± 0.05
	PrBABC-RF	0.284± 0.1	0.308± 0.09	0.069± 0.06	0.16± 0.1	0.06± 0.06	0.053± 0.03	0.123± 0.07	0.139± 0.03	0.361± 0.04
1000	PrBABC-3F	**0.227± 0.1**	**0.279± 0.13**	0.031± 0.05	**0.15± 0.09**	**0.023± 0.03**	**71.85± 8.23**	**0.095± 0.06**	**0.116± 0.05**	**0.333± 0.04**
	PrBABC-CFS	0.286± 0.1	0.375± 0.17	0.116± 0.08	0.285± 0.13	0.1± 0.06	0.08± 0.03	0.21± 0.07	0.22± 0.05	0.397± 0.06
	PrBABC-IG	0.314± 0.09	0.35± 0.01	**0.028± 0.03**	0.165± 0.11	0.033± 0.05	0.053± 0.03	0.143± 0.07	0.13± 0.04	0.368± 0.05
	PrBABC-RF	0.275± 0.11	0.313± 0.11	0.059± 0.05	0.2± 0.11	0.067± 0.05	0.057± 0.03	0.122± 0.07	0.142± 0.04	0.348± 0.04
750	PrBABC-3F	**0.223± 0.11**	**0.263± 0.09**	**0.028± 0.04**	**0.12± 0.08**	**0.013± 0.03**	**0.053± 0.03**	**0.09± 0.06**	**0.109± 0.05**	**0.33± 0.04**
	PrBABC-CFS	0.354± 0.11	0.45± 0.12	0.138± 0.09	0.305± 0.14	0.137± 0.08	0.077± 0.04	0.22± 0.07	0.219± 0.04	0.4± 0.06
	PrBABC-IG	0.282± 0.09	0.371± 0.12	0.031± 0.05	0.21± 0.16	0.033± 0.05	0.05± 0.03	0.1± 0.06	0.138± 0.05	0.384± 0.05
	PrBABC-RF	0.282± 0.09	0.283± 0.08	0.034± 0.04	0.165± 0.12	0.073± 0.07	**0.053± 0.03**	0.14± 0.09	0.145± 0.04	0.365± 0.04
500	PrBABC-3F	**0.234± 0.1**	**0.263± 0.12**	0.047± 0.05	**0.12± 0.11**	**0.017± 0.04**	**0.046± 0.03**	**0.08± 0.06**	**0.127± 0.04**	**0.339± 0.06**
	PrBABC-CFS	0.302± 0.12	0.383± 0.14	0.125± 0.06	0.33± 0.13	0.163± 0.11	0.112± 0.04	0.2± 0.08	0.23± 0.05	0.395± 0.06
	PrBABC-IG	0.314± 0.12	0.375± 0.12	0.047± 0.05	0.17± 0.13	0.04± 0.06	0.054± 0.03	0.118± 0.06	0.137± 0.05	0.414± 0.05
	PrBABC-RF	0.286± 0.1	0.271± 0.09	**0.037± 0.05**	0.185± 0.09	0.057± 0.05	0.066± 0.03	0.135± 0.08	0.162± 0.03	0.341± 0.05
300	PrBABC-3F	**0.241± 0.09**	**0.25± 0.14**	0.037± 0.05	**0.155± 0.09**	0.04± 0.05	**0.059± 0.03**	**0.087± 0.07**	**0.13± 0.05**	**0.328± 0.05**
	PrBABC-CFS	0.352± 0.1	0.404± 0.12	0.197± 0.1	0.285± 0.1	0.187± 0.1	0.079± 0.04	0.223± 0.09	0.275± 0.05	0.428± 0.06
	PrBABC-IG	0.266± 0.08	0.363± 0.15	**0.034± 0.05**	0.19± 0.12	**0.037± 0.034**	0.068± 0.04	0.113± 0.09	0.148± 0.03	0.43± 0.05
	PrBABC-RF	0.268± 0.1	0.267± 0.1	0.041± 0.04	0.16± 0.11	0.043± 0.05	**0.059± 0.03**	0.113± 0.07	0.151± 0.03	0.351± 0.04
100	PrBABC-3F	**0.209± 0.06**	**0.233± 0.08**	**0.047± 0.05**	**0.145± 0.14**	**0.043± 0.05**	**0.07± 0.04**	**0.068± 0.08**	**0.144± 0.04**	**0.337± 0.04**
	PrBABC-CFS	0.418± 0.13	0.483± 0.15	0.2± 0.07	0.335± 0.16	0.223± 0.14	0.128± 0.04	0.237± 0.11	0.398± 0.06	0.43± 0.05
	PrBABC-IG	0.275± 0.12	0.304± 0.11	0.081± 0.05	0.165± 0.12	0.077± 0.08	0.07± 0.04	0.137± 0.08	0.18± 0.05	0.402± 0.06
	PrBABC-RF	0.289± 0.09	0.271± 0.1	0.069± 0.07	0.16± 0.13	0.087± 0.09	0.105± 0.05	0.143± 0.08	0.166± 0.04	0.342± 0.05

the BinPSO algorithm with 3 filters, and BinDE-3F represents the Binary Differential Evolution Algorithm with 3 filters. Algorithms are compared in terms of test set misclassification errors (Table 4) and gene set size (Table 5). In the Table 4, the first row shows test set error of accuracy values and their standard deviations. The second row shows statistical significance obtained by Wilcoxon Rank Sum Test. '+' indicates that the PrBABC results are statistically better than the corresponding algorithm, whereas '-' indicates that the PrBABC results are statistically worse than the corresponding algorithm. '=' shows that there is no statistical significance between algorithms. The number of selected genes of the algorithms and their standard deviations are shown in Table 4. The best results are shown in bold text.

It is clearly shown from Tables 4 and 5 that PrBABC outperforms the other algorithms, in terms of test set error, for the majority of datasets. As in Table 3, other methods produced better results for some gene set sizes for the DLBCL, Leukemia 2, and Lung datasets, but due to the simplicity of these datasets, all algorithms were able to classify with almost 100% accuracy effectively, the differences among these algorithms are not meaningful. When the accuracy values were evaluated together with the gene set size, PrBABC was easily able to eliminate non-distinctive genes and achieve better or equivalent classification accuracy with a smaller number of genes for all datasets. There are two learning methods in PrBABC. While choosing the learning methods, we paid attention to the fact that they were able to find differentiated groups of genes. The results in 4 and 5 show that we were able to realize our goal.

To decide how many genes are sufficient for each dataset, we performed a gene selection process with the different number of genes: 1500, 1000, 750, 500, 300, and 100. This is important to establish a balance between the number of genes and test set errors. As the size of the dataset decreases, the number of selected genes is decreasing, but the test error does not always increase. For DLBCL, Leukemia2, and Lung, error results are close to each other for all dataset sizes; therefore, starting with the top 100 genes is sufficient. For the CLL-SUB, CNS, and Prostate Cancer datasets, PrBABC yielded the best performance with the top 100 genes. For the remaining datasets, WBC, Stjude, and Glioma, the top 300 or 750 genes are sufficient for PrBABC to select an efficient subset.

In Table 6, we compared PrBABC with other gene selection methods in the literature according to classification accuracy and number of selected genes. The numbers in parentheses indicate the numbers of selected genes. We note that there may be some differences among datasets. We use the datasets in raw data format without any feature selection or normalization method. In some sources, the same dataset is available with different numbers of genes, samples, or classes; in such cases, we ignored the difference in the number of genes and samples. However, we paid attention to the fact that the datasets we intended to compare contain the same number of classes. Only the DLBCL dataset that we used is the same size as the DLBCL dataset in the other studies. Additionally, there were also some differences in the parameters and train-test sizes. Therefore, we could not perform a 1 to 1 comparison, but our results nevertheless provide information about general trends. We compared these methods with the average validation set accuracy results of PrBABC because most of these methods did not

Table 4: Misclassification Error Comparison for Evolutionary Algorithms

		CLL-SUB	CNS	DLBCL	Glioma	Leuk2	Lung	Prostate	Stjude	WBC
1500	Pr BABC	**0.259 ±0.1**	**0.242 ±0.08**	**0.031 ±0.04**	**0.14 ±0.1**	0.03 ±0.04	0.042 ±0.03	**0.1 ±0.07**	**0.125 ±0.04**	**0.325 ±0.05**
	GA	0.268 ±0.08	0.354 ±0.09	0.044 ±0.06	0.195 ±0.12	0.047 ±0.05	0.051 ±0.03	0.14 ±0.07	0.142 ±0.04	0.338 ±0.05
		+	+	+	+	-	=	-	+	+
	Bin PSO	0.273 ±0.09	0.383 ±0.14	0.106 ±0.1	0.355 ±0.19	0.037 ±0.05	0.041 ±0.031	0.207 ±0.17	0.141 ±0.03	0.367 ±0.054
		+	+	+	+	+	+	+	+	+
	Bin DE	0.264 ±0.11	0.308 ±0.11	0.041 ±0.04	0.17 ±0.11	**0.027 ±0.03**	**0.041 ±0.03**	0.123 ±0.06	0.14 ±0.03	0.362 ±0.04
		+	+	+	+	+	+	=	+	+
1000	Pr BABC	**0.227 ±0.1**	**0.279 ±0.13**	**0.031 ±0.05**	**0.15 ±0.09**	0.023 ±0.03	**0.035 ±0.03**	**0.095 ±0.06**	**0.116 ±0.05**	**0.333 ±0.04**
	GA	0.273 ±0.12	0.321 ±0.11	0.063 ±0.06	0.2 ±0.07	**0.02 ±0.03**	0.047 ±0.033	0.125 ±0.07	0.133 ±0.03	0.358 ±0.05
		+	+	=	+	+	+	+	+	+
	Bin PSO	0.284 ±0.11	0.367 ±0.14	0.113 ±0.13	0.42 ±0.16	0.043 ±0.1	0.052 ±0.034	0.155 ±0.1	0.153 ±0.04	0.359 ±0.04
		=	+	+	+	+	+	+	+	+
	Bin DE	0.284 ±0.11	0.338 ±0.09	0.041 ±0.04	0.195 ±0.13	0.023 ±0.04	0.043 ±0.03	0.12 ±0.06	0.131 ±0.04	0.36 ±0.03
		+	+	-	+	+	+	+	+	+
750	Pr BABC	**0.223 ±0.11**	**0.263 ±0.09**	**0.028 ±0.04**	**0.12 ±0.08**	**0.013 ±0.03**	0.053 ±0.03	**0.09 ±0.06**	**0.109 ±0.05**	**0.33 ±0.04**
	GA	0.259 ±0.08	0.337 ±0.11	0.053 ±0.06	0.19 ±0.09	0.03 ±0.04	0.053 ±0.03	0.12 ±0.06	0.13 ±0.03	0.369 ±0.06
		+	+	=	+	+	+	+	+	+
	Bin PSO	0.284 ±0.11	0.4 ±0.12	0.094 ±0.12	0.295 ±0.15	0.03 ±0.05	**0.049 ±0.02**	0.202 ±0.13	0.153 ±0.04	0.363 ±0.06
		+	+	+	+	+	+	+	+	+
	Bin DE	0.277 ±0.1	0.287 ±0.12	0.041 ±0.05	0.245 ±0.11	0.023 ±0.04	0.053 ±0.03	0.118 ±0.06	0.127 ±0.04	0.343 ±0.06
		+	+	-	+	+	+	+	+	+
500	Pr BABC	**0.234 ±0.1**	**0.263 ±0.12**	**0.047 ±0.05**	**0.12 ±0.11**	**0.017 ±0.04**	0.046 ±0.03	**0.08 ±0.06**	**0.127 ±0.04**	**0.339 ±0.06**
	GA	0.257 ±0.09	0.304 ±0.12	0.063 ±0.05	0.185 ±0.13	0.03 ±0.03	0.049 ±0.03	0.118 ±0.06	0.145 ±0.06	0.361 ±0.06
		+	+	-	+	+	+	+	+	+
	Bin PSO	0.282 ±0.12	0.333 ±0.1	0.091 ±0.13	0.37 ±0.22	0.057 ±0.13	**0.03 ±0.02**	0.133 ±0.11	0.146 ±0.03	0.36 ±0.05
		+	+	+	+	-	+	+	+	+
	Bin DE	0.268 ±0.09	0.296 ±0.12	0.053 ±0.05	0.195 ±0.12	0.017 ±0.03	0.049 ±0.03	0.113 ±0.05	0.128 ±0.03	0.366 ±0.05
		+	+	=	+	+	-	+	+	=
300	Pr BABC	**0.241 ±0.09**	**0.25 ±0.14**	**0.037 ±0.05**	**0.155 ±0.09**	0.04 ±0.05	0.059 ±0.03	**0.087 ±0.07**	**0.13 ±0.05**	**0.328 ±0.05**
	GA	0.259 ±0.08	0.3 ±0.11	0.053 ±0.05	0.175 ±0.14	0.037 ±0.05	0.055 ±0.03	0.118 ±0.1	0.144 ±0.03	0.356 ±0.05
		+	+	=	+	+	+	+	+	+
	Bin PSO	0.252 ±0.08	0.313 ±0.1	0.113 ±0.13	0.295 ±0.19	0.067 ±0.12	**0.049 ±0.03**	0.105 ±0.1	0.168 ±0.03	0.346 ±0.04
		+	+	+	+	+	+	+	+	+
	Bin DE	0.245 ±0.09	0.292 ±0.1	0.044 ±0.05	0.18 ±0.08	**0.02 ±0.03**	0.051 ±0.04	0.095 ±0.06	0.145 ±0.04	0.374 ±0.04
		+	+	-	+	+	+	+	+	+
100	Pr BABC	**0.209 ±0.06**	**0.233 ±0.08**	0.047 ±0.05	**0.145 ±0.14**	**0.063 ±0.06**	0.07 ±0.04	**0.068 ±0.08**	**0.144 ±0.04**	**0.337 ±0.04**
	GA	0.27±0.1	0.317±0.11	0.066±0.06	0.215±0.12	0.073±0.05	0.072±0.03	0.153±0.06	0.151±0.03	0.371±0.05
		+	+	+	+	+	-	+	+	+
	Bin PSO	0.282 ±0.09	0.317 ±0.09	0.134 ±0.11	0.35 ±0.16	0.12 ±0.24	**0.064 ±0.035**	0.133 ±0.07	0.156 ±0.04	0.369 ±0.04
		+	+	+	+	+	-	+	+	+
	Bin DE	0.28 ±0.1	0.27 ±0.1	**0.044 ±0.04**	0.18 ±0.11	0.07 ±0.04	0.071 ±0.04	0.11 ±0.07	0.156 ±0.04	0.361 ±0.05
		+	+	+	+	+	=	+	+	+

Table 5: Gene Size Comparison of Evolutionary Algorithms

		CLL-SUB	CNS	DLBCL	Glioma	Leuk2	Lung	Prostate	Stjude	WBC
1500	Pr BABC	**98.5 ±2.31**	**103.3 ±19**	**118.15 ±11.6**	113.35 ±12.7	**121 ±13.8**	**110.7 ±10.5**	**114 ±16.16**	**98.85 ±2.14**	**58.35 ±2.19**
	GA	148.8 ±11.98	155.5 ±12.9	151.65 ±16.14	225.7 ±18.18	217.1 ±16.13	226.8 ±13.89	148.25 ±11.46	228.2 ±12.4	162.5 ±10.18
	Bin PSO	261.9 ±181.6	183.2 ±25.93	338.9 ±26.96	**2.9 ±4.86**	325.15 ±221.4	436.6 ±20.84	299.7 ±28.61	486.5 ±136.4	318.4 ±22.61
	Bin DE	241.6 ±13.15	236.3 ±13.49	230.7 ±14.19	229.3 ±14.36	231.9 ±14.64	235.1 ±16.45	227.6 ±9.94	247.8 ±17.86	245.5 ±15.63
1000	Pr BABC	**59.55 ±1.89**	**61.4 ±1.39**	**79.1 ±7.88**	**75.65 ±2.81**	**82.95 ±8.87**	**71.85 ±8.23**	**74.1 ±9.16**	**65.1 ±1.41**	**39.05 ±1.4**
	GA	99.15 ±9.02	154.8 ±12.91	99.35 ±10.01	102.35 ±9.9	152 ±8.84	152.25 ±10.8	103.1 ±9.54	114.7 ±12.32	106.2 ±10.28
	Bin PSO	256.4 ±14.16	88.1 ±15.31	192.25 ±167.37	40 ±12	312.45 ±15.1	302.2 ±11.5	306.95 ±152.8	333.65 ±67.5	268.7 ±11.73
	Bin DE	241.6 ±13.1	152.4 ±11.94	156.55 ±13.55	149.85 ±14.8	152.1 ±10.82	156.2 ±10.78	152 ±12.3	163.65 ±10.19	163.75 ±14.6
750	Pr BABC	**45.45 ±1.1**	**45.1 ±6.07**	**60.35 ±7.4**	56.35 ±1.07	**61.95 ±4.39**	**53.95 ±9.34**	**53.05 ±1.14**	**47.6 ±15.29**	**28.4 ±1.01**
	GA	114.65 ±9.12	114.3 ±10.2	74.45 ±8.69	111.45 ±6.21	77.9 ±8.4	114.35 ±10.17	77.4 ±6.92	84.55 ±9.25	124.9 ±10.3
	Bin PSO	181.05 ±93.1	71.4 ±10.83	134.31 ±88.9	**34.45 ±79.11**	219.85 ±84.14	187.2 ±96.01	79.05 ±10.34	245.4 ±56.77	150.2 ±10.57
	Bin DE	118.65 ±9.17	115.55 ±8.49	113.2 ±11.52	114.55 ±8.49	111.55 ±7,49	115.35 ±10.12	109.7 ±7.78	124.95 ±7.23	0126.6 ±11.01
500	Pr BABC	**33.25 ±6.7**	**31.65 ±9.6**	**38.25 ±8.04**	36.2 ±6.45	**38.3 ±6.05**	**38.95 ±7.6**	**37.25 ±8.87**	**33.45 ±6.06**	**20.85 ±7.73**
	GA	75.45 ±8.39	51.4 ±5.04	48.5 ±5.73	75.15 ±9.67	52.15 ±7.52	50.8 ±6.66	75.85 ±7.84	59.85 ±9.72	57.75 ±6.77
	Bin PSO	137.55 ±56.63	47.5 ±64.06	110.4 ±9.08	**12.15 ±40.13**	136.95 ±58.69	159.35 ±57.1	115.45 ±69.95	187.4 ±32.9	126.1 ±55.93
	Bin DE	80.3 ±10.03	77 ±6.54	77.2 ±9.93	75.13 ±7.05	78 ±7.04	74 ±7.26	77.1 ±7.12	83.6 ±8.62	85.75 ±9.89
300	Pr BABC	**23.35 ±7.6**	**24.45 ±5.7**	**23.55 ±4.01**	**21.25 ±4.35**	**24.4 ±4.31**	**25.45 ±4.37**	**27.15 ±5.51**	**28 ±4.77**	**15.55 ±5.36**
	GA	48.55 ±6.06	30.2 ±4.81	31.15 ±4.02	46.35 ±5.56	29.9 ±4.58	30.2 ±5.6	44.1 ±6.72	55.45 ±7.4	50.2 ±6.06
	Bin PSO	79.6 ±40.5	42.55 ±43.1	59.6 ±51.3	28.2 ±4.69	94.6 ±46.28	82.05 ±35.93	136.1 ±81.1	106.35 ±13.9	96.4 ±18.9
	Bin DE	47.35 ±4.89	46.3 ±8.77	45.05 ±6.76	43.2 ±5.86	47.15 ±6.22	45.2 ±7.79	45.65 ±7.61	50 ±6.5	50.15 ±5.78
100	Pr BABC	**8 ±2.49**	**9.1 ±3.14**	**9.65 ±2.85**	7.8 ±2.9	**8.4 ±2.66**	**8.95 ±2.44**	**8.35 ±3.52**	**15.95 ±2.8**	**10.8 ±2.95**
	GA	15.95 ±5.07	19.05 ±2.83	15.9 ±4.05	14.45 ±3.88	15.25 ±4.22	12.8 ±3.46	14.3 ±2.9	24.4 ±3.4	21.65 ±4.25
	Bin PSO	30.2 ±8.42	20.7 ±14.71	16.4 ±17.6	**6.75 ±11.71**	29.05 ±14.74	34.4 ±10.83	23.65 ±15.67	40 ±6.45	36.9 ±8.86
	Bin DE	17.1 ±3.27	14.75 ±3.29	15.6 ±3.31	17.3 ±4.09	16.15 ±3.69	15.95 ±3.10	15.7 ±3.54	22.75 ±2.51	18.75 ±3.86

use a separate test set and gave accuracy results according to train set accuracy. All methods selected the top n genes for each dataset, and this n number is between 10 and 100. Generally, in such studies, the average accuracy results of these top 10, 20, ..., 100 genes are provided. Therefore, the given results of PrBABC are obtained by the top 50 genes selected with the 3-Filter method. In some studies, there is not an exact gene size, so we could not report the number of genes in these datasets.

BCO [Wang et al. 2017] is a discrete bacterial algorithm. The population size of this algorithm was 50, iteration number was 100, and classifier was 5-NN, as in the current reported work. These authors did not use any filter algorithm; instead, they limited the selected gene number to a predefined

Table 6: Accuracy and Gene Size Comparison with Other Methods

Method	Cll-Sub	DLBCL	Glioma	Pros Can	CNS	Stjude	Lung Can	Leuk2	WBC
PrBABC	**0.973 (6.6)**	1 (5.4)	**1 (4.65)**	0.99 (5.6)	**1 (5.3)**	**0.935 (12.7)**	**0.99 (6.95)**	1 (5.8)	**0.836 (6.65)**
BCO [Wang et al. 2017]	-	**1(3.1)**	-	**1(7)**	-	-	-	**1(3.5)**	-
RLR [Guo et al. 2016]	0.747	0.93	-	-	-	0.854	-	-	-
PLSDR5 [Guo et al. 2017]	0.821	0.928	-	-	-	-	0.941	-	-
[Aziz et al. 2016]	-	-	0.793 (25)	0.886 (50)	-	-	-	-	-
MOEDA [Lv et al. 2016]	-	0.99 (4)	-	0.96 (12)	-	-	0.96 (25)	1 (4)	-
LM [Sun et al. 2016]	-	0.958	-	0.947	-	-	-	-	-
mRMR-CS [Mohamed et al. 2017]	-	-	-	-	0.714 (7)	-	-	-	-
[Mortazavi et al. 2016]	-	0.929	-	0.901	-	-	-	0.921	-
mRMR-ABC [Alshamlan et al. 2015a]	-	-	-	-	-	-	-	1 (20)	-
GBC [Alshamlan et al. 2015b]	-	-	-	-	-	-	-	1(8)	-
MGSACO [Tabakhi et al. 2015]	-	-	-	0.731 (20)	-	-	0.857 (20)	-	-
[Yang et al. 2008]	-	1(4)	-	-	-	-	-	1(11)	-
IG-GA [Yang et al. 2010]	-	1 (107)	-	0.961 (343)	-	-	0.956 (2101)	0.986 (782)	-
MCSO [Mohapatra et al. 2016]	-	-	-	0.996 (50)	-	-	-	0.817 (10)	-

number. This limit value differed according to dataset, and it could not exceed 50. RLR [Guo et al. 2016] is a logistic regression-based feature selection method that uses SVM as the classifier. The authors used the top 1, 2, ..., 50 features selected by different filter algorithms, and the results in Table 6 are the average accuracy results of these subsets. PLSDR5 [Guo et al. 2017] is another logistic regression-based method; in contrast to the previous work, the authors used Partial Least Squares for feature extraction. The results in Table 6 is present the average accuracy results obtained by LDA using the top 10, 20, ..., 100 genes. [Aziz et al. 2016] eliminated irrelevant features using ICA and Fuzzy Backward Feature Elimination. They classified data with SVM and Naive Bayes. We present their best results. MOEDA [Lv et al. 2016] is a multi-objective heuristic algorithm. Datasets are filtered with the mRMR method, and an EDA-based heuristic algorithm is applied to the selected top n genes, where n varies from 1 to 50. The population size was 100, and SVM is used as classifier. In [Sun et al. 2016], the gene selection process was performed with a Lagrange Multiplier. Naive Bayes, K-NN, Random Forest, and Classification and Regression Tree (CART) algorithms were used as the classifier with the top 100 genes and Table 6 reports the best results. In mRMR-CS [Mohamed et al. 2017], the authors applied CS, PSO, and ABC algorithms with the mRMR filter method.

They took the best performance with CS; therefore, we provide the CS results in Table 6. The population size of swarms was 50, and the algorithms ran 1,000 iterations. [Mortazavi et al. 2016] reduced the size of datasets with Fisher Ratio or Mutual Information methods; subsequently features were weighted using Cooperative Game Theory. mRMR-ABC [Alshamlan et al. 2015a] is a combination of the mRMR filter and ABC algorithm. The colony size was set to 80, and the iteration number was set to 100. For each dataset, the top 50, 100, 150, ..., 400 genes were selected with mRMR, and these subsets were classified with SVM. ABC is initialized with the subset that has 100% accuracy. GBC also [Alshamlan et al. 2015b] hybridized mRMR and ABC algorithms with the same parameters. Unlike mRMR-ABC, GBC proposed a new binary ABC version using genetic operators. MGSACO [Tabakhi et al. 2015] is an Ant Colony Optimization-based gene selection method. The colony size was 100, and the maximum number of iterations was 50. SVM, Naive Bayes, and Decision Tree Algorithms were used as classifiers. The top 10, 20, 30, ... , 100 genes were selected with mRMR, and classification accuracy was given for each of them. We list the best classifier results at Table 6 when the feature size is 20. In [Yang et al. 2008], IG and CFS are used as filters. A threshold value is determined, and features with weight higher than this threshold value are selected. The authors proposed an improved Binary PSO version and used it as a wrapper. There are 30 particles in the swarm, and the iteration number was 100. Datasets were classified with K-NN and SVM. The accuracy results in Table 6 were obtained using a K-NN classifier. IG-GA [Yang et al. 2010] includes GA with the IG filter method. There is not an exact feature size that is selected by this filtering method; instead, all features with an IG value of 0 were eliminated. There were 30 individuals in population, and the generation number was 100. Datasets were classified with 1-NN. MCSO [Mohapatra et al. 2016] is an improved version of CSO Algorithm. Ten gene subsets that include 10, 20, ... , 100 genes were obtained using MCSO; they were classified with K-NN; and the best one was selected. We provide the best accuracy results in Table 6.

According to the results in Table 6, BCO produced better results for the DLBCL, Prostate Cancer, and Leukemia 2 datasets. For the DLBCL and Leukemia 2 datasets, BCO classified datasets with the same accuracy but fewer genes than PrBABC. For the Prostate Cancer dataset, the gene set size of PrBABC was smaller than that of BCO, and the difference of accuracy values of these two algorithms is very small. For other datasets, the results of PrBABC is superior in terms of accuracy and gene set size values.

We analyzed gene subsets that were selected by algorithms to identify the most commonly selected genes For each dataset, how many times each gene was selected in 80 gene subsets (4 algorithms x 20 runs) is calculated. The most-selected 10 genes for the datasets are given in Table 7. Numbers in brackets indicate gene ranks.

Table 7: Mostly Selected Genes

Dataset	Affymetrix Number
CLL-SUB	100_g_at(1), 1040_s_at(46), 1076_at(85), 104_at(45), 1080_s_at(90), 1014_at(17), 1045_s_at(51), 103_at(34),1030_s_at(35), 1077_at(86)
Dlbcl	D61380_at(618), D13897_rna2_at(260), D64142_at(659), AB000584_at(16), AFFX-BioB-M_at(121), AFFX-LysX-M_at(162), AFFX-HUMGAPDH/M33197_M_st(149), D10522_at(203), D78261_at(675), AB006782_at(44)
Glioma	1113_at(126), 1130_at(145), 108_g_at(89), 1088_at(98), 1150_at(167), 1176_at(195), 1243_at(265), 1046_at(52), 1090_f_at(101)
Pros. Can.	1074_at(83), 1008_f_at(10), 1088_at(98), 1020_s_at(24), 1045_s_at(51), 108_g_at(89), 1086_at(96), 1133_at(148), 1178_at(197), 1003_s_at(5)
CNS	AB000460_at(10), AC002486_at(67), AF001548_rna1_at(85), AB000467_at(14), AF000430_at(77), AF006084_at(98), AF007111_at(101), AB000410_s_at(7), AB000462_at(11), AB006190_at(42)
Stjude	154_at(584), 1066_at(74), 1262_s_at(285), 1428_at(463), 1512_at(554), 1603_g_at(679), 1046_at(52), 1335_at(361), 1391_s_at(423), 1461_at(499)
Lung Can.	104_at(45), 1175_s_at(194), 1342_g_at(369), 1416_g_at(450), 1072_g_at(81), 1092_at(103), 1365_at(394), 1231_at(253), 1235_at(257), 1290_g_at(315)
Leuk2	1126_s_at(140), 1271_g_at(300), 1011_s_at(14), 1249_at(275), 1310_at(343), 1038_s_at(43), 1068_g_at(76), 1107_s_at(119), 1239_s_at(264), 1337_s_at(372)
WBC	200708_at(236), 200770_s_at(298), 200884_at(412), 200068_s_at(89), 200617_at(145), 200846_s_at(374), 200917_s_at(445), 200096_s_at(117), 200706_s_at(234), 1294_at(6)

Figure 4 shows CPU time comparisons when the algorithms were initialized with the top 100 genes. According to these results, the execution time of PrBABC is close to that of BinDE, and the execution time of GA is close to that of BinPSO in most of the datasets. Generally, PrBABC required more time to select subsets, but the time differences between algorithms are approximately 5-10 seconds. This is a small difference.

5 Conclusion

In this paper, we have proposed an efficient gene selection method for microarray data based on the ABC algorithm. In order to decide which normalization method is useful, we normalized datasets with three well-known microarray normalization methods: MAS5, RMA, and GcRMA, and compared results according to their clustering accuracy, purity, and silhouette coefficient values. We saw that

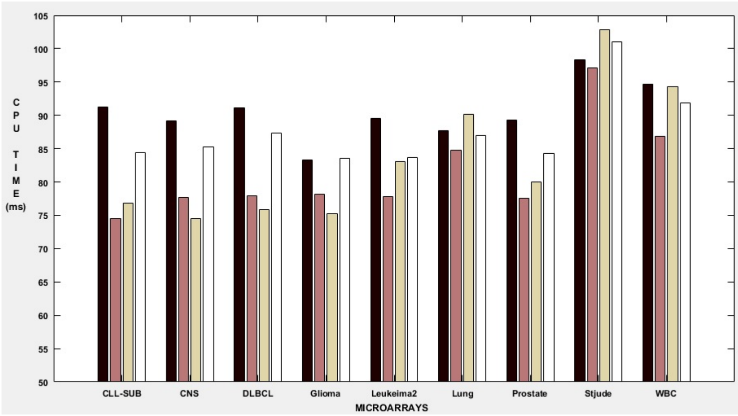


Figure 4: CPU Time Comparison

RMA is the most useful normalization method among them. We applied a hybrid gene selection process to microarrays that include filter and wrapper steps. Genes are weighted using three filter methods: Information Gain, Correlation-Based Feature Selection, and ReliefF. We combined the weighted results of the three filter methods in the pre-processing step. In this way, we benefited from the advantages of 3 different filter methods. For the gene selection process, we proposed a self-adaptive binary ABC algorithm that can efficiently select a learning method according to the current dataset. Results show that applying the 3 methods together yielded more successful results than applying them individually. Additionally, we compared the results of PrBABC with well-known meta-heuristic algorithms: Genetic Algorithm, Differential Evolution, and Binary Particle Swarm Optimization. The proposed probabilistic ABC method outperforms these methods by obtaining superior results with respect to both classification accuracy and gene set size. Finally, we provide the top selected genes for each dataset for using subtype prediction selected by all methods.

References

[Abdi et al. 2012] Abdi, M. J., Hosseini, S. M., Rezghi, M.: "A novel weighted support vector machine based on particle swarm optimization for gene selection and tumor classification"; Computational and Mathematical Methods in Medicine, vol. 2012, (2012), 1-7.

[Alshamlan et al. 2015a] Alshamlan, H., Badr, G., Alohali, Y.: "mRMR-ABC: a hybrid gene selection algorithm for cancer classification using microarray gene expression profiling"; BioMed Research International, vol. 2015, (2015), 1-15.

[Alshamlan et al. 2015b] Alshamlan, H. M., Badr, G. H., Alohali, Y. A.: "Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification"; Computational Biology and Chemistry, 56, (2015), 49-60.

[Apolloni et al. 2016] Apolloni, J., Leguizamón, G., Alba, E.: "Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments"; Applied Soft Computing, 38, (2016), 922-932.

[Aziz et al. 2016] Aziz, R., Verma, C. K., Srivastava, N.: "A fuzzy based feature selection from independent component subspace for machine learning classification of microarray data"; Genomics Data, 8, (2016), 4-15.

[Bolón et al. 2017] Bolón-Canedo, V., Sechidis, K., Sánchez-Marono, N., Alonso-Betanzos, A. and Brown, G.: "Exploring the consequences of distributed feature selection in DNA microarray data"; Proc. IJCNN'17, IEEE, Anchorage (2017), 1665-1672.

[Broad Institute] "Broad Institute Cancer Program Legacy Publication Resources"; [online] http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi, (Accessed 11 September 2018)

[Dalma et al. 2006] Dalma-Weiszhausz, D. D., Warrington, J., Tanimoto, E. Y., Miyada, C. G.: "[1] The Affymetrix GeneChip Platform: An Overview"; Methods in Enzymology, 410, (2006), 3-28.

[Dara et al. 2017] Dara, S., Banka, H., Annavarapu, C. S. R.: "A rough based hybrid binary pso algorithm for flat feature selection and classification in gene expression data"; Annals of Data Science, 4, 3 (2017), 341-360.

[Do and Choi 2006] Do, J. H. and Choi, D. K.: "Normalization of microarray data: single-labeled and dual-labeled arrays"; Molecules and Cells, 22, 3 (2006), 254-261.

[El Akadi et al. 2011] El Akadi, A., Amine, A., El Ouardighi, A., Aboutajdine, D.: "A two-stage gene selection scheme utilizing MRMR filter and GA wrapper"; Knowledge and Information Systems, 26, 3 (2011), 487-500.

[Engelbrecht and Pampara 2007] Engelbrecht A.P. and Pampara G.: "Binary differential evolution Strategies"; Proc. CEC2006, IEEE, Vancouver (2006), pp. 1942-1947.

[Govindarajan et al. 2012] Govindarajan, R., Duraiyan, J., Kaliyappan, K. and Palanisamy, M.: "Microarray and its applications"; Journal of Pharmacy Bioallied Sciences, 4, 2 (2012), 310-312.

[Guo et al. 2016] Guo, S., Guo, D., Chen, L., Jiang, Q.: "A centroid-based gene selection method for microarray data classification"; Journal of Theoretical Biology, 400, (2016), 32-41.

[Guo et al. 2017] Guo, S., Guo, D., Chen, L., Jiang, Q.: "A L1-regularized feature selection method for local dimension reduction on microarray data"; Computational biology and chemistry, 67, (2017), 92-101.

[Guyon and Elisseeff 2003] Guyon, I. and Elisseeff, A.: "An introduction to variable and feature selection"; Journal of Machine Learning Research, 3, Mar (2003), 1157-1182.

[Hall 1999] HALL, M. A.: "Correlation-based feature selection for machine learning"; The University of Waikato, PhD Thesis,Hamilton (1999).

[Hasnat and Molla 2016] Hasnat, A. and Molla, A. U.: "Feature selection in cancer microarray data using multi-objective genetic algorithm combined with correlation coefficient"; Proc. ICETT'2016, IEEE, Kollam (2016), 1-6.

[Hubbell et al. 2002] Hubbell, E., Liu, W. and Mei, R.: "Robust estimators for expression analysis"; Bioinformatics, 18, 12 (2002), 1585-1592.

[Irizarry et al. 2003] Irizarry, R. A, Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B. and Speed, T. P.: "Summaries of Affymetrix GeneChip probe level data"; Nucleic Acids Research, 31, (2003), e15-e15.

[Jia et al. 2014] Jia D., Duan X., Khan M. K.: "Binary artificial bee colony optimization using bitwise operation (BitABC)"; Computers and Industrial Engineering, 76, (2014), 360–365.

[Kalaviani and Kumar 2017] Kalaviani K.and Kumar S.: "An Enhanced Technique for Identifying Cancer Biomarkers from Microarray Data Using Hybrid Feature Selection Technique"; International Journal of Scientific Research in Computer Science, 2, 3 (2017), 192-198.

[Karaboga and Akay, 2009] Karaboga, D. and Akay, B.: "A survey: algorithms simulating bee swarm intelligence"; Artificial Intelligence Review, 31, (2009), 61-85.

[Karaboga and Basturk 2008] Karaboga, D. and Basturk, B.: "On the performance of artificial bee colony (ABC) algorithm"; Applied soft computing, 8, 1 (2008), 687-697.

[Kasha et al. 2012] Kasha M K, Nahavandi N, Kashan A H.: "DisABC:A new artificial bee colony algorithm for binary optimization"; Applied Soft Computing, 12, (2012), 342-352.

[Kennedy and Eberhart 1997] Kennedy J. and Eberhart R.: "A Discrete Binary Version of the Particle Swarm Algorithm"; Proc. World Multiconference on Systemics, Cybernetics and Informatics, IEEE, Orlando (1997), 4104–4109.

[Kiran and Gündüz 2013] Kiran M. S. and Gündüz M.: "XOR based artificial bee colony algorithm for binary optimization"; Turkish Journal of Electrical Engineering and Computation Sciences, 21, (2013), 2307–2328.

[Kiran 2015] Kıran M. S.: "The continues artificial bee colony algorithm for binary optimization"; Applied Soft Computing, 33, C (2015), 15-23.

[Liu et al. 2010] Liu, H., Bebu, I., Li, X.: "Microarray probes and probe sets"; Frontiers in Bioscience (Elite edition), 2, (2010), 325.

[Lv et al. 2016] Lv, J., Peng, Q., Chen, X., Sun, Z.: "A multi-objective heuristic algorithm for gene expression microarray data classification"; Expert Systems With Applications, 59, (2016), 13-19.

[Mandala and Gupta 2014] Mandala M. and Gupta C. P.: "Binary artificial bee colony optimization for GENCO's profit maximi-zation under pool electricity market"; International Journal of Computer Applications, 90, (2014), 34-42.

[Mirjalili and Lewis 2012] Mirjalili S. and Lewis A.: "S-shaped versus v-shaped transfer functions for binary particle swarm optimization"; Swarm Evolution Computation, 9, (2012), 1–14.

[Mohamed et al. 2017] Mohamed, N. S., Zainudin, S., Othman, Z. A.: "Metaheuristic approach for an enhanced mRMR filter method for classification using drug response microarray data"; Expert Systems with Applications, 90, (2017), 224-231.

[Mohapatra et al. 2016] Mohapatra, P., Chakravarty, S., Dash, P. K.: "Microarray medical data classification using kernel ridge regression and modified cat swarm optimization based gene selection system"; Swarm and Evolutionary Computation, 28, (2016), 144-160.

[Mortazavi et al. 2016] Mortazavi, A. and Moattar, M. H.: "Robust feature selection from microarray data based on cooperative game theory and qualitative mutual information"; Advances in bioinformatics, vol.2016, (2016), 1-15.

[NCBI] "National Center of Biotechnology Information"; [online] https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE2466, (Accessed 11 September 2018)

[NCBI] "National Center of Biotechnology Information"; [online] https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE82009, (Accessed 11 September 2018)

[NCBI] "National Center of Biotechnology Information"; [online] $https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc = gse2034$", (Accessed 11 September 2018)

[Ozmen et al. 2018] Ozmen O., Batbat T., Ozen T., Sinanoglu C., and Guven A.: "Optimum Assembly Sequence Planning System Using Discrete Artificial Bee Colony Algorithm"; Mathematical Problems in Engineering, vol 2018, (2018), 1-14.

[Ozturk et al. 2014] Öztürk C, Hançer E, Karaboğa D.: "Dynamic clustering with improved binary artificial bee colony-IDisABC"; Applied Soft Computing, 28, C (2014), 69-80.

[Ozturk et al. 2015] Öztürk C, Hançer E, Karaboğa D.: "A novel binary artificial bee colony algorithm based on genetic operators"; Information Science, 297, (2015), 154-170.

[Qin et al. 2005] Qin, A. K., Suganthan, P. N.: (2005, September). "Self-adaptive differential evolution algorithm for numerical optimization"; Proc. Evolutionary Computation'2005, IEEE, Edinburgh, Scotland (2005),1785-1791.

[Quackenbush 2002] Quackenbush, J.: "Microarray data normalization and transformation"; Nature genetics, 32, (2002), 496.

[Robnik-Sikonja and Kononenko 2003] Robnik-Sikonja, M. and Kononenko I.: "Theoretical and empirical analysis of ReliefF and RReliefF"; Machine Learning, 53, (2003), 23–69.

[San Segundo-Val and Sanz-Lozano 2016] San Segundo-Val I. and Sanz-Lozano C.S.: "Introduction to the Gene Expression Analysis"; Springer, Humana Press/ New York (2016).

[Sivanandam S. and Deepa 2008] Sivanandam S. and Deepa S.: "Introduction to Genetic Algorithms"; MIT Press, Heidelberg/Berlin (2008).

[St.Jude] "St.Jude Children's Research Hospital"; [Online] $http://www.stjuderesearch.org/site/data/ALL1/all_rawdata$ (Accessed 11 September 2018)

[Sun et al. 2016] Sun, S., Peng, Q., Zhang, X.: "Global feature selection from microarray data using Lagrange multipliers"; Knowledge-Based Systems, 110, (2016), 267-274.

[Tabakhi et al. 2015] Tabakhi, S., Najafi, A., Ranjbar, R., Moradi, P.: "Gene selection for microarray data classification using a novel ant colony optimization"; Neurocomputing, 168, (2015), 1024-1036.

[Tran and Wu 2014] Tran D. C., and Wu Z.: "New approaches for binary artificial bee colony algorithm for solving 0-1 knapsack problem"; Advances in information Sciences and Service Sciences, 4, 22 (2014), 464-471.

[Tsamardinos and Aliferis, 2003] Tsamardinos, I. and Aliferis, C. F.: "Towards principled feature selection: relevancy, filters and wrappers"; Proc. AISTATS'2003, Morgan Kaufmann Publishers, (2003).

[Veček et al. 2017] Veček, N., Liu, S. H., Črepinšek, M., Mernik, M.: "On the importance of the artificial bee colony control parameter 'Limit"'; Information Technology And Control, 46, 4 (2017), 566-604.

[Wang et al. 2017] Wang, H., Jing, X., Niu, B.: "A discrete bacterial algorithm for feature selection in classification of microarray gene expression cancer data"; Knowledge-Based Systems, 126, (2017), 8-19.

[Wei and Hanning, 2012] Wei L. and Hanning C.: "BABC: A binary version of artificial bee colony algorithm for discrete optimization"; International Journal of Advancements in Computing Technology, 4, 14 (2012), 307-314.

[Wilcoxon, 1945] Wilcoxon, F.: "Individual Comparisons by Ranking Methods"; Biometrics Bulletin, 1, 6 (1945), 80-83.

[Wu et al. 2004] Wu, Z., Irizarry, R. A., Gentleman, R., Martinez-Murillo, F., Spencer, F.: "A model-based background adjustment for oligonucleotide expression arrays"; Journal of the American statistical Association, 99, 468 (2004), 909-917.

[Yang et al. 2008] Yang, C. S., Chuang, L. Y., Ke, C. H., Yang, C. H.: "A Hybrid Feature Selection Method for Microarray Classification"; IAENG (International Journal of Computer Science), 35, 3 (2008), 285-290.

[Yang et al. 2010] Yang, C. H., Chuang, L. Y., Yang, C. H.: "IG-GA: a hybrid filter/wrapper method for feature selection of microarray data"; Journal of Medical and Biological Engineering, 30, 1 (2010), 23-28.

[Yurtkuran and Emel 2016] Yurtkuran A. and Emel E.: "A discrete artificial bee algorithm for single machine scheduling problem"; International Journal of Production Re-search, 54, 22 (2016), 6860-6878.

[Zhang and Gu, 2015] Zhang S. and Gu X.: "An effective discrete artificial bee colony algorithm for flow shop scheduling problem with intermediate buffers"; Journal of Central South University, 22, (2015), 3471-3484.