Genomics 109 (2017) 91–107

Contents lists available at ScienceDirect

Genomics

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

Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts

M. Dashtban *, Mohammadali Balafar

Department of Computer Engineering, Faculty of Electrical & Computer Engineering, University of Tabriz, Iran

ARTICLE INFO

Article history:
Received 5 August 2016
Received in revised form 9 January 2017
Accepted 24 January 2017
Available online 1 February 2017

Keywords:
Gene selection
Cancer classification
Microarray data analysis
Intelligent Dynamic Algorithm
Random-restart hill climbing
Reinforcement learning
Penalizing strategy
Cut and splice crossover
Self-refinement strategy
Feature selection

ABSTRACT

Gene selection is a demanding task for microarray data analysis. The diverse complexity of different cancers makes this issue still challenging. In this study, a novel evolutionary method based on genetic algorithms and artificial intelligence is proposed to identify predictive genes for cancer classification. A filter method was first applied to reduce the dimensionality of feature space followed by employing an integer-coded genetic algorithm with dynamic-length genotype, intelligent parameter settings, and modified operators. The algorithmic behaviors including convergence trends, mutation and crossover rate changes, and running time were studied, conceptually discussed, and shown to be coherent with literature findings. Two well-known filter methods, Laplacian and Fisher score, were examined considering similarities, the quality of selected genes, and their influences on the evolutionary approach. Several statistical tests concerning choice of classifier, choice of dataset, and choice of filter method were performed, and they revealed some significant differences between the performance of different classifiers and filter methods over datasets. The proposed method was benchmarked upon five popular high-dimensional cancer datasets; for each, top explored genes were reported. Comparing the experimental results with several state-of-the-art methods revealed that the proposed method outperforms previous methods in DLBCL dataset.

1. Introduction

Since the emergence of novel biotechnology, several methods have been proposed for microarray data analysis. Utilizing high-density oligonucleotide chips and cDNA arrays enable researchers to measure the expression levels of thousands of genes simultaneously in a single microarray experiment. The obtained genes could be used in various applications such as medical diagnosis and prognosis. One of the most important applications of microarray data is the classification of tissue samples into the normal or cancerous tissues. One of the most important applications of microarray data is the classification of tissue samples into the normal or cancerous tissues. However, a significant number of genes are irrelevant or insignificant to clinical applications [13,43,77]; consequently, they are unrelated to the classification tasks [31,84]. On the other hand, interpreting such huge number of genes is impossible. Therefore, selecting a proper number of most discriminating genes has been the most challenging task in microarray data analysis.

There are some major challenges associated with the analysis of microarray data, for example, they have a large number of genes (the curse of dimensionality) and a few number of experiments (the curse of data sparsity) usually <100 samples [12]. Moreover, they have high complexity because most of the genes are directly or indirectly correlated with each other, e.g., a gene with high expression level may simply be activated by a high-regulated gene. Several methods have been proposed in the literature to deal with such issues.

Several statistical methods have been developed to select the genes for disease diagnosis, prognosis, and therapeutic targets [30,73]. In addition to the statistical methods, recently, data mining and machine learning solutions have been widely used in genomic data analysis [41,56,88]. For example, Cho et al. [18] used a modified kernel Fisher discriminant analysis (KFDA) to analyze the hereditary breast cancer dataset [34]. The KFDA classifier used the mean-squared-error as the gene selection criterion. Besides, many hybrid evolutionary algorithms have been proposed to improve the accuracy of the classification methods [36,69,72]. Various evolutionary algorithms aim to find an optimal subset of features by using bio-inspired solutions (such as PSO, Honey Bee, Firefly algorithms). These kinds of algorithms have shown appropriate performances over various problems but are dependent on experts" intervention to obtain the desired performance. Gene selection is a subgroup of larger machine learning class of feature selection. Feature selection methods could be roughly classified into four distinct models: filter, wrapper, hybrid, and embedded models [41,65].

The filter model relies on the general statistical properties of training data without using any learning algorithm. In this model, genes are usually ranked individually using few criteria [61,62,70]. The genes with the highest rank can be selected for further analysis. Some of the successful

* Corresponding author.
E-mail addresses: dashtban@tabrizu.ac.ir, Dashtban.Edu@gmail.com (M. Dashtban).

http://dx.doi.org/10.1016/j.ygeno.2017.01.004
0888-7543/© 2017 Elsevier Inc. All rights reserved.

filter approaches include Laplacian score [33], signal-to-noise ratio [30], mutual information [11], information gain [59], consensus independent component analysis that uses gene expression value for cancer classification [90], T-test feature ranking for gene selection [92], fuzzy logic for eliminating of redundant features, [37], maximum–minimum correntropy criterion [54], and receiver operating characteristics analysis [42]. Also, there is an excellent survey of filter techniques in [44] which focused on gene selection for microarray data analysis.

The wrapper model often utilizes evolutionary strategies to guide their searches. It often starts with a population of solutions; each contains a subset of features. Then each subset would be evaluated using a learner to assign fitness to each subset. Usually, an iterative process is used to improve the solutions (i.e., feature subsets). Some of the state-of-the-art wrapper approaches are particle swarm optimization [39], ant colony optimization [87], artificial bee colony (ABC) algorithm [29,32], ADSRPCL-SVM [81], genetic algorithm with SVM [78] and genetic programming (used for the prediction of alternative mRNA splice variants) [76].

Gene selection is a subgroup of larger machine learning class of feature selection. Feature selection methods could be roughly classified into four distinct models: filter, wrapper, hybrid, and embedded models [41,65]. The filter model relies on the general statistical properties of training data without using any learning algorithm. In this model, genes are usually ranked individually using few criteria [61,62,70]. The genes with the highest rank can be selected for further analysis. Some of the successful filter approaches include Laplacian score [33], signal-to-noise ratio [30], mutual information [11], information gain [59], consensus independent component analysis that uses gene expression value for cancer classification [90], T-test feature ranking for gene selection [92], fuzzy logic for eliminating of redundant features, [37], maximum–minimum correntropy criterion [54], and receiver operating characteristics analysis [42]. Also, there is a good survey of filter techniques in [44] which focused on gene selection for microarray data analysis.

The wrapper model often employs evolutionary strategies to guide their searches. It often starts with a population of solutions; each contains subset of features. Then each subset would be evaluated using a learner to assign fitness to each subset. Usually, an iterative process is used to improve the solutions (i.e., feature subsets). Some of the state-of-the-art wrapper approaches are particle swarm optimization [39], ant colony optimization [87], artificial bee colony (ABC) algorithm [29, 32], ADSRPCL-SVM [81], genetic algorithm with SVM [78] and genetic programming (used for the prediction of alternative mRNA splice variants) [76].

The performance of wrapper approaches is typically better than filter models because they employ the interactions between the solutions and predictors. However, the high time complexity of this model, particularly for high-dimensional data makes the need for using hybrid approaches that have lower time complexity, while a hybrid approach uses a filter model to reduce the dimensionality, it is aimed to achieve a trade-off between the time complexity and feature space size. Some state-of-the-art hybrid approaches are information gain with a novel mimetic algorithm [94], chi-square statistics with GA [45], mRMR with GA [2], a novel similarity scheme with ABC [32], and hybrid between genetic algorithm and SVM [49].

There are other feature selection approaches in which the process of learning a classifier is concurrent with feature selection. It does not use a filter model and accordingly does not shrink the feature space, instead, it tries to select high discriminant features and remove poor ones by analyzing and measuring their effects upon constructing a classifier. Furthermore, their time complexities are relatively high, particularly for high-dimensional data such as microarray data. Some of the latest embedded methods are the random forest for genomic data analysis [17], convergent random forest for predicting drug response [8], and artificial neural network approach for improving classification of precursor microRNA [63].

In the present study, a novel hybrid evolutionary algorithm called intelligent dynamic genetic algorithm (IDGA) based on genetic algorithm and some artificial intelligence concepts and techniques is described. The proposed method mainly consists of two major steps. In the first step, a score-based method is used to reduce the dimensionality and more importantly to provide statistically significant genes to the next step. In the second step, quite different scoring methods are used, the Fisher score and the Laplacian score. The performance of Fisher score and its robustness to noise has already been proven in the literature for various applications [51,58,82]. Furthermore, the high performance of Fisher score for gene selection against other widely used methods such as T-test [16], information gain, and *Z*-score was shown by [83, 85]. Nonetheless, each method has its own characteristics that affect the stability of final results [23,91]. Further, Laplacian discriminant analysis [50,57] shows its competitive performance for identifying predictive genes in cancer datasets. The Laplacian score is an unsupervised method that relies on the underlying structure of a dataset. This characteristic motivated us to utilize and investigate it as a preprocessing step despite that is an unsupervised method. The proposed IDGA is benchmarked in combination with both Laplacian and Fisher score. Beforehand, a comparison based on dissimilarity of selected top M genes are performed. To the best of our knowledge, it is the first time that the Laplacian score is used directly as a gene ranking method for reducing dimensionality in a hybrid method for cancer classification.

After reducing dimensionality and selecting statistically significant genes, the IDGA method is applied. The presented evolutionary strategy is, in fact, an integer-coded genetic algorithm with dynamic-length genotype, intelligent adaptive parameters, and modified genetic operators, followed by some random initializations (populations including chromosomes with randomly generated length). To the best of our knowledge, it is the first genetic algorithm-based strategy that uses dynamic length with integer-encoding scheme for feature selection at all. The fast convergence of this method motivates us to exploit it on the high-dimensional microarray data. In fact, the variable length chromosomes with integer-encoding scheme followed by adapted genetic operators capable of dealing with dynamic chromosomes with straightforward and random initialization made this algorithm quite effective.

A few genetic algorithms with adaptive mutation and crossover probability for feature selection have proposed in the literature [45, 68], which have several parameters to adjust. In the present study, an adaptive crossover and mutation rate based on the social concept of ‘encouraging and penalizing strategy,’ is proposed. The IDGA algorithm obtains its parameters simply with regards to the quality of explored solution compared with its pair and with total solutions. One significant advantage of this method is that the probability of promoting poor solutions would become higher by adopting more forces on the poor solutions to mutate and to cross over. Furthermore, this algorithm practically uses the well-known artificial intelligence as a concept, the random restart hill climbing for avoiding biased or improper initializations. It performs through initializing new population with randomly generated length a few times and running evolutionary process on each.

Five high-dimensional microarray cancer datasets are used to demonstrate the performance of the proposed evolutionary algorithm. The general convergence trend of IDGA is also studied, and its convergence over five different datasets regarding the number of selected genes and error loss are exhibited. The experimental results demonstrate the comparable performance of simple IDGA regarding prediction error and number of selected genes.

2. Materials and methods

Microarray dataset is usually represented as N by M matrix, where N is the number of experimental samples and N is the corresponding gene expressions. In the present study, five duplicate microarray datasets were utilized to evaluate the proposed method for gene selection and cancer classification. A brief description of these datasets is demonstrated in Table 1. The first dataset was the small round blue cell tumor (SRBCT) dataset [40]. The SRBCT was a four-class cDNA microarray dataset, which contained 83 samples with 2308 genes. The second set is the breast cancer dataset [34]. This dataset contained 22 breast tumor samples from 21 breast cancer patients, of which all of them were women except one. Fifteen women had hereditary breast cancer, seven BRCA1-associated tumors and eight BRCA2-associated tumors. The other seven samples were not cancerous. For each breast tumor sample in this dataset, 3226 genes were existed.

Another dataset was the distinct types of diffuse large B-cell lymphoma (DLBCL) obtained from gene expression data from Stanford University [4]. It contained 4027 attributes and 47 instances. The fourth microarray was the acute lymphoblastic leukemia (ALL)/acute myeloid leukemia (AML) dataset [30]. It was a two-class oligonucleotide microarray dataset for leukaemogenesis obtained from the Broad Institute website. It contained 72 samples (47 ALL samples and 25 AML samples) with 5147 genes. The last one was prostate cancer dataset by Singh et al. [67]. It included 136 samples with 12,600 genes. It was the most complicated dataset for cancer classification among others. However, we found that what affects final results is often the ratio of the train to test. The lower the number of train samples compared with test samples, the more complicated situation imposed. Several analyses were performed that will be seen in experimental results. Beforehand, the proposed method is described in detail in the following section.

2.1. Proposed method

The proposed evolutionary method consists of two main steps. First, a filtering method was employed to reduce the number of genes and to select the top statistically relevant genes. In this step, two well-known filter methods were described, compared, and exploited in the whole process. Later, a novel evolutionary algorithm, which is based on the concept of genetic algorithms, reinforcement learning, and random restart hill climbing, is developed to explore the subset of highly relevant genes. The whole process is demonstrated in Flowchart 1.

According to Flowchart 1, just after the original dataset loaded and training and testing samples were separated, top M genes, which were statistically assessed and ranked using a filter method, were used to create a new dataset with the smaller feature set. The new dataset was used to create a population of chromosomes containing subsets of genes. Subsequently, a random number such as r was generated and new chromosomes have been set up with a random length at most equals to r. The parameter r was at most equal to M, i.e., the dimensionality of the new dataset. Furthermore, a minimum length of r_0 could also be defined. Next, the population was evaluated using a fitness function that employs a classifier. After assigning fitness, the population was sorted to find the elite individuals to be stored. Genetic operators were then applied over the selected population individuals and accordingly the variable length offspring were created. The pressure on individuals to evolve was adjusted adaptively and intelligently based on some effective concepts.

New offspring, after each iteration, was transferred to the next generation. To ensure elitism, superior individuals were transferred directly to other generations. After several iterations, elite individuals were selected to be stored in the repository for further analysis. Subsequently, the evolutionary process restarts by creating the new population of uniform-length individuals. Immediately after a few random restarts, the obtained solutions in repository were evaluated using some classifiers. This procedure was applied on five different datasets as aforementioned in the previous section. It is worth noting that the proposed method is, in fact, an extended genetic algorithm, which despite its simplicity has a great potential that effectively produces inclusive results with several state-of-the-art methods.

Overall, the proposed methodology imposes more focus on searching for superior solutions by exploring new regions of search space. It performs that by ensuring diversity via intelligent force upon population individuals to evolve, and random restart initializations to avoid bad initializations and large local minimum surface while exploring wider regions of search space. The detailed description of proposed method, including the initial gene selection and IDGA, is explained in the following sections.

Table 1
Microarray datasets.

Datasets	Genes	Class	Data		
			# samples	# train	# test
SRBCT	2308	4	83	63	20
Breast	3226	2	22	11	11
DLBCL	4026	2	47	30	17
Leukemia	7129	2	71	38	34
Prostate	12,600	2	136	102	34

2.1.1. Initial gene selection

Feature selection is a preprocessing step in many data mining applications to lower the time complexity via reducing search space while increasing the chance of finding high-quality solutions. Most hybrid approaches exploit a filter method in this step to score and select the genes that reveal more information statistically. Applying such filtering techniques helps in removing a large number of irrelevant and redundant genes, leading in alleviating the high dimensionality problem, which is common in microarray datasets. Reducing the number of irrelevant genes improves the efficiency of microarray data analysis [74]. A variety of scoring methods used for gene selection has been reported in the literature. For instance, signal-to-noise [30,53], the traditional *t*-test [24], Cho's method [19], Mann–Whitney rank sum statistic [60], and between-groups to within-groups sum of square ratio [25].

In the present study, two different scoring methods were selected based on their reported performance in the literature: The Fisher score and the Laplacian score. The first method is supervised, i.e., it considers the class labels to measure the discriminant potential of each gene, and the second is an unsupervised approach. The performance of Fisher score and its robustness to noise has already been proven in the literature [51,82]. Furthermore, the high performance of Fisher score for gene selection against other widely used method such as T-test [16], information gain, and *Z*-score has been already reported in the literature [83,85]. Nevertheless, each method has its own characteristics that affect the stability of final results [23]. Furthermore, Laplacian discriminant analysis [50,57] showed its competitive performance for identifying predictive genes in cancer datasets. The Laplacian score is an unsupervised method that relies on the underlying structure of the dataset, which motivated us to exploit it as a preprocessing step despite that was an unsupervised method. Moreover, a comparison based on the dissimilarity of selected top M genes using these two scoring methods was performed. Both scoring methods were separately used in the present study as the initial feature selection method. To our knowledge, it is the first time that the Laplacian score is used directly as a gene ranking for reducing dimensionality in a hybrid method for cancer classification. To see to what degree the Fisher score is different from Laplacian score, the dissimilarity and the similarity of their selected indices were calculated as demonstrated in Fig. 1. The number of different genes among top 100, 300, and 500 genes are depicted in Fig. 1(a). At first glance, one remarkable difference can be seen in the case of Prostate dataset. In fact, there was almost no similarity between the selected genes of two methods in Prostate, whereas, the lowest dissimilarity was for SRBCT and leukemia with 264 and 262 different genes, respectively, among top 500 ranked genes. The dissimilarity among top 100 genes over all datasets was remarkable. There was near 100% difference among top 100 ranked genes for datasets, except for the SRBCT and leukemia, which were slightly smaller. By juxtaposition of this bar chart with Table 1, it can be observed that the dimensionality of a dataset was

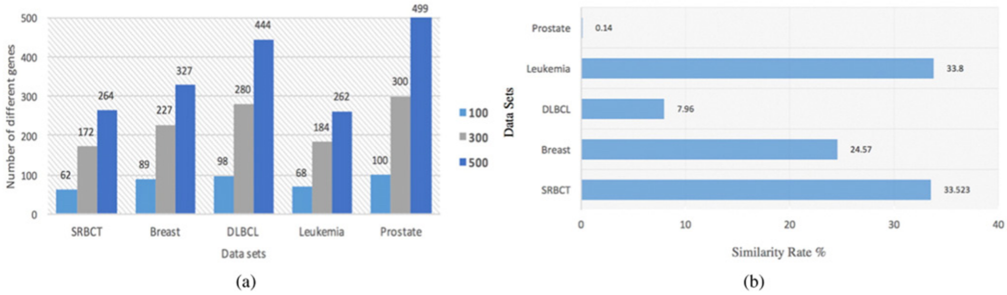


Fig. 1. Dissimilarity and similarity of top-ranked genes by Fisher-criterion and Laplacian-score: (a) the number of different genes among top 100, 300 and 500 genes, respectively, and (b) the similarity rate for each dataset considering top 500 genes, the number in front of a dataset is the ratio of the number of similar genes in that dataset to the total number of similar genes.

not associated with the resulting difference. For example, the number of genes in leukemia is almost twice than that in DLBCL, although the number of different top genes in that was dramatically lower than that in DLBCL with 262 against 444.

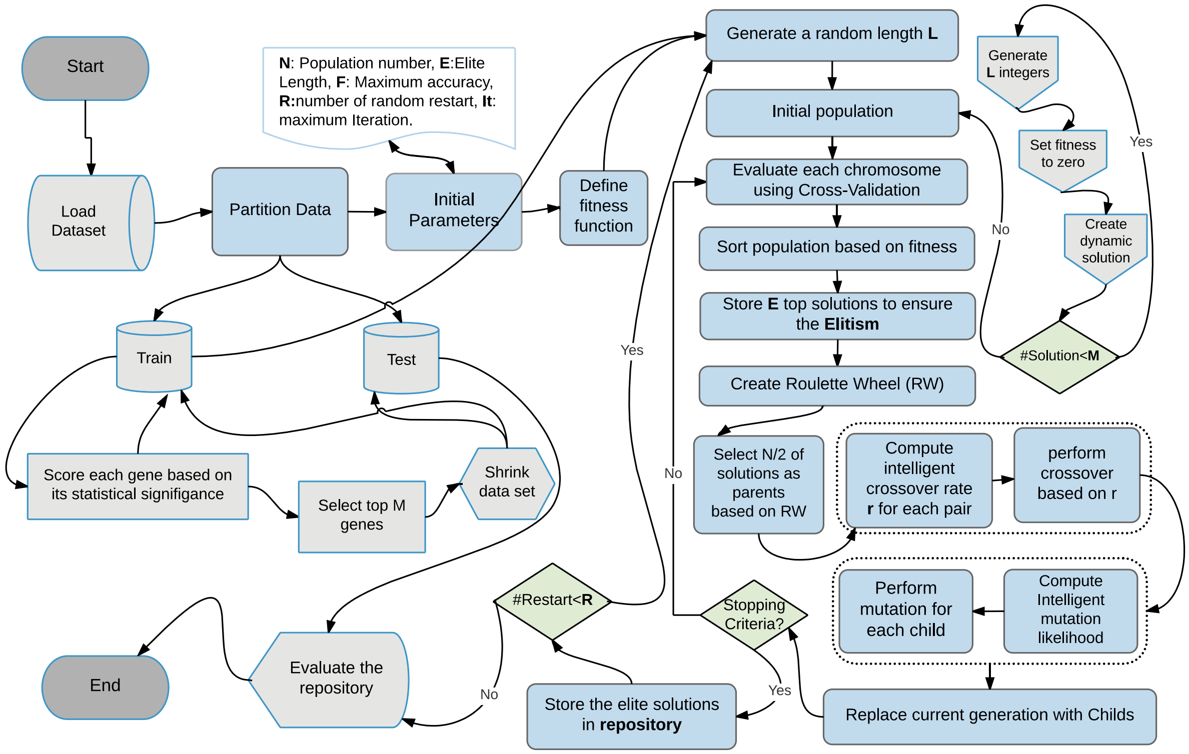
The horizontal bar chart in Fig. 1(b) demonstrates similarity ratio for each dataset, i.e., the number of similar genes among 500 top ones in a dataset to the total number of similar genes took from all datasets. In fact, the bar chart demonstrates the similarity of the filter methods in various datasets. According to that, it is apparent that the similarity of these methods in leukemia and SRBCT was of the highest level compared with others (33 and 34, respectively), while the lowest similarity was conspicuously in the Prostate with only 0.1%, followed by DLBCL with 8% similarity.

2.1.2. Intelligent dynamic genetic algorithm

Finding a minimum number of genes for classification of arrays (samples) into their corresponding categories is an NP-hard problem. Hence, most often, heuristic approaches have been used to find an optimal subset of features (genes). In this section, an evolutionary approach based on *evolutionary concepts*, *artificial intelligence* (*AI*) *concepts*, and techniques such as *random restart hill climbing* and '*encouraging and penalizing strategy*,' for individuals in society, is proposed. Overall, the key features of this algorithm are as follows:

- Dynamic-length chromosomes,
- Intelligent mutation and crossover probability employing AI concepts
- Two-step initialization with randomly generated length,
- New genetic operators able to deal with variable length,
- Two-step analysis.

The whole process from loading data to final evaluation is demonstrated in Flowchart 1. At first, a dataset was loaded and some preprocessing techniques were applied to remove and replace miss-value features. The training group was then divided into two sub-data groups, including training and testing samples sub- data. The training sub-data were used only for constructing a classifier and evaluating individuals during the evolutionary process, while the test sub-data were used to assess the final solutions which were in the repository. In the present study, the original train and test data were employed as described in Table 1. Immediately after partitioning, the dataset was shrunk concerning top selected genes by statistical scoring as described in 'initial feature selection' section. A predetermined number of top genes, 500 genes in the current study as suggested by Lee & Leu [45], were


Start
N: Population number, E:Elite Length, F: Maximum accuracy, R:number of random restart, It: maximum Iteration.
Generate a random length L
Generate L integers
Initial population
Set fitness to zero
Yes
Load Dataset
Partition Data
Initial Parameters
Define fitness function
Evaluate each chromosome using Cross-Validation
No
Create dynamic solution
Sort population based on fitness
Store E top solutions to ensure the Elitism
#Solution<M
Train
Test
Yes
Create Roulette Wheel (RW)
Score each gene based on its statistical signifigance
Select top M genes
Shrink data set
No
Select N/2 of solutions as parents based on RW
Compute intelligent crossover rate r for each pair
perform crossover based on r
End
#Restart<R
Stopping Criteria?
Perform mutation for each child
Compute Intelligent mutation likelihood
Evaluate the repository
No
Store the elite solutions in repository
Yes
Replace current generation with Childs

Flowchart 1. Intelligent dynamic genetic algorithm (IDGA) model for feature subset selection in microarray data.

selected to disjoin full-feature dataset. All two subgroups, including train and test groups, should be contracted. Subsequently, the evolutionary process starts by creating a population of dynamic individuals, the individuals that are variable in length and can change their length within the evolutionary process. At first, the whole population had the same length as a randomly generated length. However, as evolutionary algorithm proceeded, the crossover operator created variability among the population. Population evolve by inducing the effects of three main operations including:

- Spreading good genes within population, practically using crossing and sharing information,
- Adaptive local search around individual using mutation operators,
- Ensuring elitism by maintaining best individuals obtained yet.

As the population individuals evolve, the genetic-algorithm operators compelled the best individuals to get higher talents by incorporating new proper genes and assisted them to get rid of some improper genes. The IDGA method uses a decimal encoding scheme to represent population's chromosomes. Each allele within a chromosome contains an index of a gene; accordingly, a chromosome is represented with a list of integers. In this algorithm, each individual, at first, has the same length. The idea of creating a population with uniform length originates from a social environment model [79] that offers to provide the same environment for every individual to evolve. In this case, in the beginning, every individual had the same number of characteristics. However, superior or inferior individuals were created as the algorithm progresses and the parents shared its features. The selection strategy was based on the merit of an individual. Therefore, beforehand, each individual must be evaluated. The evaluation function in this method concerned two facts: (a) The discriminant potential of an individual that makes it a better predictor or not and (b) number of characteristics (the genes). The general evaluation function is demonstrated in Eq. (1).

$$f_i = \alpha \times prediction\_accuracy_i + \beta \times \frac{1}{features} \tag{1}$$

As is evident, the goodness of an individual is directly proportional to the classification ability, while there is an invert relation regarding the number of features. The parameters α and β were set experimentally to 1 and 0.001. It seemed that the number of features in an individual does not influence the fitness values properly. That was because, in the best case, where there was only one feature, the magnitude of its effect was 0.001 compared with the classification accuracy of most individuals that were higher than 0.70. This subtle value influences practically the evolutionary selection process, particularly when the algorithm converged in respect of classification accuracy. In fact, this subtle difference guides the searching process to gradually converge concerning the number of features. This effect is demonstrated in Fig. 5 in the section of algorithmic behavior. Furthermore, it dramatically affects advancing the evolutionary process by its significant contribution in elitism part. It helps the evolutionary search to move forward by practically bypassing the local minimum surface that is often created, where many individuals have the same superior prediction ability. It also prevents the evolutionary process to get caught on local minimum surfaces on which several poor-accuracy-individuals with a tiny number of features are. Moreover, it was observed the higher value of β could practically mislead the whole search process by producing the state in which several individuals have very poor accuracy, but contain a tiny number of features. Therefore, it is important to employ its effect in the right place, i.e., the elitism section.

After evaluating every chromosome's fitness based on Eq. (1), the cumulative fitness was calculated to build the conventional roulette wheel. The regions on roulette wheel corresponded to the selection share of each individual. Thus, the probability of selection of a chromosome, after each spinning, associated with its fitness. In fact, the higher the fitness of an individual, the higher was its chance of selection. The selection process was for choosing parents to generate new offspring.

After selecting all parents, the population was sorted based on its fitness to obtain the superior individuals. The superior solutions are then stored to ensure elitism. The stored individuals are finally utilized to transfer to the next generation without comparing them with the new coming offspring in the current generation. The number of transferred chromosomes is important in the sense that it could cause the *premature convergence phenomena*. Transferring many chromosomes directly to the next generation causes the population to be flown only with the characteristics of these individuals. This procedure significantly reduces the chance of generating offspring from other regions of search space, perhaps exists global optimum points. As such, only 10% of superior chromosomes are stored in the present study.

Next, just after selecting parents and ensure elitism, it was time to produce offspring. In this stage, the probability of crossover between each parent pair is computed based on a straightforward and intuitive relation proposed in Eq. (2).

$$C_p = \begin{cases} \frac{1-f_m/f}{1-f_M/f}; f_m > f_M \\ 1, otherwise. \end{cases} \tag{2}$$

where f is the maximum fitness that a solution could obtain. If there is only the classification accuracy, then f would be '1'. However, regarding the fitness function presented in Eq. (1), f is slightly greater than maximum accuracy (i.e., '1'), f_m is the average fitness of two parents and f_M is the average fitness of whole population.

Consider f_M, the mean point of overall population fitness and the case that $f_m > f_M$. It is clear that the more the distance from f_M, the more appropriate the solution is; as a consequence, we expect a lower crossover rate. This simple relation is inspired by *behaviorist psychology* [7,55] connected especially to the punishment and reinforcement strategy. The poor individuals are forced to have more interaction to elevate them. Whereas, in contrast, the superior cases are intended for being sought gently for better situations. This simple relation, actually, imposes an *encouraging and penalizing behavior* in population individuals. It is the strategy that has been widely used in educational and learning systems. Moreover, from the machine learning perspective, assigning lower crossover rate for superior individuals provides two possibilities for evolutionary search. First, it results in smoothing movements toward global optimum(s). In fact, subtle changes around superior solutions give rise to the higher possibility of finding global optimum(s). It is intuitive that big changes could simply propel searching process toward fluctuations around optimum apices. This phenomenon could impede the proper convergence of any evolutionary process. Therefore, the relation proposed in Eq. (2) is conceptually sound and acceptable in the sense that it produces lower values for superior individuals and higher values for inferior ones. However, it might not be optimum. For instance, consider the case of choosing median instead of the average point. The same has been expressed for further research in future work section. Second, lower crossover rate for superior individuals reinforces the possibility of 'backward search'. Suppose the state in which the searching process was caught in a wide local minimum surface resulted from new offspring. In this situation, the existence of some superior individuals from previous generations elevates the probability of producing new superior offspring belonged to other regions of search space. Actually, smooth changes in population could provide backward search possibility. Furthermore, there are some shreds of evidence and characteristics about the proposed simple and intuitive relations for computing probability of crossover and mutation as listed below:

- Constant time complexity. The 'mean' operator takes a running time of constant complexity of O('1'), whereas, any single iterative operator such as 'min,' 'max,' and 'median,' the magnitude of time complexity is $\theta(n)$, where n is the number of individuals in the population. It is because these operators need 'sorting' or 'searching' to obtain their target, whereas the 'mean' does not require any searching or sorting.

- The term 'optimum' does not seem to make sense in this case. There has not been any optimum value reported in the literature yet, but instead, there have always been recommendations that are based on shreds of evidence. In fact, most of the parameter tunings are based on given problems, in other words, the problem nature. Besides, most recommendations have been based on pieces of evidence on some particular issues; however, no proven theorem yet. Some values are recommended empirically in the literature. Consequently, an estimated value could be quite effective.
- Superb convergence was observed without any further tuning! Benchmarking the proposed method upon five different datasets over several runs depicts proper convergence trends that make us conclude that the probability adjustment of the proposed relations is adequate enough, at least in the case of this study.

Moreover, the Eq. (2) could be drawn easily from $f_m>f_M$ by simple operations: $f_m>f_M\rightarrow -f_m\leftarrow f_M\rightarrow 1-f_m<1-f_M\rightarrow (1-f_m/1-f_M)<1$.

After computing crossover probability for each parent, a random number was generated, and the crossover operation was carried out based on crossover rate. The conventional crossover for fix-length chromosomes is demonstrated in Fig. 2(b). As seen, it is a two-point crossover that produces two new offspring with the same length. In that method, only two random numbers are generated, and each chromosome inherits the middle points from the other. There is another type of crossover that produces chromosomes with different lengths namely, 'cut and splice crossover' (CSC). In CSC, two random numbers are generated one for each chromosome. Later, the first part of each chromosome combines with second part of another one. In this study, the CSC is extended to a doubled version so that four points are randomly generated, two for each chromosome. A sample of four randomly generated points is demonstrated in Fig. 2(a). Each point is highlighted by a circle where the numbers 2 and 6 are selected in chromosome 1 against two adjacent numbers in the second chromosome. According to offspring, the crossing had two distinct effects including inheritance and dimensionally reduction. These effects are very important in advancing an evolutionary search toward proper solutions concerning the number of genes in a smooth manner. It is also worth noting that the original version of CSC could be used, but it preserves the very small number of links between parents and offspring. The link between parent and offspring generations must sufficiently be preserved to ensure that positive advances, in successful moves toward the optimal organism, are preserved [27]. When there are more links between parents and offspring, results that are more promising could be gained, which may fit the quality of inheritance, even though there is no proven theorem for computerized simulation of evolutionary search about this fact. In the case of fix-length chromosomes, the uniform crossover was the most interesting and promising type of crossing [66]. Three comparatively positive virtues of uniform crossover are listed by Fogel & Atmar [27]:

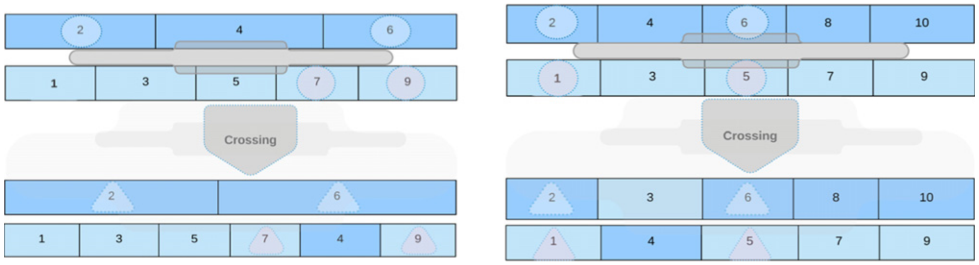
- The disruption of hyperplane sampling under uniform crossover does not depend on the defining length of hyperplanes. That reduces the possibility of representation effects, as there is no defining length bias.
- The disruption potential is easily controlled via a single parameter (Disruption Theory). This suggests the need for only one crossover type (uniform crossover), which is adapted to different situations by adjusting this parameter.
- When a disruption does occur, uniform crossover results in a minimally biased exploration of the space being searched.

The first claim is satisfied in variable-length structure to some extent because there is no constant or predetermined length in the population. Every individual is subject to vary in length. The disruption could be controlled more than the case of one-point cut-and-splice version. However, the most promising type of mutation operator due to Fogel's claims is uniform one. However, for the variable-length structure, there could be only some near-uniform patterns which high time-complexity is required to identify. Hence, only a doubled version of CSC was simulated and utilized. Later, when crossover operations were accomplished, new offspring move forwards for mutation. In this step, first, the mutation probability of each offspring was approximated adaptively using Eq. (3).

$$M_p=\begin{cases} C\times\dfrac{1-f_g/f}{1-f_M/f}, & f_g>f_M \\ 1-f_g/f, & f_g<f_M \end{cases} \tag{3}$$

where f_g is chromosome's fitness, f is the maximum fitness that could be gained by an individual, f_M is average fitness of all population, and Ç is a *contracting* factor for shrinking $M_p{}'$ values to lower ranges.

The same scenario repeats for the proposed dynamic mutation rate as the aforementioned crossover operator. It is expected that the chromosomes with lower fitness values must have higher mutation likelihood which increases the number of genotypes being changed. As such, an encouraging and penalizing mechanism was employed to force wanting chromosomes to grow up by exploring other regions of search space. The elite individuals must be refined much smoothly (than others) to avoid fluctuations around the global optimum. It is evident that high mutation rate brings about wild fluctuations within search space. Such fluctuations do not allow an evolutionary process to converge. On the other hand, very small mutation probabilities lead to late convergence. In the presented equation, the value of mutation likelihood is adjusted adaptively based on an intuitive relation between its fitness and its expected value considering machine learning concepts. In this regard, two distances were employed in this relation



(a) Two-point cut and splice crossover

(b) Fixed two-point crossover

Fig. 2. Comparison of two-point crossover upon fixed and variable-length chromosomes with decimal encoding scheme: (a) two-point crossover based on cut and splice method and (b) a commonplace approach for two-point crossover upon chromosomes with fix length.

including (a) the distance to the maximum achievable fitness, normalized to be out of 1.0, and (b) the distance to the average population fitness. It is obvious that the more the distance from maximum achievable fitness, the lower quality is expected.

A constructive consideration upon the mutation of individuals could even generate superior individuals via mutating the inferior ones. Hence, a relation that regards an own individual's fitness could be a logical criterion. Thus, in the present study, normalized distance ‘$1 - f_g/f$’ was employed for those individuals that were below the overall quality of the population. However, a gentle relation is offered for those individuals that are higher qualities. Considering the overall mean, it is explicit that the higher the distance from this point, the higher the quality of an individual, and accordingly, the lower number of genes is expected to be altered. Consequently, the mutation likelihood is inversely proportional to this distance. On the contrary, the mutation likelihood is directly proportional to the distance of the fitness of the current individual to the maximum fitness value. That is apparent that the more the distance of an individual to the maximum fitness, the lower quality, and consequently, the mutation likelihood must be higher. Multiplying these two intuitive facts, that are of a constant time complexity order, results in producing values that are frequently high. Thus, it should be adjusted accordingly. That is why the contracting factor was introduced. Fortunately, fine-tuning the contracting factor is not a difficult task. It is enough to consider the worst case in which the fitness is almost equal to the overall mean. In this case, the value of equation would be equal to 1.0. Considering the expected value of the overall mean and regarding the other equation for poor individuals, the contracting factor must be at most 0.050. It is worth noting that selecting a very low value, such as 0.10, causes a hindering effect in the algorithm's convergence. That is because, as a population evolves, the number of genes declines and the average of population fitness plunges. Therefore, respecting such tiny number of features, even for the high mutation rate such as 0.020, it is possible that no mutation occurs for too many times. Consider, for instance, the status that there are only five genes. Then, the mutation likelihood of 0.020 results in the mutation of only one gene. To this regard, a gentle relation is presented for those individuals that are higher qualities. Considering the overall mean, it is explicit that the higher the distance from this point, the higher the quality of an individual, and accordingly, the lower number of genes is expected to be altered. Consequently, the mutation likelihood is inversely proportional to this distance. On the contrary, the mutation likelihood is directly proportional to the distance of the fitness of the current individual to the maximum fitness value. That is apparent that the more the distance of an individual to the maximum fitness, the lower quality, and consequently, the mutation likelihood must be higher. Multiplying these two intuitive facts, that are of a constant time complexity order, results in producing values that are frequently high. Thus, it should be adjusted accordingly. That is why the contracting factor was introduced. Fortunately, fine-tuning the contracting factor is not a difficult task. It is enough to consider the worst case in which the fitness is almost equal to the overall mean. In this case, the value of equation would be equal to 1.0. Considering the expected value of the overall mean and regarding the other equation for poor individuals, the contracting factor must be at most 0.050. It is worth noting that selecting a very low value, such as 0.10, causes a hindering effect in the algorithm's convergence. That is because, as a population evolves, the number of genes declines and the average of population fitness plunges. Therefore, respecting such tiny number of features, even for the high mutation rate such as 0.020, it is possible that no mutation occurs for too many times. Consider, for instance, the status that there are only five genes. Then, the mutation likelihood of 0.020 results in the mutation of only one gene. Considering all the evidence, the contracting factor Ç was set to 0.50 in our implementation.

After the mutation, the new offspring are transferred to a new generation. The new generation must be sorted again, and the above steps are repeated until a stop criterion met as demonstrated in Flowchart 1. Generally, the stopping criterion is arbitrary and should be defined based on a given problem. In the present study, three situations were considered as stopping criteria including (a) the agglomerative performance reaches to a constant, (b) the number of genes in the best individual stay unchanged after a predefined number of iterations, and (c) reaching to the maximum iteration. The first two criteria are paramount to accelerate the searching process. In fact, the variable-length chromosomes and the corresponding operators in the proposed method highly enhance the interactions among individuals so that it is exceptional observing the number of genes in the best individual stayed unchanged for a long time; it either increases or decreases. Otherwise, the searching process is caught in a local surface, or there is no better solution, or there is such solution but the feasibility of touching that is extremely low and requires huge effort to reach that. The strategy of random restart hill climbing was used to deal with all these problems. It breaks the process and restarts searching from another location within the search space. Notwithstanding, more iterations are necessary for searching process to converge at some datasets. Therefore, stopping criteria must be experimentally set for various datasets. Moreover, applying the stopping criteria, the average number of iterations for every dataset, had never been >40 over tenths of runs. Plus, in most cases, the algorithm ceased mainly because of reaching the predefined performance that was near the perfect accuracy.

Afterwards, the few elite individuals found over iterations are stored in a repository for further analysis. The algorithm was then restarted from the initialization of a new population. This procedure partially avoids the evolutionary process getting caught in some local minimum surfaces that arise from improper stochastic initializations. Such procedure follows the fundamental principles of random restart hill climbing algorithm [64,80]. Besides, initializing the searching process from new stochastic regions boosts clearly the probability of finding the global optimum. The number of random initializations was set only to 10 in our implementation.

Furthermore, once the evolutionary process was completed, the collected chromosomes in the repository are evaluated using some classifiers to identify the best subset of genes. One consideration is the quality of collected chromosomes should be measured over the testing samples or using leave-one-out- cross-validation (LOOCV) as exhibited in the Flowchart 1.

3. Experimental results

After initial feature selection using either Fisher-score or Laplacian-score, top 500 genes was selected. Then, a new dataset was obtained using the filtered genes. Five high-dimensional microarray datasets were used for evaluation of the proposed method (see Table 1). First, the original split of test/train data was loaded. Then, the proposed method was performed independently on each dataset using either of the filter methods separately. Three widely used classifiers, namely, Support Vector machines (SVM), Naïve Bayes (NBY) and K-Nearest Neighbor (KNN) were used to measure the performance of IDGA. Moreover, since the proposed method is not a filter method, it is possible to obtain a different subset of genes over independent runs with different quality and performance. Therefore, the IDGA were performed independently seven times for each dataset. The obtained results after each run were evaluated using leave-one-out-cross-validation (LOOCV) that is the most promising evaluation criteria (Baeza-Yates & Ribeiro-Neto 1999) and is not subject to change. The average LOOCV results are reported in Table 2 and compared with several well-known and state-of-the-art methods [1,3,20,46,48,71,86]. The comparing results are either LOOCV or 10-fold cross-validation after 2006. Further, the best results of authors were reported wherever available. However, the average results of the proposed method were solely used in the statistical analysis of this paper.

Table 2
Statistical Performance comparison of the proposed method, over 5 public cancer dataset with computational intelligence and statistical approaches. The approaches are chronologically ordered from 2001 to 2016. The number in the parentheses denotes the number of genes.

Methods	***datasets***				
	SRBCT	**Breast**	**DLBCL**	**Leukemia**	**Prostate**
[40]	**100**(96)	-	-	-	-
[71]	**100**(43)	-	-	-	-
[47]	**100**(20)	**-**	-	97.06 (40)	
[46]	-	**100**(10)	84.6(50)	-	-
[18]	-	77.3 (3)	-	73.53 (1)	-
[86]	KNN SVM			KNN SVM	KNN SVM
-GS1	98.8(57) 98.8(34)	-		98.6(60) 98.6(4)	94.1(8) 95.1(47)
-GS2	98.8(78) **100**(96)			98.6(10) 98.6(25)	95.1(8) 95.1(49)
-Cho's	92.8(82) 98.8(80)			97.2(9) 98.6(80)	94.1(8) 94.1(20)
[48]	-	-	93(20)	97.1(20)	91.7(20)
[20]	-	-	93(25)	95.9(25)	-
[1]	-	-	**-**	**100**(18)	-
[26]	-	-	MLP SVM 92.0 93.9	MLP SVM 93.9 99.7	MLP SVM 94.2 96.6
[9]	-	-	D-CFS D-IG 73 93	D-CFS D-IG 94.1 91.1	D-CFS D-IG 58 97.1
[3]	-	-	lasso aplr cbplr 91.5 94.4 95.3	-	lasso aplr cbplr 91.5 92.9 94.8
[39]	SVM KNN NBY 96.4 97.6(6) 97.6(6)	-	-	SVM KNN NBY 83.3 83.3 83.3	-
[83, 85]		bvsscp blasso bpr svmrfe L1svm **100**(10) 91(21)91(19) 95.4(31) 95.4(23)			
IDGA-F					
-SVM	98.6±0.8(21±2.4)	93.0±4.2(7.6±1.8)	98.8±0.8(9.7±3.0)	98.1±0.9(11.7±5.0)	96.8±0.7(24.9±7.7)
-KNN	97.8±0.8(19.6±4.1)	91.7±3.1(8.9±3.1)	95.4±1.7(7.7±1.6)	98.1±0.6(13.7±6.1)	92.5±0.6(28.7±14.2)
-NBY	97.9±1.0(29±5.5)	93.4±2.2(9.1±3.0)	97.9±0.9(8.7±2.8)	95.2±4.5(18.4±4.4)	92.3±1.1 (29.6±8.4)
IDGA-L					
-SVM	97.3±1.2(31.3±6.2)	90.5±4.4(31.3±6.2)	99.7±0.6(20.9±3.5)	95.6±2.2(13.2±2.9)	89.3±1.7(32.3±7.0)
-KNN	96.9±1.0(28.7±7.7)	88.4±3.0(13.0±6.0)	91.2±1.7(22.6±6.6)	97.4±1.0(8.3±1.5)	73.6±0.7(33.4±7.9)
-NBY	96.1±1.4(28.9±7.6)	92.3±1.3(10.3±4.3)	93.0±3.1(19.7±6.3)	97.7±1.1(8.2±2.1)	56.7±4.6(25.3±6.0)
Best	SVM KNN NBY	SVM KNN NBY	SVM KNN NBY	SVM KNN NBY	SVM KNN NBY
Obtained Genes	**100**(18) 91.6 89.2	**100**(2) 95.5 **100**	**100**(9) 97.9 95.8	**100**(15) 97.2 93.1	96.3(14) 95.6 93.4

* Numbers in parenthesis show the number of features, the sign ± indicates a confidence interval of 95%. NBY indicates Naïve Bayes Classifier, SVM is Support Vector Machine and KNN comes from K-nearest Neighbors.

3.1. Evaluating obtained results

Considering Table 2, the IDGA with Fisher outperformed apparently the Laplacian one over 4 out of 5 datasets, particularly when the SVM used as final evaluator. However, in the case of Breast, the IDGA-L performed slightly better with 99.7% (against 98.8%) and a lower standard deviation. Meanwhile, IDGA-L obtained a considerably higher number of genes than that of IDGA-F. Consequently, a conclusion could not be drawn about the supremacy of either one. The IDGA-F could find solutions of the highest quality containing very few numbers of genes as shall be discussed in 'explored genes' section. In the case of DLBCL, the proposed method outperformed other methods in the literature, especially when considering both the number of selected genes and obtained performance. However, the DLBCL with Laplacian obtained higher average performance than that of Fisher with very close performance to 100% but with a quite bigger number of genes. Nevertheless, the number of selected genes for an evolutionary method is imperative since several filtering methods could obtain perfect accuracy but with a large number of genes. On the other hand, the solutions with a smaller number of genes could be more easily interpreted and utilized by experts. In other datasets, the performance of IDGA with Fisher score was quite better than that with Laplacian score, measuring by all three classifiers. However, on Leukemia, the IDGA-L with Naïve Bayes performed considerably better than IDGA-F with 97.7% vs. 95.2, and with quite smaller number of genes with the average of 8.2 against 18.4.

On the other hand, without considering top explored solutions, the proposed method has a comparable performance with the literature methods even though it outperforms others in some datasets considering both accuracy and number of genes. However, without taking into account the number of genes, some statistical tests were performed to see whether the proposed method had a competitive performance with the literature ones.

Table 3 demonstrates the resulting *p*-values from testing the differences in means and variances of results of IDGA against other that of other methods. F-test was firstly used to determine whether the results of IDGA and literature come from two populations with unequal variance or not. The resulting *p*-values in Table 3 explain that there was a relatively significant difference between the variances of two groups in every dataset. However, the difference was not statistically significant in DLBCL and Prostate, under a significant level of 0.05. Afterward, the student-test was applied with the determined assumption to test the significant difference between the means of the performances of two groups over each dataset separately. The resulting *p*-values signified a statistically significant difference only in the case of DLBCL with *p*-value 0.04. This fact shows that the proposed method intuitively and statistically outperformed the comparing methods over DLBCL dataset, while, upon the other datasets, performs competitively with other methods, even without considering top explored genes.

On the other hand, apart from the average results, the potential of a method to obtain perfect results is worth noting. For instance, the average performance of the current approach is considerably higher on

Table 3
The obtained *p*-value from testing the differences in means and variances of the results of the proposed method and the literature ones for each dataset.

Methods	Datasets				
	SRBCT	Breast	DLBCL	Leukemia	Prostate
F-test	0.03	0.003	0.07	0.001	0.09
T-test	0.23	0.68	0.08	0.24	0.17

SRBCT than Breast. Despite that, the capability of that to detect excellent solutions with a tiny number of genes is significantly higher in Breast than that of SRBCT. Considering the average and the best results showed that none of the comparing methods outperformed the others over all datasets. In fact, each method has some particular potential only in some datasets. For examples, [47] achieved the highest performance in Breast, without considering the number of genes, while it performed poorly in DLBCL, or [9] outperformed other methods in Prostate using DRF-IG. However, it did not have a good performance in Leukemia. The proposed method had its highest capability upon the Breast and the DLBCL datasets when considering all factors for comparison.

Another statistical test was performed addressing the issue that in which datasets the methods in the literature or the proposed method performed significantly better. In other words, which datasets were significantly easier being explored or whether was there any statistically significant difference between the performances of methods over different datasets. In this test, there were 5 treatment groups (the datasets) with unequal size. Two separate tests using one-way analysis of variance (ANOVA1) including one over literature results and one over the IDGA results were performed, independently. In each test, all observed performances were utilized. For instance, regarding the proposed method, all the single observations, from which the average results had been computed upon the Fisher and Laplacian, were employed in the test. After performing one-way ANOVA, the Bonferroni method [10], that is a conservative correcting procedure intended to keep the total type I error at most to the predetermined significance level, was applied to ANOVA situation to pick out the multiple comparisons [35]. Table 4 shows the resulting *p*-values for both tests.

As is clear, the performance of proposed method over three datasets including SRBCT, DLBCL, and Leukemia had a significant difference with that in Prostate, with the *p*-values bolded in Table 4, under significance level 0.10, while no considerable difference was seen in the performance of literature methods over different datasets. In other words, Literature methods perform equivalently over various datasets. Despite that, the proposed method had intuitively poorer performance upon Prostate than others. Nevertheless, its average performance did not have a significant difference comparing with the literature results as was shown in Table 3. Besides, looking more closely at Table 2, it is clear that there was no accuracy of 100% within the results of the literature in Prostate and DLBCL while there existed for others. That indicates the complexity of Prostate dataset which originated from high correlation between gene expression values in Prostate. Overall, the proposed method performed equally with existing approaches in Prostate while significantly performed better in DLBCL and Breast.

Table 4
Testing the differences between means of the observed performance in each datasets using ANOVA1 and Bonferroni multi-comparison procedure. Each pair of cells contains the resulting *p*-values of two independent tests, one with the results of proposed method (P) and one with the reported results in the literature (L).

Method	Breast		DLBCL		Leukemia		Prostate	
	P	L	P	L	P	L	P	L
SRBCT	1.0	0.65	1.0	0.38	1.0	0.77	**0.02**	0.33
Breast	–	–	1.0	1.0	1.0	0.95	0.63	1.0
DLBCL	–	–	–	–	1.0	0.93	**0.06**	1.0
Leukemia	–	–	–	–	–	–	**0.03**	0.87

3.2. Choice of classifiers

The choice of the classifiers is critical in every evolutionary process to identify high-quality solutions, particularly when used within an evolutionary process to obtain the fitness of individuals. It could even mislead the searching process by, for instance, prioritizing some poor individuals or overestimating them, and accordingly leads the searching process to a superficial optimal end. We saw this peculiar evidence in the case of Breast dataset. The KNN classifier that is among the fastest classifiers [28] was first adopted for fitness evaluation. The IDGA approached very fast to a superficially optimal end with almost 100% accuracy over the training samples. There were many high-accuracy individuals within the population. However, the quality of the obtained solutions for predicting the testing samples were quite poor. The KNN was not overfitted with 10-fold cross validation. In fact, the KNN was not a proper classifier for this dataset as shall be discussed later. Therefore, the IDGA was applied with the SVM classifier and the same behavior as that of KNN was detected again. As a result, we investigated Naïve Bayes” Behavior considering multiple assumptions about its predictor's distribution including ‘normal distribution’, ‘Multivariate Multinomial Distribution’, and ‘Kernel Distribution’. Normal distribution is based on the assumption that the samples belong to a given class have the normal distribution. This assumption is expected for each class separately. Running the IDGA using Naïve Bayes with this assumption did not lead to a good result even though that the aforementioned behavior was not observed and the results were improved over KNN and SVM. The multinomial or multivariable multinomial distribution is for categorical observations and could not be applied to real-valued variables. Finally, Kernel distribution that is appropriate for observations that have continuous distribution was utilized. This distribution does not require a strong assumption like the normal distribution and can be used in cases “where the distribution of a predictor may be skewed or have multiple peaks or modes” [52]. We identified that Naïve Bayes with this distribution worked strongly. Also, the supremacy of Naïve Bays over other classifiers in Breast was statistically confirmed (Table 4). This further augments that the Naïve Bays was the only classifier that significantly outperformed others in case of Breast cancer dataset. Thus, the reported results for Breast in Table 2 produced by the Naïve Bayes with Kernel distribution.

Furthermore, considering the results in Table 2 and the number of samples in each dataset, it is obvious that the KNN performed almost equally with Naïve Bayes when there was a higher number of samples. Fig. 3 shows the performance of KNN classifier along with the Naïve Bayes over datasets, sorted ascending based on their number of samples (took from Table 1). As is clear, the KNN performed worse over the datasets with very few number of samples. Nonetheless, it performs better or almost equally in larger datasets.

In the case of DLBCL, the KNN had a poor performance rather than others. Considering Fig. 3, this is the second dataset with the lowest number of samples. Several trial and error showed us that Naïve Bays with normal distribution could have found better results than the KNN, but the SVM dramatically outperformed both of them. Therefore, the SVM was employed for guiding the evolutionary process in DLBCL. Considering Table 2 for DLBCL, the SVM substantially outperformed other classifiers, particularly the KNN, using either of filter methods. It achieved 99.7% performance over Laplacian filter with a very low standard deviation and attained 98.8% performance over Fisher's filter but with a very lower number of genes. Nonetheless, in a statistical test in the next section, Table 10, that considered only the prediction accuracies, it was observed that only the KNN had a significantly lower performance than others and there was no statistically significant difference between the performance of SVM and Naïve Bayes upon Fisher-score. However, regarding the Laplacian filter, the SVM significantly

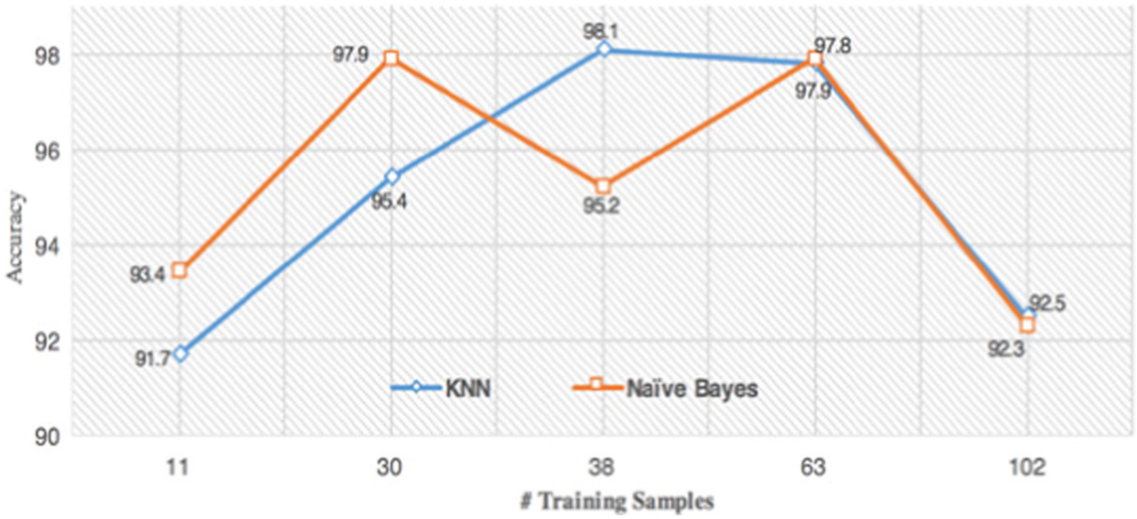


Fig. 3. KNN and Naïve Bays against datasets with different number of samples. The horizontal axis indicates the number of training samples for each dataset took from Table 1 sorted ascending. 11 is the number of samples of Breast data, 30 the DLBCL, 38 the Leukemia, 63 the SRBCT and 102 the Prostate data.

outperformed others while no statistically significant difference was observed between the performance of KNN and NBY.

Furthermore, it was found that the SVM classifier had dramatically better performance to guide the search process for Prostate, Leukemia and SRBCT as has already been successfully exploited within the evolutionary process or for cancer classification by [26,39,45,89,93] However, its time complexity was considerably higher than that of KNN.

From Table 2, it is evident that the SVM, as a final evaluator, outperformed the other classifiers with either of the filter methods in SRBCT, DLBCL and Prostate datasets. However, in Leukemia and Breast, its performance was competitive with that of KNN and Naïve Bays. Overall, one could draw a conclusion that the SVM (as a final evaluator) with proposed method outperformed other classifiers and also the Fisher score selected quite better genes.

3.3. Choice of filter methods

The quality of top selected genes and the influence of using either of the aforementioned filter methods were studied in this section. Table 5 demonstrates the quality of top selected genes by the filter methods over six subsets of top genes for each dataset which were cross-validated by three classifiers using LOOCV validation. At first glance, the best subset among six subsets, considering all datasets, is the top 10 genes selected by Fisher score in Breast that had a 100% performance using the Naïve Bays classifier. It seems that the best performance of the filter methods was on Breast and DLBCL. However, the proposed method and other hybrid methods in the literature obtained higher average performance with fewer numbers of genes than these filtering methods. In addition, it was not clear that how many genes should have been selected to achieve higher performance. Hence, six numbers were selected after some trial and error in a way that could be more comparable with the results of other methods. In the following, we raise two questions, and accordingly, four statistical tests were performed on the results of Table 2 and Table 5, each separately. The first question is whether there is a significant difference between the quality of top genes selected by Fisher and Laplacian. In other words, in which datasets the difference in the quality of selected top genes by two filter methods is statistically significant if there is any! For this test, all observed performance, including the performance of all classifiers over 6 points that were reported in Table 5, were considered. In this case, the studentized *t*-test in conjunction with f-test was used. First, f-test was utilized to determine whether these two groups had the same variance or not. In this test, the performance of all classifiers on every point, for each dataset separately, was considered. The null hypothesis was the two groups had the same variance. The test was performed at the significance level of 0.05. Table 6 shows the resulting *p*-value and F-statistics. The F critical one-tail under 0.05 and 0.01 significance levels are 2.27 and 3.24, respectively, with 17 degrees of freedom.

Considering Table 6, it is apparent that a statistically significant difference existed between the variance of the observed performance of two ranking methods over Leukemia, DLBCL and Prostate. However, it was not convincing enough to reject the null hypothesis in the case of

Table 5
The LOOCV results of top features of 5 datasets selected by two filtering methods namely Fisher and Laplacian Score using three distinct classifiers including K-nearest Neighbor (KNN), Naïve Bayes (NBY) and Support Vector Machine (SVM) applied upon 6 subsets of top features from 1 top feature to 500 top ones.

Dataset	Filter Method						
	#Top	Fisher-Score			Laplacian-Score		
		KNN	NBY	SVM	KNN	NBY	SVM
SRBCT	1	48.2	50.6	42.2	39.8	45.8	68.2
	5	94.0	95.2	92.8	82.0	86.8	68.2
	10	95.1	96.4	92.8	88.0	92.8	81.8
	50	98.8	**100**	98.8	95.2	**100**	63.6
	200	**100**	**100**	**100**	98.8	**100**	68.2
	500	**100**	**100**	**100**	**100**	**100**	81.8
BREAST	1	90.9	86.4	68.2	72.7	77.3	68.2
	5	77.3	95.5	86.4	59.1	77.3	68.2
	10	81.8	**100**	90.9	77.3	95.5	81.8
	50	81.8	**100**	72.7	68.2	95.5	63.6
	200	72.7	**100**	77.3	77.3	95.5	68.2
	500	68.2	**100**	68.2	68.2	95.5	81.1
DLBCL	1	89.4	91.5	89.4	51.1	53.2	02.1
	5	93.6	93.6	93.6	51.1	53.2	04.3
	10	95.7	93.6	95.7	51.1	57.4	02.1
	50	95.7	97.9	97.9	63.8	**100**	72.3
	200	95.7	**100**	**100**	74.5	**100**	78.7
	500	89.4	**100**	72.3	78.2	**100**	74.5
LEUKEMIA	1	83.3	88.9	82.4	68.1	52.8	65.3
	5	93.1	93.1	89.0	83.3	88.9	27.8
	10	91.7	97.2	91.2	81.9	61.1	70.8
	50	91.7	97.2	89.7	84.7	93.1	84.7
	200	93.1	98.6	91.7	91.7	98.6	91.7
	500	95.8	**100**	91.6	84.7	98.6	73.6
PROSTATE	1	77.9	62.5	82.3	82.4	60.3	49.3
	5	92.6	59.6	89.0	89.0	58.8	55.1
	10	91.9	58.1	91.2	91.2	61.0	70.6
	50	91.9	61.8	89.7	89.7	64.0	71.3
	200	91.9	59.6	86.0	86.0	66.2	83.1
	500	89.0	58.8	66.2	66.2	69.1	81.6

The perfect accuracies are bolded within the table.

Table 6
The results of testing differences between variances of observed results over top selected genes of Fisher and Laplacian methods, without applying IDGA, for each set using F-test at significance level of 0.05.

Results	Datasets				
	SRBCT	Breast	DLBCL	Leukemia	Prostate
F-statistics	1.1	1.0	22.4	14.8	2.5
p-value	0.42	0.49	~0	~0	0.03
Variance pair	(348,386)	(137,139)	(954,43)	(325,22)	(83,207)

SRBCT and Breast. Moreover, testing under significance level of 0.01 leaded to accept the alternative assumption in the case of Prostate.

Afterwards, the *t*-test was performed for comparing the means of the observed performance over each dataset, with the identified assumption (whether the numbers had equal or unequal variance took from f-test). The resulting critical two-tail values and t-stats are reported in Table 7. This test was also performed under significance level of 0.05.

Comparing the t-statistics with two-tail critical values revealed a statistically significant difference between the quality of the selected genes by Fisher and Laplacian. One consideration is the significance places come from where the variances were significantly different (as observed by F-test) even under significance level of 0.01. However, we see that there is not such evidence about the results of the proposed method. To this regard, a similar test was performed over all observed performances of the proposed method to see where the IDGA performed significantly better when a particular filter method had been used. In fact, this test dealt with the influence of the filter methods upon IDGA for each dataset, separately. Note that, to perform this test, the results of IDGA should only be used. The *t*-test with f-test was used to perform this test under the significance level of 0.01. Table 8 exhibits a brief report that shows in which significant difference lies on under the significance level of 0.01.

Furthermore, studding other statistical tests is also a worthwhile endeavor. The one-way ANOVA was also adopted to test the differences to see whether its outcomes were different from *t*-test or not. It was observed that the results of both tests (ANOVA1 and *t*-test) were in agreement regarding the significant regions at the levels of 0.05 and 0.01. Moreover, considering Table 7 and Table 8, the common location of significancy is upon the results on Prostate. It is worth reminding that the performance of two filtering methods to select top genes was overtly different upon Prostate with only one common gene out of 500 ones, as discussed in Fig. 1; but in case of other datasets, there were enough similarity, at least 262 out of 500 genes, to expect performing almost equally. Overall, these results show that the choice of filter method can significantly affect evolutionary search results. Nonetheless, it does not mean that a filter method performs better in all datasets even so it could be said that it performs relatively better in some datasets.

It is worth noting that it is erroneous to test the average results that were reported in Table 2, instead, all single observations from which the average values were computed must be considered. In fact, those values are roughly comparable, i.e., the mean values with the single observations. Nevertheless, we performed several tests to see where the results of such erroneous comparison comes close to real results; for instance, when such single observations do not exist. One-way ANOVA, *t*-test in conjunction with f-test, and Welch's test were inquired. It was surprisingly observed only the results of Welch test, especially when population variance is computed instead of sample variance, were nominally congruent with that of single observations under 0.01.

Table 7
The results of testing differences between the means of two filter methods' results, without applying IDGA, using *t*-test at 0.05 for each dataset.

Values	Datasets				
	SRBCT	Breast	DLBCL	Leukemia	Prostate
t-Statistics	−1.3	−1.9	−4.6	−3.3	−3.4
Critical two-tail value	2.03	2.03	2.09	2.09	2.05

Table 8
The results of significance test for comparing the means and variances of IDGA's results between Fisher and Laplacian for each datasets using *t*-test and f-test, respectively.

Methods	Datasets				
	SRBCT	Breast	DLBCL	Leukemia	Prostate
F-test	no	no	sig	sig	no
T-test	sig	no	no	no	sig

'sig' states significant difference and 'no' states no difference.

Another question to address in this section is which classifiers performed better for a specific dataset. In other words, was there any statistically significant difference between the performances of the classifiers on a specific dataset? This test was performed under two different objectives including the case when the filter methods apply without the IDGA and with the IDG. To conduct this test, the reported results in Table 2, for the first objective, and all original observations using which the results of IDGA were computed and showed in Table 2, for the second objective, were utilized, respectively. Considering the first objective i.e., the difference when using only the filter methods, the test is performed for either of 5 datasets considering each filter methods separately. Thus, $5 * 2 = 10$ tests, two tests upon each dataset, were performed. Note that, in each test, there were three groups (treatments) that were classifiers, and 18 samples that were the performance of classifiers over 6 points in a specific filter method for a dataset. In each test, one-way ANOVA with Bonferroni procedure was applied with 17 degrees of freedom. Interestingly, there was not found any statistically significant difference except in the case of Breast dataset. In Breast, the Naïve Bayes significantly outperformed other classifiers in both filter methods, the evidence that also were seen in our trial and error and within the top explored genes as shall be discussed later. Table 9 depicts the observed *p*-values from this test. Considering Table 9, the performance of SVM and KNN are in complete harmony over either of filter methods. Nonetheless, under the significance level of 0.01, the SVM does not have a significant difference with NBY when using the Laplacian method. As that was intuitive, the performance of Naïve Bayes was significantly different with other classifiers in this dataset.

The same tests using one-way ANOVA were performed considering the second aforementioned objective, upon the observed performance of IDGA, concerning the differences of classifiers upon each dataset when using a particular filter method with IDGA. To perform these tests, again, the results of all independent runs should be considered.

There were three groups i.e., the three classifiers each contains at least 7 values from 7 independent runs leading in a total of 18(21-2) degrees of freedom. Ten tests were performed unveiled that the performance of the classifiers was significantly different in DLBCL and Prostate. Therefore, only the obtained *p*-values of these two datasets are demonstrated in Table 10. Considering this table in DLBCL, the performance of KNN is considerably different with others. It is worth recalling the Fig. 3 in the previous section where the KNN considerably performed poorer than the other classifiers. That point was statistically

Table 9
The *p*-values observed by testing the significant difference between the means of the performance of three classifiers over top selected genes by two filter methods, without applying IDGA, in Breast dataset.

Classifiers	Methods			
	Fisher		Laplacian	
	KNN	SVM	KNN	SVM
SVM	1.0	–	1.0	–
NBY	**0.003**	**0.001**	**0.007**	0.013

The significant p-values are bolded.

Table 10
The obtained *p*-values of the significance test for comparing differences between means of classifiers' performance over either of filter methods, Laplacian and Fisher score, in DLBCL and Prostate datasets.

Datasets	DLBCL				Prostate			
Methods	Fisher		Laplacian		Fisher		Laplacian	
Classifiers	KNN	SVM	KNN	SVM	KNN	SVM	KNN	SVM
SVM	0.004	–	0.001	–	0.0	–	~0	–
NBY	0.04	0.96	0.72	0.001	1	0.0	~0	~0

Table 11
Obtained genes in Breast dataset.

Data	Breast			Genes (2)	
	Classifier			F-Id	D-Id
	SVM	KNN	NBY		
Train	100	100(90.9)	100	122, 389	1148, 2992
Test	100	90.9(81.8)	100		
LOOCV	100	100(95.5)	100		

confirmed by this test. However, using the Laplacian filter, the SVM significantly outperformed the others, with the average performance of 99.7% (with very low standard deviation) versus <93.5% for others. This fact was also statistically confirmed with the *p*-value of 0.001. On the other hand, in Prostate, all classifiers had significantly different performance with each other. Meanwhile, the SVM outperformed other classifiers intuitively and statistically considering both Table 2 and Table 10. As such, the performance of the KNN and NBY were in harmony when using Fisher criterion.

It is worth noting that over every dataset that there was a significant difference, it was for both filter methods, and when there was no difference, it was not for both of them. This evidence indicates that either of filter methods did not have a significant influence on the performance of classifiers. In fact, the complexity if a dataset is the most influential itself that induces some classifiers performs significantly better. Yet, the feature reduction step could either enhance or degrade the classifiers' performance.

3.4. Top explored genes

Top subsets of genes found by the present study are reported in Tables 11–16. Three classifiers were used to evaluate the explored genes including SVM, Naïve Bayes (NBY) and KNN. NBY was applied once assuming the normal distribution and once with kernel distribution. The results of NBY with Kernel are shown in parenthesis where there were different results. Thus, in the case of similar results, only one number is shown. SVM was used only with the linear kernel. Other kernels such as polynomials were not shown since we found no preferences over linear one except that were quite more time-consuming. The KNN was applied with K = 1 and K = 3 separately. The results of that using K = 3 are shown in parenthesis when different results produced.

In Breast, two high-quality subsets of genes were found, one with six genes and the other with only three genes. We studied the three-gene-subset and found that by removing one of them, the results would be improved and this is why only two genes are reported in Table 11. Using only two explored genes, the Naïve Bayes and SVM achieved 100% accuracy for LOOCV, training and testing samples. However, KNN had a lower performance particularly over testing samples with <82% and 92% accuracy setting up the parameter K to three and one respectively. Furthermore, KNN had a promising accuracy 100% in the 6-gene-subset for each partition using K = 1. However, with K = 3, its performance declined. That originated from the tiny number of training samples as aforementioned in Fig. 3. In fact, the proper value for K in this dataset was one as discussed in previous sections. Furthermore, evaluating top explored genes revealed that the Naïve Bayes with normal distribution had a quite higher performance than that of Kernel one whereas using NBY with kernel distribution during the evolutionary process dramatically distinguished better genes than that of the normal distribution. This phenomenon could be a future-work and shall not be further discussed Moreover, considering Table 5 in the previous section, it can be seen that Naïve Bayes with 10 top genes scored by Fisher method achieved 100% LOOCV accuracy. It was motivating to see if a fewer number of those top genes could repeat such accuracy or not. After some trial and error, it was viewed that the top 6 genes led to 100% LOOCV with Naïve Bayes. On the other hand, two obtained genes were not even among top 100 ranked genes by Fisher score. In fact, that was the evolutionary method that resulted in identifying such high-quality subset with few number of genes ever reported in the literature for this complex dataset. Furthermore, a subset of 9 highly informative genes was found in DLBCL dataset. That was, to our knowledge, the best subset ever reported in the literature. It had 100% prediction accuracy using SVM in LOOCV and both training and testing samples as shown in Table 13. However, the KNN had slightly lower performance, particularly when K was set to 3. Looking more closely, it is evident when changing K, the performance of KNN differed only upon testing samples. It was a significant evidence once seen during the evolutionary process. In fact, setting K to 3 within the evolutionary algorithm and set that back to 1 for final evaluation over test data, resulted in better results while other options such as setting K to 1 and back to 1, K to 1 and back to 3 or to 3 and back to 3 did not produce better results on this dataset. This fact might be because of the very few number of observations as discussed in the previous section.

At Prostate, the best subset of genes was picked concerning the number of genes and accuracy (see Table 14). In this dataset, the proposed method rarely found such solutions containing few number of genes while being highly discriminant. The average number of genes it found was about 30 genes as reported in Table 2. The prediction performance of this subset was not >96.3% and 91.2% in LOOCV and testing

Table 12
Obtained genes in Breast dataset.

Data	Breast			Genes (6)	
	Classifier			Fisher-Index	Database-Index
	SVM	KNN	NBY		
Train	100	100(91.0)	100	25, 193, 268, 302, 343, 459	2219, 985, 1663, 549, 530, 132
Test	91.0	100(91.0)	100(91)		
LOOCV	100	100(91.0)	100(95.5)		

Table 13
Obtained genes in DLBCL dataset.

Data	DLBCL			Genes (9)		
	Classifier			Fisher-Index	Database-Index	Accession code
	SVM	KNN	NBY			
Train	100	100	100	42,100,77,264,5,447, 287,43,362	2060,3884,2442,2226,1291,1552,2309,1289,1857	GENE3340X, GENE3314X, GENE2639X, GENE2897X, GENE2010X, GENE1233X, GENE1150X, GENE81X, GENE1554X
Test	100	100(94.1)	88.2			
LOOCV	100	97.2	95.5			

Table 14
Obtained genes in Prostate dataset.

Data	Prostate			Genes (14)		
	Classifier			Fisher-Index	Database-Index	Accession code
	SVM	KNN	NB			
Train	100	100(97.1)	92.2(100)	37,52,86,129,141, 218,236, 286, 313, 332, 358, 365, 406, 419	10234,9093,6145,9201,6575,7623, 9010,6842,3120,7465,1812,9174, 5862,11774	32929_at, 37827_r_at, 36836_at, 37599_at, 38986_at, 39732_at, 32151_at, 32786_at, 37765_at, 38087_s_at, 38408_at, 38435_at, 41504_s_at, 960_g_At.
Test	91.2	32.4(79.4)	26.5(73.5)			

Table 15
Obtained genes in Leukemia dataset.

Data	Leukemia			**Genes (15)**		
	Classifier			Fisher-Index	Database-Index	Accession Code
	SVM	KNN	NBY			
Train	100	100	97.4	32, 35, 144, 171, 178, 186, 221, 260, 357, 379, 395, 409, 411, 439, 440	1120, 1156, 1456, 1906, 2121, 2426, 2592, 3334, 3605, 3740, 4211, 4710, 5314, 6645, 6768	J04615_at, K01383_at, L29277_at, M29551_at, M63138_at, M98539_at, U02020_at, U50928_at, U67963_at, U78628_at, X51521_at, X84194_at, L33075_at, U90552_s_at, U90546_at
Test	88.2	94.1(88.2)	88.2			
LOOCV	100	97.2	93.1			

samples, respectively. One evidence is the accuracy of KNN with K = 1 was quite lower than that with K = 3. It has been already seen in the literature that KNN makes poor models with K = 1 especially when the number of samples increases [95]. This issue was expected for Prostate that had a quite higher number of samples than others, while in contrast, in the case of DLBCL that there were very few samples, the KNN with K = 1 was able to build more precise models.

One important evidence in the Prostate dataset was that the Naïve Bayes with Kernel distribution obtained dramatically higher performance than that with the normal distribution. This fact not only was evident in this subset but also was seen in many trial and error that have been performed. It was perhaps because the distribution of its samples was skewed, the case for which the kernel distribution is designed! [52].

The SVM classifier achieved 100% LOOCV performance in Leukemia using only 15 genes as detailed in Table 15. However, the accuracy achieved over the testing samples was not >88.2%. The KNN obtained its best performance over the testing samples with 94.1% while, at the same, achieved 100% performance on training samples. However, its LOOCV performance was slightly lower than SVM. Considering Table 5 in the previous section, the filter methods, solely, achieved the same performance as that of IDGA, but with a quite higher number of genes, i.e., 500 genes using Fisher score. Furthermore, the KNN had the same performance using either K = 1 and K = 3 over training samples and LOOCV. One evidence is that the same pattern does not exist among the performance of classifiers as that of filter methods in Table 5 in Leukemia. For instance, the SVM with these genes obtained highest and the KNN the lowest performance while in filter methods, consider Table 5, the highest performance is for NBY and the lowest for the SVM. Comparing this result with the literature cases reveals that the proposed method obtained the same performance with a lower number of genes.

Considering Table 16, a perfect performance was obtained using SVM by 18 genes in SRBCT. The KNN had a better performance using these genes with K = 1 than K = 3. In this dataset, NBY had considerably lower performance over testing samples with 60% accuracy than others with 85% accuracy. However, within the reported average results in Table 2, only the SVM had slightly higher performance while similar average results were perceived among other classifiers. On the other hand, considering Table 5, in the case of top ranked genes, the NBY classifier achieved quite higher accuracy, either by Fisher or Laplacian, than others with 100% performance over 50 genes. Perhaps, that was because of the type of classifier used within the evolutionary method.

3.5. Algorithmic behavior

The algorithmic behaviors of the IDGA method were studied concerning the algorithm "s convergence trends, its running time, the cross-over and mutation rate changes. The following subsections detail various characteristics of the IDGA. Some implementation issues relating to the current study is also addressed.

3.5.1. Likelihood changes

In this section, the changes in mutation likelihood and crossover rate were being studied over several iterations in a sample run. Fig. 4 shows the average crossover rate and mutation likelihood of the entire population over 100 iterations in an independent experiment upon Prostate dataset. According to Fig. 4 (a), the average crossover rate ranged between 0.60 and 1.0. At the first 20 iterations, the population had an average likelihood of near 0.90. However, just then, there was a sharp decline at the 20th iteration with crossing rate of near 0.75. This evidence denotes the quality of population at that iteration.

The other relevant evidence is the exceptional behavior of the proposed method to maintain diversity. Maintaining diversity in a population is a key task for every evolutionary algorithm toward optimal convergence [22,38]. That actually helps the algorithm to explore more regions of search space. In fact, it is the nature of encouraging and penalizing strategy that compels the individuals to have binding interactions (whether lower or higher) to maintain diversity. Accordingly, this strategy performs by adjusting a trade-off between exploration and exploitation by suggesting lower crossover and mutation rate for

Table 16
Obtained genes in SRBCT dataset.

Data	SRBCT			Genes (18)		
	Classifier			Fisher-Index	Database-Index	Accession code
	SVM	KNN	NBY			
Train	100	100 (92.5)	98.4	3,4,28,46,57,62,112, 113,198,235,278,279, 305,380,389,405,431, 462	1158, 742, 998, 1327, 719, 174, 1795, 1318, 2117, 753, 1758, 2166, 554, 2175, 1335, 2053, 801, 1661	814,526, 812,105, 785,793, 491,565, 159,455, 769,716, 45,291, 855,487, 139,957, 752,652, 363,103, 296,616, 461,425, 246,194, 590,759, 161,195, 755,239, 509,943
Test	85.0	90 (85)	60.0			

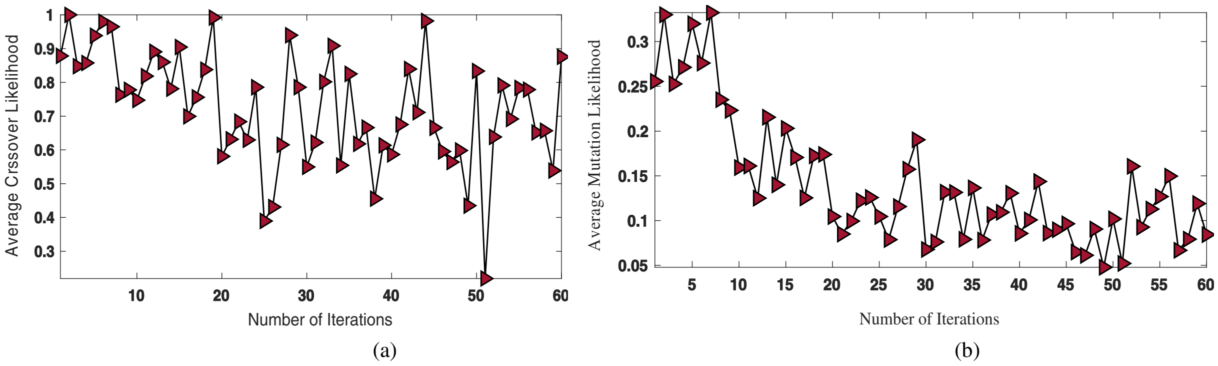


Fig. 4. Average crossover and mutation rates: (a) the average crossover rate of whole population over 100 iterations and (b) the average mutation rate of whole population in each generation.

superior individuals to improve 'local search' and bringing about the possibility of 'backward-search' by holding such superior individuals while forcing the poor individuals to have more interactions to achieve more exploration.

Further, as the evolutionary process progressed, considering crossover diagram in Fig. 4 (a), the higher number of likelihood values approached toward expected value of Eq. 2, i.e. 0.05, Such behavior was more conspicuous in the last 20 iterations. Moreover, the general trend of crossover rate, while erratic, is descending. Additionally, looking more closely, it is evident that most of the points are near 0.80. This evidence was previously seen in 30 independent runs. Furthermore, the average value of the crossover rate of the whole population converged toward 0.80 in an independent run. This rate has been used and recommended in several engineering applications [5,75].

Considering the mutation likelihood in Fig. 4 (b) reveals that its average rate varied between 0.05 and 0.40 over iterations. There were high mutation rates at the beginning iterations, similar to what was observed in crossover diagram. Such high rate was expected at the first few steps since the individuals had not evolved yet. In fact, at the few beginning iterations, the mutation rate was near to 0.05, the expected value of the second part of Eq. 3. However, whatever the evolutionary process progressed, the mutation rate approached to lower values such as 0.10 as can be seen in the sample experiment shown in Fig. 4 (b). Such rate (0.10) is intuitively big, particularly when the number of features is high. However, we would see in Fig. 5, in the following section, that the length of elite chromosomes approached very soon toward small numbers usually lower than 40 after only ten iterations and reached soon even to half of that after 20th generation for several datasets. Therefore, with the average likelihood of 0.10, it is expected to have at most 4 out of 40 genes and 2 out of 20 genes being altered; That is a quite adequate mutation for the local search around superior and ordinary individuals.

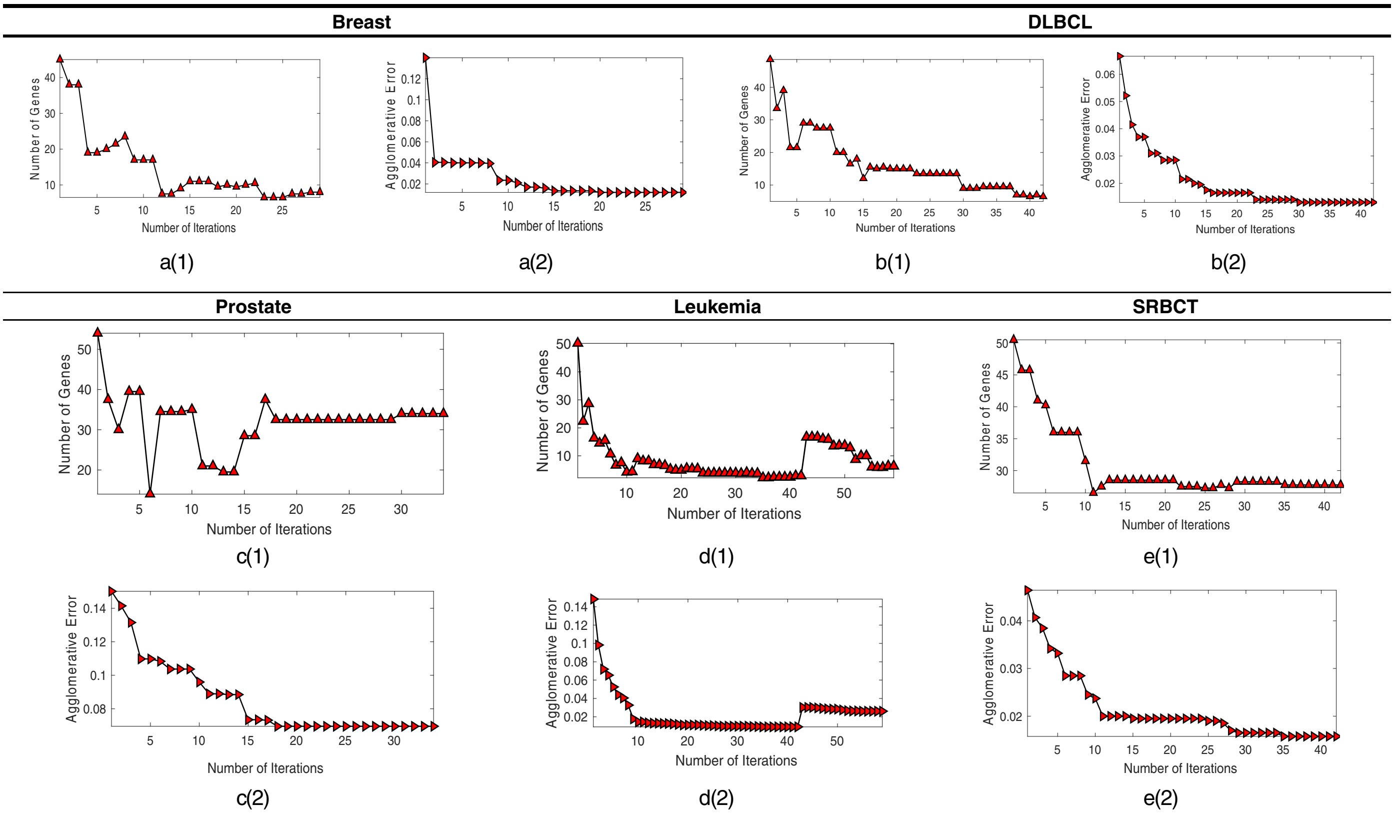


Fig. 5. Convergence behavior in terms of the number of genes and the prediction error over 5 datasets in a sample independent run: figures a(1), b(1), c(1), d(1), and e(1) demonstrate the convergence trends for reducing the number of genes within elite chromosomes while figures a(2), b(2), c(2), d(2), and e(2) display the general trends in terms of agglomerative error.

3.5.2. Convergence trends

The convergence behavior of the proposed algorithm is being studied in this section. For each of dataset, two types of graphs are displayed in Fig. 5 involving: (a) the convergence trend regarding the average number of genes within the elite individuals. That reveals how the proposed method reduced the number of genes in chromosomes gradually over iterations and (b) the convergence trend regarding the agglomerative error that depicts how the proposed method progressively decreases the prediction error over iterations. It should be reminded that the agglomerative error was a combination of prediction error and the slight effect of the number of genes in an individual. Recall that the effect of the number of genes in the presented fitness function was adjusted by the contracting factor Ç to be a multiple of 0.001. It should be noted that all diagrams were obtained in training samples. Hence, reaching the prediction error of 0.0 does not indicate the same performance as that of testing samples or LOOCV. In addition, for each dataset, the proposed method was performed with the same implementation as already utilized in experimental results, to name a few, the maximum number of iteration is set to 60, the Elite chromosomes to 2, population number to 10; in addition, the same stopping criterions, the SVM as a part of fitness evaluator, the random initialization that was performed once only, and the maximum number of genes in the primary population was set to be a random number between 50 and 100.

Fig. 5 demonstrates convergence diagrams for each dataset in an independent sample run. According to Fig. 5, the graphs a(1), b(1), c(1), d(1), and e(1) demonstrate the convergence trends in respect of number of genes while the graphs of a(2), b(2), c(2), d(2), and e(2) depict the declining trends regarding the prediction error in each dataset. According to Fig. 5, it is obvious that a general declining pattern existed in all diagrams. It is also remarkable that there were some wild fluctuations in the number of features. These fluctuations are more conspicuous in Prostate where there was a sharp leap in the number of genes from near ten genes to >30 genes after a steep drop at the first six iterations. This evidence, as can be seen in every diagram, denotes the great interactions within individuals that revealed even within few elite chromosomes. It is important to note that the oscillations in error diagrams seem conflicting with an expected outcome of elitism strategy. However, that was a right behavior since more than one elite chromosome were involved so that their average agglomerative error was drawn. Thus, it is possible to see the average error was increasing as can be seen in Fig. 5. d (2) after 40th iterations. This phenomenon is, in fact, the potent of the IDGA to maintain the diversity even within the elite individuals. Such interaction could also be seen in the other diagrams. For instance, on Prostate, at the beginning steps, the solutions with a smaller number of genes, even near ten genes, were selected as elite individuals but soon after the algorithm explored better solutions containing >30 genes. Other evidence is the effect of stopping criterions as can be clearly seen looking at the ending iterations in each diagram. The algorithm was stopped far before reaching the maximum iteration. There were >50 iterations only in case of Leukemia.

Overall, it could be said that the IDGA method with its dynamic representation and intelligent reinforcement strategies created a quite high interaction among population individual, as can be seen even within few elite ones, which resulted in a proper diversity and fast convergence. Also, it avoided premature convergence and getting caught in the vast local-minimum spaces through its intelligent adaptation for new individuals.

3.5.3. Running time

The average runtime of the proposed algorithm was studied. In the present study, the IDGA method and Simple Genetic Algorithm (SGA) with and without using feature reduction (FR) step were applied over SRBCT dataset in 20 independent runs. All implementations were performed under the same circumstances, the same hardware and software (as expressed in the section of Implementation Notes). Fisher Score was used as the feature reduction method and the KNN as the classifier of the comparing algorithms. The stopping criterion in all algorithms was reaching to 1000 iterations or achieving <0.10 cumulative error over training samples. 500 genes were used when performing an algorithm with feature reduction step. The proposed algorithm reached very fast to such error, but the SGA hardly touched it. Therefore, we imposed another criterion that ends the algorithm if reaching both total error of 0.10 and prediction error of 0.05. Table 17 shows the average error and runtime.

Table 17
The average performance obtained over 20 independent runs for each method in terms of Error and Time in second.

Methods	SGA	SGA-FR	IDGA	IDGA-FR
Error	0.89 (±0.004)	0.125 (±0.004)	0.041 (±0.013)	0.018 (±0.004)
Time/s	1270.7 (±23.4)	1156.3 (±36.3)	192.9 (±20.0)	173.3 (±16.2)

The number in parenthesis indicates the confidence interval of 95%.

As is clear, not only the average runtime of the proposed method was overtly better than the SGA but also its error was dramatically less than that of SGA method. One surprising point was that there were almost similar runtimes for algorithms with and without feature reduction. This issue perhaps was because the sufficient amount of RAM was existed for allocating whole arrays, or that was the operating system management to dispatch jobs to other CPU cores under loads. However, there was a remarkable difference between the runtime of proposed algorithm that was actually an extension of the genetic algorithm. Another evidence is that the feature reduction step had a great influence on the result of SGA with 0.12 error against near 0.90 error while at the case of IDGA, error difference is only 0.02. However, it is noteworthy that the IDGA method with reduction step obtained the solutions with a quite lower standard deviation (0.004) that was interestingly equivalent to the SGA and SGA-FR. To this regards, the one-way ANOVA with Bonferroni procedure was employed at the level of 0.05 to uncover whether the differences in the average runtimes were also statistically significant. Table 18 depicts resulting *p*-values from testing each pair of methods.

As is obvious there was a significant difference between the average runtime of the proposed method and the simple GA. However, it could not be seen any statistical difference when using feature reduction step. Such difference between IDGA and SGA, comes from the fact that the SGA moves slowly through the search space while the IDGA picks big steps by producing dynamic length chromosomes and escaping from search sub-optimal surface by random restarting and adaptive strategies, while in contrast, SGA does not benefit from any of them.

3.5.4. Implementation notes

In this section, some implementation issues and assumptions are described. First of all, in all experiments, the unchanged original test set and train set were used. No normalizations were performed. Moreover, the proposed method utilized only 500 top genes (as already offered in other research work) to obtain discriminant genes. However, its performance may be plunged using a lower number of top ranked genes or in contrast using whole feature space.

Another consideration is the scoring method was applied only using the train set as expected. Then, the top ranked genes (features) were

Table 18
The *p*-value matrix for average runtime differences between IDGA, SGA, IDGA with Feature Reduction (FR) and SGA with FR using one-way analysis of variance at significance level of 0.05 and Bonferroni procedure for multiple comparison.

Methods	IDGA-FR	SGA-FR	SGA
IDGA	0.4460	**0.0489**	**~0**
SGA	**~0**	0.2931	
SGA-FR	**0.0002**		

The character '~' defines as 'very close to' or less than 1e-6. The statistically significant differences are bolded!

evaluated using LOOCV. Furthermore, the number of population was experimentally set to 10, the number of iterations to 60, the random initialization to 5, and the elite chromosomes (that was directly transferred to the next generation) to 3. The range of length of chromosomes for creating initial population was a random number between 50 and 100. The breaking agglomerative fitness used as stopping criterion was set experimentally to 0.994. Recall the other stopping criterion in IDGA, the maximum number of iterations that the number of genes within the best individual stayed unchanged was experimentally set to 20 except for the SRBCT and Breast that was 15. The contracting factor was 0.50 for mutation operator. The maximum agglomerative fitness was 1.001. All implementations would be available online in mathworks.com searching for **IDGA** method.

All experiments were carried out on MacBook Pro MJLT2 with 16 GB RAM, Intel Core i7 CPU 3.7 GH and 512 GB of SDD storage using Matlab 2015b. Some figures and statistical test were obtained using Microsoft Excel 15 for mac. The classifiers employed in the present study were used with its default settings in Matlab unless otherwise stated.

4. Conclusion and future works

In this paper, a novel evolutionary algorithm, called intelligent dynamic genetic algorithm (IDGA), based on the concepts of genetic algorithms and artificial intelligence, for gene selection and cancer classification in microarray data, was proposed. The proposed algorithm mainly consisted of two parts. In the first part, a ranking method was used to reduce the dimensionality and to provide statistically significant genes as the input to the second step. Next, an integer-coded genetic algorithm with variable-length genotype, adaptive parameters and modified genetic operators was proposed. In the proposed algorithm, the concept of encouraging-and-penalizing strategy inspired by behaviorist psychology was employed to obtain crossover and mutation probability. The proposed adaptive relations were conceptually discussed and shown to be coherent with literature finding.

Furthermore, the presented method practically exploited the concept of random-restart hill climbing to avoid biased or improper initializations by initiating a few population of individuals with randomly generated length. The proposed intelligent dynamic genetic algorithm (IDGA) method was used along with two different ranking methods, the Fisher-score and the Laplacian-score. Both scoring methods were compared regarding the similarity and dissimilarity upon different datasets. Experimental results demonstrated that these filtering methods chose entirely different genes from prostate cancer data while their most similarity for other datasets was <48% among top 500 genes. Moreover,

The algorithmic behavior of the IDGA including the convergence trends and running time was also studied which revealed the fast convergence of the IDGA in detecting predictive genes. The proposed method was evaluated on five high-dimensional cancer datasets using three distinct classifiers, namely, SVM, KNN and Naïve Bayes. The proposed method obtained its best results in combination with Fisher-score and SVM. The experimental results were compared with several well-known and state of the art methods which depicted that the proposed method outperformed existing methods in DLBCL dataset. Moreover, several statistical tests were performed to analyze the quality of top selected genes by filter methods and to study the influence of using a filter method with IDGA. It was shown that the choice of filter method could significantly affect the obtained results in some datasets. Further, the effect of using three classifiers within evolutionary process was also studied that unraveled that the SVM outperformed other comparing classifiers in most datasets except the Breast in which the Naïve Bays with Kernel distribution outperformed others.

In conclusion, the study of possible effects of various crossover types in algorithm convergence could be an invaluable research work. For example, designing a uniform crossover version for variable-length chromosomes and investigating its effects to algorithm convergence. In addition, considering the median instead of the average point to incorporate in adaptive crossover relation is another case for further research.

Conflict of interests

We wish to confirm that there are no conflicts of interest associated with respect to this study, and no financial support has been received for the research that could have influenced its outcome.

Acknowledgment

We would like to express our gratitude to Dr. Prashanth Suravajhala, for sharing his pearls of wisdom and his invaluable time with us, and to Bioclues.org particularly Mohan Kumar Megha for final proofreading. Also, thanks to the revered reviewers for their insightful suggestions that made considerable improvements to our work.

Appendix A. Supplementary data

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.ygeno.2017.01.004.

References

[1] M.J. Abdi, D. Giveki, Automatic detection of erythemato-squamous diseases using PSO–SVM based on association rules, Eng. Appl. Artif. Intell. 26 (1) (2013) 603–608.
[2] A. El Akadi, et al., A two-stage gene selection scheme utilizing MRMR filter and GA wrapper, Knowl. Inf. Syst. 26 (3) (2011) 487–500.
[3] Z.Y. Algamal, M.H. Lee, Penalized logistic regression with the adaptive LASSO for gene selection in high-dimensional cancer classification, Expert Syst. Appl. 42 (23) (2015) 9326–9332.
[4] A.A. Alizadeh, et al., Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature 403 (6769) (2000) 503–511 Available at: http://www.nature.com/doifinder/10.1038/35000501 ([Accessed June 20, 2016]).
[5] A.T. Azar, S. Vaidyanathan, Computational Intelligence Applications in Modeling and Control, Springer, 2015.
[6] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, ACM Press, New York, 1999.
[7] A. Bandura, R.H. Walters, Social Learning and Personality Development, 1963 (JSTOR).
[8] J.R. Bienkowska, et al., Convergent random Forest predictor: methodology for predicting drug response from genome-scale data applied to *anti*-TNF response, Genomics 94 (6) (2009) 423–432.
[9] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, Distributed feature selection: an application to microarray data classification, Appl. Soft Comput. 30 (2015) 136–150.
[10] C.E. Bonferroni, Teoria Statistica Delle Classi e Calcolo Delle Probabilita, Libreria internazionale Seeber, 1936.
[11] R. Cai, et al., An efficient gene selection algorithm based on mutual information, Neurocomputing 72 (4) (2009) 991–999.
[12] J. Cao, et al., A fast gene selection method for multi-cancer classification using multiple support vector data description, J. Biomed. Inform. 53 (2015) 381–389.
[13] E. Capriotti, R.B. Altman, A new disease-specific machine learning approach for the prediction of cancer-causing missense variants, Genomics 98 (4) (2011) 310–317.
[16] D. Chen, et al., Selecting genes by test statistics, Biomed. Res. Int. 2005 (2) (2005) 132–138.
[17] X. Chen, H. Ishwaran, Random forests for genomic data analysis, Genomics 99 (6) (2012) 323–329.
[18] J.H. Cho, et al., Gene selection and classification from microarray data using kernel machine, FEBS Lett. (2004) 93–98.
[19] J.H. Cho, et al., New gene selection for classification of cancer subtype considering within class variation, FEBS Lett. (2003) 3–7.
[20] S.B. Cho, H.-H. Won, Cancer classification using ensemble of neural networks with multiple significant gene subsets, Appl. Intell. 26 (3) (2007) 243–250.
[22] D. Dasgupta, Z. Michalewicz, Evolutionary Algorithms in Engineering Applications, Springer Science & Business Media, 2013.
[23] N. Dessì, E. Pascariello, B. Pes, A comparative analysis of biomarker selection techniques, Biomed. Res. Int. 2013 (2013).
[24] J.L. Devore, Probability and Statistics for Engineering and the Sciences, Duxbury Press, California, 1995.
[25] S. Dudoit, J. Fridlyand, T.P. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, J. Am. Stat. Assoc. (2002) 77–87.
[26] V. Elyasigomari, et al., Cancer classification using a novel gene selection approach by means of shuffling based on data clustering with optimization, Appl. Soft Comput. 35 (2015) 43–51.
[27] D.B. Fogel, J.W. Atmar, Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems, Biol. Cybern. 63 (2) (1990) 111–114.

[28] F. Gagliardi, Instance-based classifiers applied to medical databases: diagnosis and knowledge extraction, Artif. Intell. Med. 52 (3) (2011) 123–139.

[29] B.A. Garro, K. Rodríguez, R.A. Vázquez, Classification of DNA microarrays using artificial neural networks and ABC algorithm, Appl. Soft Comput. (2015).

[30] T.R. Golub, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.

[31] P. Guo, et al., Gene expression profile based classification models of psoriasis, Genomics 103 (1) (2014) 48–55.

[32] E. Hancer, et al., A binary ABC algorithm based on advanced similarity scheme for feature selection, Appl. Soft Comput. 36 (2015) 334–348.

[33] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, Advances in Neural Information Processing Systems 2005, pp. 507–514.

[34] I. Hedenfalk, et al., Gene expression profiles in hereditary breast cancer, N. Engl. J. Med. (2001) 539–548.

[35] Y. Hochberg, A.C. Tamhane, Multiple Comparison Procedures, John Wiley & Sons Inc., 2008 Available at: http://doi.wiley.com/10.1002/9780470316672.ch1 ([Accessed June 12, 2016]).

[36] H.L. Huang, F.L. Chang, ESVM: evolutionary support vector machine for automatic feature selection and classification of microarray data, Biosystems (2007) 516–528.

[37] E. Huerta, B. Duval, J.-K. Hao, Fuzzy logic for elimination of redundant information of microarray data, Genomics Proteomics Bioinformatics 6 (2) (2008) 61–73.

[38] Y. Jin, Knowledge Incorporation in Evolutionary Computation, Springer, 2013.

[39] S. Kar, K. Das Sharma, M. Maitra, Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique, Expert Syst. Appl. 42 (1) (2015) 612–627.

[40] J. Khan, et al., Classification and diagnostic prediction of cancers using expression profiling and artificial neural networks, Nat. Med. 7 (2001) 673–679.

[41] M.W. Khan, M. Alam, A survey of application: genomics and genetic programming, a new frontier, Genomics 100 (2) (2012) 65–71.

[42] N.N. Khodarev, et al., Receiver operating characteristic analysis: a general tool for DNA array data filtration and performance estimation, Genomics 81 (2) (2003) 202–209.

[43] A. Kumar, et al., Identification of genes associated with tumorigenesis of meibomian cell carcinoma by microarray analysis, Genomics 90 (5) (2007) 559–566.

[44] C. Lazar, et al., A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinf. 9 (4) (2012) 1106–1119.

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

[46] K.E. Lee, et al., Gene selection: a Bayesian variable selection approach, Bioinformatics 19 (1) (2003) 90–97.

[47] Y. Lee, C.K. Lee, Classification of multiple cancer types by multicategory support vector machines using gene expression data, Bioinformatics 9 (2003) 1132–1139.

[48] G.-Z. Li, et al., Partial least squares based dimension reduction with gene selection for tumor classification, 2007 IEEE 7th International Symposium on Bioinformatics and BioEngineeringIEEE 2007, pp. 1439–1444.

[49] L. Li, et al., A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset, Genomics 85 (1) (2005) 16–23.

[50] B. Liao, et al., Gene selection using locality sensitive Laplacian score, IEEE/ACM Trans. Comput. Biol. Bioinf. 11 (6) (2014) 1146–1156 Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6824828 (Accessed June 19, 2016).

[51] W. Malina, On an extended Fisher criterion for feature selection, IEEE Trans. Pattern Anal. Mach. Intell. 5 (1981) 611–614.

[52] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press Cambridge, 2008.

[53] D. Mishra, B. Sahu, Feature selection for cancer classification: a signal-to-noise ratio approach, Int. J. Sci. Eng. Res. 2 (4) (2011) 1–7.

[54] M. Mohammadi, et al., Robust and stable gene selection via maximum–minimum correntropy criterion, Genomics 107 (2) (2016) 83–87.

[55] O. Mowrer, Learning Theory and Behavior, 1960.

[56] K. Nakai, M. Kanehisa, A knowledge base for predicting protein localization sites in eukaryotic cells, Genomics 14 (4) (1992) 897–911.

[57] S. Niijima, Y. Okuno, Laplacian linear discriminant analysis approach to unsupervised feature selection, IEEE/ACM Trans. Comput. Biol. Bioinf. 6 (4) (2009) 605–614 Available at: http://www.ncbi.nlm.nih.gov/pubmed/19875859 (Accessed June 20, 2016).

[58] S. Olyaee, Z. Dashtban, M.H. Dashtban, Design and implementation of super-heterodyne nano-metrology circuits, Front. Optoelectron. 6 (3) (2013) 318–326.

[59] 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 (8) (2005) 1226–1238.

[60] N.P. Pérez, et al., Improving the Mann–Whitney statistical test for feature selection: an approach in breast cancer diagnosis on mammography, Artif. Intell. Med. 63 (1) (2015) 19–31.

[61] V. Pihur, S. Datta, S. Datta, Finding common genes in multiple cancer types through meta–analysis of microarray experiments: a rank aggregation approach, Genomics 92 (6) (2008) 400–403.

[62] Y. Qi, et al., Ranking analysis for identifying differentially expressed genes, Genomics 97 (5) (2011) 326–329.

[63] M.E. Rahman, et al., MiRANN: a reliable approach for improved classification of precursor microRNA using artificial neural network model, Genomics 99 (4) (2012) 189–194.

[64] S.J. Russell, J. Stuart, P. Norvig, J. Canny, Artificial Intelligence: A Modern Approach, second ed. Prentice Hall, Upper Saddle River, New Jersey, 2003 Available at: http://aima.cs.berkeley.edu/ (Accessed June 23, 2016).

[65] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.

[66] N.I. Senaratna, Genetic Algorithms: The Crossover-Mutation Debate, Degree of Bachelor of Computer Science of the University of Colombo, 2005.

[67] D. Singh, et al., Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2) (2002) 203–209.

[68] M. Srinivas, L.M. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, IEEE Trans. Syst. Man Cybern. 24 (4) (1994) 656–667.

[69] F. Tan, et al., A genetic algorithm based method for feature subset selection, Appl. Soft Comput. (2007) 11–20.

[70] Y.-D. Tan, M. Fornage, Y.-X. Fu, Ranking analysis of microarray data: a powerful method for identifying differentially expressed genes, Genomics 88 (6) (2006) 846–854.

[71] R. Tibshirani, B. Hastie, C. Chu, Diagnosis of multiple cancer types by shrunken centroids of gene expression, USA: Proceedings of the National Academy of Sciences 2002, pp. 6567–6572.

[72] D.L. Tong, A.C. Schierz, Hybrid genetic algorithm-neural network: feature extraction for unpreprocessed microarray data, Artif. Intell. Med. 53 (1) (2011) 47–56.

[73] K. Tu, et al., Learnability-based further prediction of gene functions in gene ontology, Genomics 84 (6) (2004) 922–928.

[74] L.J. van't Veer, et al., Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (6871) (2002) 530–536.

[75] S. Voß, et al., Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization, Springer Science & Business Media, 2012.

[76] I. Vukusic, S.N. Grellscheid, T. Wiehe, Applying genetic programming to the prediction of alternative mRNA splice variants, Genomics 89 (4) (2007) 471–479.

[77] A. Wang, et al., Improving PLS–RFE based gene selection for microarray data classification, Comput. Biol. Med. 62 (2015) 14–24.

[78] Y. Wang, et al., Predicting human microRNA precursors based on an optimized feature subset generated by GA–SVM, Genomics 98 (2) (2011) 73–78.

[79] B. Whitworth, A.P. Whitworth, The social environment model: small heroes and the evolution of human society, First Monday 15 (11) (2010).

[80] Y. Xiao, et al., Predicting the functions of long noncoding RNAs using RNA-Seq based on Bayesian network, Biomed. Res. Int. 2015 (2015).

[81] W. Xiong, Z. Cai, J. Ma, ADSRPCL-SVM approach to informative gene analysis, Genomics Proteomics Bioinformatics 6 (2) (2008) 83–90.

[82] J. Xuan, et al., Gene selection for multiclass prediction by weighted fisher criterion, EURASIP J. Bioinforma. Syst. Biol. 2007 (2007) 3.

[83] A. Yang, et al., Bayesian variable selection with sparse and correlation priors for high-dimensional data analysis, Comput. Stat. (2016) 1–17.

[84] F. Yang, et al., Emphasizing minority class in LDA for feature subset selection on high-dimensional small-sized problems, IEEE Trans. Knowl. Data Eng. 27 (1) (2015) 88–101.

[85] J. Yang, et al., Applying the Fisher score to identify Alzheimer's disease-related genes, Genet. Mol. Res. 15 (2) (2016).

[86] K. Yang, et al., A stable gene selection in microarray data analysis, BMC Bioinf. 7 (1) (2006) 1.

[87] H. Yu, et al., A modified ant colony optimization algorithm for tumor marker gene selection, Genomics, proteomics & bioinformatics 7 (4) (2009) 200–208.

[88] J. Zahiri, et al., PPIevo: protein–protein interaction prediction from PSSM based evolutionary information, Genomics 102 (4) (2013) 237–242.

[89] D. Zhang, et al., A Genetic algorithm based support vector machine model for blood-brain barrier penetration prediction, Biomed. Res. Int. 2015 (2015).

[90] C.-H. Zheng, et al., Gene expression data classification using consensus independent component analysis, Genomics Proteomics Bioinformatics 6 (2) (2008) 74–82.

[91] M. Zhou, et al., Constraint programming based biomarker optimization, Biomed. Res. Int. 2015 (2015).

[92] N. Zhou, L. Wang, A modified T-test feature selection method and its application on the HapMap genotype data, Genomics Proteomics Bioinformatics 5 (3) (2007) 242–249.

[93] X. Zhou, D.P. Tuck, MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data, Bioinformatics 23 (9) (2007) 1106–1114.

[94] 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 (4) (2013) 1274–1281.

[95] L. Li, et al., Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method, Bioinformatics 17 (12) (2001) 1131–1142.