Genomics 112 (2020) 1916–1925

Contents lists available at ScienceDirect

Genomics

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

Original Article

Unsupervised feature selection algorithm for multiclass cancer classification of gene expression RNA-Seq data

Pilar García-Díaz[a,*], Isabel Sánchez-Berriel[b], Juan A. Martínez-Rojas[a], Ana M. Diez-Pascual[c]

[a] Department of Signal Theory and Communications, Polytechnic School, University of Alcalá, 28805 Alcalá de Henares, Madrid, Spain
[b] Department of Computer and Systems Engineering, Higher School of Engineering and Technology, University of La Laguna, 38200 San Cristobal de La Laguna, S/C de Tenerife, Spain
[c] Department of Analytical Chemistry, Physical Chemistry and Chemical Engineering, Faculty of Sciences, University of Alcalá, 28805 Alcalá de Henares, Madrid, Spain

ARTICLE INFO

Keywords:
Gene expression cancer
Feature selection
Multi-classification
Grouping genetic algorithm
Extreme learning machine

ABSTRACT

This paper presents a Grouping Genetic Algorithm (GGA) to solve a maximally diverse grouping problem. It has been applied for the classification of an unbalanced database of 801 samples of gene expression RNA-Seq data in 5 types of cancer. The samples are composed by 20,531 genes. GGA extracts several groups of genes that achieve high accuracy in multiple classification. Accuracy has been evaluated by an Extreme Learning Machine algorithm and was found to be slightly higher in balanced databases than in unbalanced ones. The final classification decision has been made through a weighted majority vote system between the groups of features. The proposed algorithm finally selects 49 genes to classify samples with an average accuracy of 98.81% and a standard deviation of 0.0174.

1. Introduction

The field of biotechnology has currently developed techniques such as microarrays or RNA-Seq suitable to record data of gene expression in tissue samples. The classification of cancer using the genetic profiles obtained in the sequencing allows to discriminate between healthy and sick individuals or between various types and subtypes of cancer [1,2]. These results facilitate the diagnosis, adaptation and improvements in the treatments of patients [3,4].

One of the most complex problems in the classification of cancer is the well-known Curse of Dimensionality or Hughes Effect [5,6], which states that the increasing the dimensionality decreases the reliability of the estimation of the statistical parameters required to calculate the probabilities. The DNA sequences include tens of thousands of genes against a relatively low number of samples. In addition, the DNA sequences contain a very high number of genes that are irrelevant for target cancer types because they do not influence the classification. This causes the performance of the classification algorithms to be dramatically reduced [7,8].

For this reason, numerous studies have been developed focusing on the selection of a small group of genes that are significant for classification. The problem of the identifying genetic markers of the disease has been solved by many authors using feature selection techniques in machine learning algorithms [3,7–12]. Filter selection methods establish rankings in the features set to obtain the ones that are the most effective for classification. Pavithra et al. [10] set the filters according to the mutual information, Guyon et al. [11] ordered them according to the weights of a recursive Support Vector Machine (SVM) trained in the classification. On the other hand, the selection criterion in the wrapping methods is based on the performance of the classifier. These techniques are more computationally intensive, but they obtain more effective results in the selection of genes [8]. Therefore, it is important to improve the algorithms that solve the problem of extracting features without a large computational requirement. According to the wrapping method, there are different techniques to explore in search space, finding groups of characteristics or features that work as efficient classifiers [6–13]. Diverse techniques have been used to solve a maximally diverse grouping problem, such as Genetic Algorithms (GA), SVM [14–16], or K-Nearest Neighbors [16], Local Search [17], Tabu Search Approach [18], Particle Swarm Optimization (PSO) or Artificial Bee Colony (ABC) [15]. Zhu et al. [15] compared GA with hybrid algorithms such as GA-SVM, PSO-SVM and ABC-SVM, concluding that GA is more effective for extracting features from the original data: “Therefore, when compared with the ABC and PSO algorithms, the GA had more advantages in terms of feature band selection, small sample size classification, and classification accuracy.” GA also performed better

* Corresponding author.
E-mail addresses: pilar.garcia@uah.es (P. García-Díaz), isanchez@ull.edu.es (I. Sánchez-Berriel), juanan.martinez@uah.es (J.A. Martínez-Rojas), am.diez@uah.es (A.M. Diez-Pascual).

https://doi.org/10.1016/j.ygeno.2019.11.004
Received 24 July 2019; Received in revised form 4 October 2019; Accepted 11 November 2019
Available online 20 November 2019

than PSO and ABC when they hybridized with SVM. In their experiments, GA-SVM had the highest average accuracy (91.77%) and the best stability (the standard deviation of its classification accuracy was 0.82%). Singh et al. [19] also obtained better results for GA than for Local Search [17] and Tabu Search [18].

GA have also been applied for the optimal selection of gene sets in the cancer classification through microarrays or RNA-Seq [1,10,20,21]. Several datasets of gene expression cancer RNA-Seq have been published to facilitate cancer classification research: for example, the public databases available on the GEO platform (www.ncbi.nlm.nih.gov/geo/) or in the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/index.php). Recent literature has diverse studies with these databases [21–23]. Zhang et al. [21] applied an SVM algorithm for the selection of features and classification of peripheral blood data with accession number GSE16443 in GEO platform. This dataset is composed of 130 samples of the gene expression cancer RNA-Seq that belong to two categories: breast cancer samples and healthy ones. The authors obtained an accuracy of 81.54% applying training and testing datasets of equal size. Salman et al. [22] works with data from The Cancer Genome Atlas Pan-cancer Analysis Project to apply a Hybridized Genetic Algorithm with an Artificial Neural Network with 4 hidden layers (ANN + GA) without feature selection. The GA algorithm adjusts the weights of the neural network according to the influence of the all characteristics.

Our proposed algorithm consists of a Genetic Grouping Algorithm (GGA) applied to the gene selection from gene expression RNA-Seq data for multi-classification of cancer. The GGA integrates an Extreme Learning Machine (ELM) algorithm into the fitness function. The ELM has demonstrated a high computational speed (generating non-iterative solutions) and a very good performance in binary and multiclass classification [24–28]. Since the evolutionary algorithms offer solutions close to the optimum, a voting system among the best classifiers found by the GGA is proposed. The voting system has been used successfully in various areas such as in the diagnosis of diseases, in Natural Language Processing (NLP) or in the treatment and classification of images. Bashir et al. [29] used a weighted vote ensemble integrating several heterogeneous classifiers for the diagnosis of breast cancer. In the field of NLP, Ekbal et al. [30] incorporated the voting decisions in the encoding operation of the GA for Named Entity Recognition (NER); Ankit et al. [31] and Onan et al. [32] performed sentiment analysis in texts also using voting systems. García-Gutiérrez et al. [33] implemented an evolutionary-weighted majority voting strategy and an SVM for contextual classification of LiDAR and imagery data fusion.

The paper is structured as follows: Section 2 explains the theoretical concepts of the algorithms GGA and ELM. Because the GGA is an adaptation of the GA for the resolution of grouping problems, the GA is briefly explained. Section 3 describes the dataset used to carry out the experiments. Two sets of data are composed according to the number of samples for each type of cancer. Section 4 presents the results obtained for the two sets of data analyzed. Section 5 offers the discussion of the results and, finally, the conclusions of the research are included in Section 6.

2. Algorithm for classification problem

The problem of classifying a set of samples of gene expression cancer RNA-Seq belongs to the category of grouping problems. The proposed algorithm to carry out the classification is the GGA, which is an adaptation of the GA for the resolution of grouping problems. The fundamental parts of the GA are described in the following subsections: encoding of the solutions, fitness function, parental selection, recombination, mutation and stop conditions. It is extremely important for the performance and efficiency of the algorithm to correctly define the encoding and the fitness function of the algorithm. The transformation of a GA into a GGA requires the encoding to be conveniently established. The fitness function implemented in the GGA is the classification accuracy calculated through the Extreme Learning Machine (ELM). The effectiveness and efficiency of the ELM have already been demonstrated in the literature [34–42]. The ELM is described in the third subsection.

2.1. The Genetic Algorithm as evolutionary algorithm

The Genetic Algorithm (GA) is one of the best-known evolutionary algorithms in the area of optimization. As stated in [34], optimization techniques are generally applied to solve problems in which it is extremely difficult to find the optimal solution due to the existence of opposing criteria. In these cases, it is quite tedious to obtain the optimal solution and, in general, it is not possible to distinguish whether there is a single optimal solution or several solutions and how many solutions are close enough to the optimum and they are much easier to be found than the optimum one. Then, the objective is focused on finding a solution sufficiently near to the optimum, with a limited execution time.

The GA is bio-inspired in the theory of evolution by natural selection proposed by Charles Darwin. A set of solutions form a population modeled for the optimization problem. Each solution corresponds to an individual who fight to survive in the ecosystem. At the beginning it is not important how good the population fitness is, in fact, the initial solutions are randomly created and most of them are very far from the optimal solution. Individuals with better fitness value are more likely to survive than individuals with worse fitness. The survival of individuals depends not only on their fitness, but also on their lot in life. The GA finds better solutions to the optimization problem after many generations of evolution. After a sufficient number of generations, the GA can find solutions enough close to the optimum.

Fig. 1 shows the general flow chart of the GA, where a complete cycle represents a generation of the evolutionary process. During a generation, several operations are carried out: evaluation of the fitness, selection of individuals in the mate choice, recombination or crossover, mutation and selection of survivors for the next generation. The application of GA to a specific type of problem determines the characteristics and parameters of the algorithm such as: encoding of solutions, parental selection, recombination and mutation. The algorithm begins with an initial population. Each solution is characterized by a fitness value, which is a measure of how well it solves the optimization problem. After the initial evaluation the algorithm generates new solutions as offspring of the current population. The method to create a new solution is generally based on the content of two existing solutions, bio-inspired by the creation of offspring from the genetic makeup of their parents. The GA models the mutation as a random modification in a part of the solution code. After recombination and mutation, the fitness value of the offspring is calculated.

The offspring is added to the current population. As the size of the total population increases and resources are restricted, it is necessary to apply a population control process that selects the surviving individuals, discarding the rest. There are different discard methods applied to GA, the tournament selection operator has provided very good results in previous applications [43,44]. The round of tournaments is carried out with the merged population formed by parents and offspring, based on the best fitness values of the fighters. The foe is chosen at random for each tournament. This technique allows that the solutions with better fitness state generally win the tournaments, but it also depends on the chance when selecting the opponent.

The algorithm continues generation after generation until any of the defined stop conditions are met. Stop conditions for GA usually consist of a maximum number of generations or reaching population convergence [45,46]. The convergence of the algorithm occurs when it does not progress for successive generations. This means that the best fitness value does not change for a certain number of generations. Another definition of convergence is when all individuals in a population have very similar fitness. With a population like this, it will be very unlikely to find a new individual with better fitness in the next

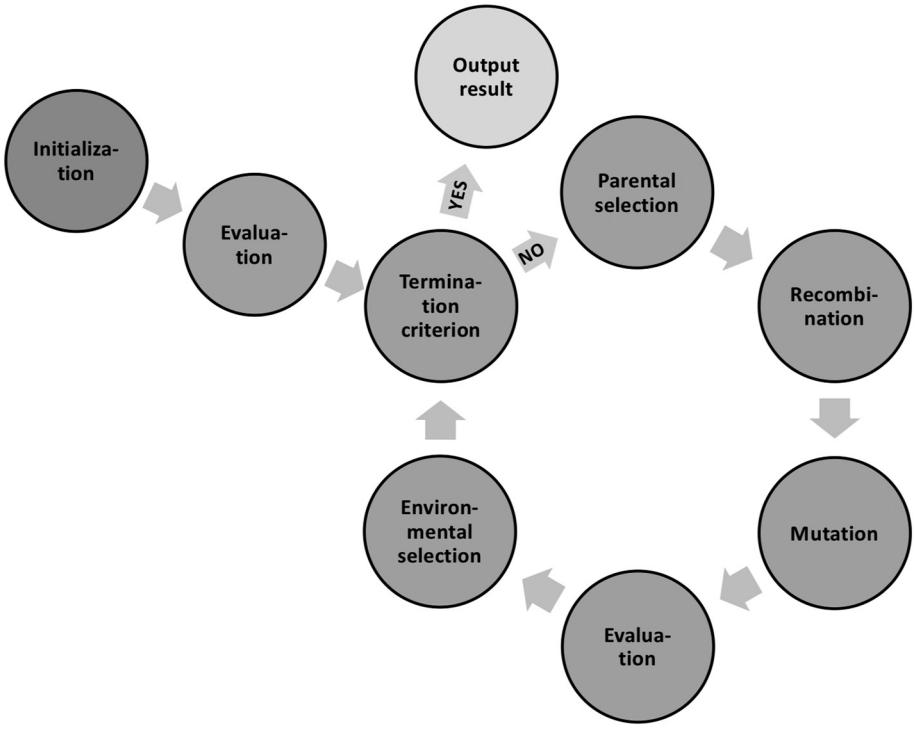

Fig. 1. General flow diagram of the Genetic Algorithm (GA). A population or group of individuals, representing solutions to the problem to be solved, advances generation after generation, becoming at each stage individuals better adapted to the environment.

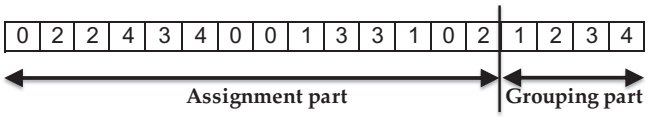

Fig. 2. Coding of a solution of grouping of 14 elements. Each element reserves a cell in the assignment part. The content of this sub-array indicates the group to which each element belongs. In the example, 4 groups are established, and each element is assigned to a single group. Items not yet classified keep a zero in the assignment array.

generations. In this case, convergence is calculated by comparing the fitness of the best current individual f_{best} with the average fitness of the population $f_{average}$, as shown in Eq. (1). This paper applies two stopping conditions: a maximum number of generations and the convergence caused by a population with very similar fitness, with an epsilon of 0.001. This epsilon value is so small that in practice it is a control measure to detect clone populations. When the diversity of the population is guaranteed, the existence of few clones, the real stop condition will be a maximum number of generations processed.

$$f_{average} - \epsilon \leq f_{best} \leq f_{average} + \epsilon \quad (1)$$

When the algorithm is stopped, the GA offers the set of individuals with the best fitness found, from which one or several can be selected, as appropriate. These individuals encode the best solutions found for the proposed problem.

2.2. The Grouping Genetic Algorithm

The Grouping Genetic Algorithm (GGA) is a modification of the GA for solving clustering and grouping problems [34,47–50]. The word 'grouping' refers to a technique that takes advantage of special encoding strategies to obtain compact hierarchical arrangements with a high performance in terms of a hierarchy-dependent metric in grouping-based problems [51].

2.2.1. Encoding of solutions

GGA solutions are encoded as an array of elements composed of two sub-arrays: assignment part and grouping part. Both sections are arrays of natural numbers. The length of the assignment part coincides with the number of elements to be classified. The length of the grouping part is the number of groups established for the encoded solution. The values stored in the grouping part are the identifiers of the groups, while the value stored in each cell of the assignment part is the group identifier to which the element to be classified is associated. Note that the information about the classification is in the content of the encoding and also in the length of the grouping part. For this reason, individuals have variable length, since each solution can consider a different number of groups.

Fig. 2 shows the encoding of a grouping solution of 14 elements (the length of the assignment part is 14). The individual considers four groups named from 1 to 4 (the length of the grouping part is 4). The identifiers of the groups must be consecutive natural numbers starting at 1. This is a requirement to make easier the crossover operation in the algorithm. The content of the assignment part is the set of associations of each element to a single group. If an element is not yet classified, the array stores a zero value in the cell. The first element in Fig. 2 has not been yet associated to any group, whereas the second, third, and last elements of the assignment part are connected with group 2. The elements in the fourth and sixth positions are associated with class 4, and so on.

2.2.2. Recombination and mutation operators

The GGA also respects the general flow process described in Fig. 1 for the GA. However, the recombination and mutation operations have different characteristics from the corresponding operations in the GA. As far as the mutation of an individual is concerned, this operation can be understood as a recombination with another random individual. The GGA crossover operation is a two-point crossover that creates a single offspring from every two parents. Fig. 3 shows an example of recombination to generate a new individual from the two parents: Individual A and Individual B. The new individual is initially a copy of A,

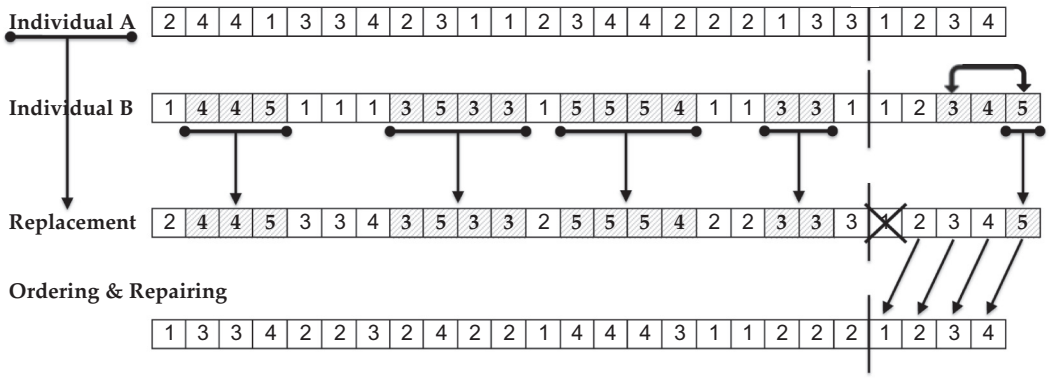

Fig. 3. Two-point crossover example for GGA. A new individual is created with combined information from their parents. A clone of the Individual A is modified with some random groups of the Individual B and the corresponding classification stored in the assignment part of B. Finally, group identifiers are renamed to be natural numbers starting at 1.

in which different actions are performed. Two-points of the grouping part of the individual B are randomly selected. In the example on Fig. 3, the interval [3–5] has been picked. This selection points to the information that will be transmitted from B to the offspring in the replacement phase. On the one hand, the descendant will have all the groups of the interval [3–5], adding the groups that do not exist in A. Therefore, a fifth group is added to the grouping part of the offspring. On the other hand, all assignments to the groups [3–5] present in B are copied literally in the descendant. These assignments are marked in Fig. 3 with blue horizontal lines. Note that the descendant keeps a group with the identifier 1 but there is no element associated with this group in the assignment array. This empty group will disappear in the next phase of the crossover operation.

The last stage of recombination is the repair and renaming of groups in the new individual. This phase guarantees that all the elements in the offspring are classified among the groups (there are not empty groups) and that the identifiers of the groups are natural numbers starting at 1. The process does not modify the information of the solution, only its format. The names of the groups are changed and, therefore, the content of the array is also modified: where group 2 takes identifier 1, group 3 takes identifier 2 and so forth.

2.2.3. Fitness function

The algorithm must solve a selection wrapping features that minimizes the output error [13,52]. The fitness function uses the Extreme Learning Machine (ELM) algorithm to calculate the accuracy of the classification according to a data selection. Minimizing the output error is understood as equivalent to maximizing the accuracy of the classification data. ELM has the two main properties to be an effective fitness function [34–42]: the regressor is accurate enough and the evaluation process is extremely fast, as its name implies.

All C_i individuals must be evaluated, with $i \in \{1, \ldots, N_{ind}\}$, where N_{ind} is the population size. Fig. 4 shows an example of how the fitness of an individual C_i is evaluated. C_i is composed of $n_i = 4$ groups of features, named as $j \in \{1, \ldots, n_i\}$. The features of each group G_{ij} are stored in the assignment part of the individual. Table 1 registers the content of each group G_{ij} in the 'Features' column. ELM is applied in each group G_{ij} to calculate its accuracy of classification (A_{ij}) in the testing data as Eq. (2).

Table 1
Fitness function for the individual C_i of Fig. 4. The column 'Features' indicates the composition of the $n_j = 4$ groups of C_i. For each group G_{ij}, the accuracy of the classification A_{ij} is calculated according to it selection of features. The fitness of C_i is the maximum A_{ij}. In the example, the selection of winning features is defined by the group G_{i2} that has the maximum accuracy value ($A_{i2} = 0.81$).

j	Features of group G_{ij}	Accuracy A_{ij} $f_{ELM}(G_{ij})$ in testing data
1	{1, 12, 17, 18}	0.48
2	**{5, 6, 8, 10, 11, 19, 20, 21}**	**0.81**
3	{2, 3, 7, 16}	0.57
4	{4, 9, 13, 14, 15}	0.22
		$\mathscr{F}(C_i) = \max(A_{ij}) = 0.81$

The bold text highlights the group with the maximum accuracy value (Ai2 = 0.81).

$$f_{ELM}(G_{ij}) = A_{ij};\ i \in \{1,\ldots,N_{ind}\};\ j \in \{1,\ldots,n_i\} \quad (2)$$

The accuracy of the classification A_{ij} depends on the number of samples correctly classified. It is calculated by the Eq. (3), where TP means the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN means the number of false negatives:

$$A_{ij} = \frac{TP + TN}{TP + TN + FP + FN}\ (\%) \quad (3)$$

Eq. (4) defines the fitness value of the individual C_i as the maximum of all A_{ij}. The group with $j = 2$ in Fig. 4 has the maximum fitness value of all groups ($A_{i2} = 0.81$). This group determines the selection of features found {5, 6, 8, 10, 11, 19, 20, 21} that classify the data with greater accuracy. The rest of the groups have no interest and their A_{ij} values are discarded, since they define other feature groups with worse accuracies.

$$\mathscr{F}(C_i) = \max_{j \in \{1,\ldots,n_i\}} (A_{ij});\ i \in \{1,\ldots,N_{ind}\} \quad (4)$$

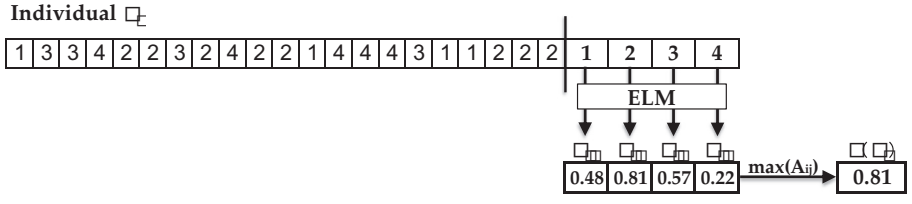

Fig. 4. Fitness function for the individual C_i in the GGA. ELM is applied to each group G_{ij} obtaining its classification accuracy A_{ij} in the testing data. The fitness value of C_i is the maximum value of them.

Like the GA, the GGA evolves generation after generation until either of the two stop conditions is satisfied: a maximum number of generations or the convergence of the population. After the end of the execution, the best groups of features found by the GGA are recorded as classifiers.

This work combines a subset of the best classifiers found by the GGA in a majority voting system to increase the accuracy of the classification. The accuracy of the voting system is greater than the accuracy of any classifier that operates individually. The classifiers and their weights used in the voting system are set experimentally. Not all combinations of good classifiers increase the accuracy by a similar proportion. Section 4 (Experimental Results) presents the accuracy of three voting systems, which combine from 5 to 10 classifiers that operate individually and then the final decision is made according to the majority of the votes.

2.3. The Extreme Learning Machine algorithm

The Extreme Learning Machine or ELM fulfills the two mentioned properties of an effective fitness function: it is a machine learning algorithm that achieves a very good generalization performance with an extremely fast speed. The ELM algorithm was initially proposed by Huang et al. [24,25,27,28,53–55]. ELM performs the selection of the weights of the hidden neurons of a Single-hidden Layer Feedforward Neural Network (SLFN). ELM has shown good performance in large multi-label dataset classification applications, regression applications and dimensional reduction problems [35–42,56,57]. To give a brief description of how the ELM works, the authors use the same notation as [55]. The authors do not consider necessary to include here the same figure of [55], that complements the explanation. Consider a SLFN with d input nodes and m output nodes. The network is trained with N data such as:

$$(\boldsymbol{x}_j, \boldsymbol{t}_j) \in \mathbb{R}^d \times \mathbb{R}^m,\ j = 1,\ ...,N \tag{5}$$

where $\boldsymbol{x}_j$ is the data input vector j^{th}, $\boldsymbol{t}_j$ is the corresponding class of $\boldsymbol{x}_j$. The single hidden layer is composed of L nodes, each having an activation function $G(\boldsymbol{a}_j, b_j, \boldsymbol{x})$ where $\boldsymbol{a}_j$ is the associated connection weight vector and b_j is the bias. The $\boldsymbol{a}_j$ and b_j parameters are assigned randomly. The hidden layer output matrix $\boldsymbol{H}$ is defined as:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{x}_N) \end{bmatrix} = \begin{bmatrix} G(\boldsymbol{a}_1, b_1, \boldsymbol{x}_1) & \dots & G(\boldsymbol{a}_L, b_L, \boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ G(\boldsymbol{a}_1, b_1, \boldsymbol{x}_N) & \dots & G(\boldsymbol{a}_L, b_L, \boldsymbol{x}_N) \end{bmatrix}_{NxL} \tag{6}$$

Defined $\boldsymbol{\beta}$ as the weight vector that connects the hidden layer with output layer, the algorithm must calculate the weight vector $\boldsymbol{\beta}$ that minimizes the least squares error. The ELM output is the optimal weight vector $\boldsymbol{\beta}^{\sim}$ of the network:

$$\boldsymbol{\beta}^{\sim} = \boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{7}$$

where $\boldsymbol{H}^{\dagger}$ stands for the Moore-Penrose inverse of matrix $\boldsymbol{H}$ and $\boldsymbol{T}$ corresponds to the class labels of the data:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1^T \\ \vdots \\ \boldsymbol{t}_N^T \end{bmatrix}_{Nxm} \tag{8}$$

3. Data

The proposed algorithm has been proved with a gene expression cancer RNA-Seq data set used in recent literature [22]. The gene expression cancer RNA-Seq dataset is downloaded from UCI Repository [58]. This dataset is part of The Cancer Genome Atlas Pan-cancer Analysis Project [59,60]. The original data set is maintained by the cancer genome atlas pan-cancer analysis project.

As stated by Zhang et al. [21], "Gene expression data contains DNA microarray data and RNA-seq data. Analysis of microarray data helps clarify biological mechanisms and push drugs toward a more predictable future. Compared to hybridization-based microarray technology, RNA-seq has a larger range of expression levels, and more information is detected."

RNA-Seq is a random extraction of gene expression of patients with five different types of tumors in the dataset: BRCA (breast), KIRC (kidney), COAD (colon), LUAD (lung) and PRAD (prostate) [61]. Table 2 collects the number of samples used for each type of tumor. The size of data is 801 samples, each of which is defined by 20,531 features or genes. The genes are identified with labels from gene_0 to gene_20530.

Table 2 indicates that the data set is not balanced because it has a different number of samples from each type: the largest group is BRCA with 300 samples, while the smallest is COAD with 78 samples. In order to use a more balanced subset of data (stratified random sampling) [6], a balanced set of data with the five types of tumors was composed. For such purpose, the size of the five subsets was fixed to the 78 elements as the COAD group. The samples are randomly selected until the amount established for each group is completed. Table 3 shows the number of samples of the new data set consisting of 5 classes with the same number of elements.

Table 2
Division of DNA microarray data of UCI Repository, used in Section 4.1.

Class	Class ID	Number of samples
BRCA	1	300
PRAD	2	136
KIRC	3	146
LUAD	4	141
COAD	5	78
	Total	801

4. Experimental results

The GGA was applied to the dataset presented in the previous section to extract a selection of characteristics or genes (to solve a wrapper feature selection) that determined the most influential genes and their combinations in order to classify the samples with good accuracy: according to the types of tumors in the RNA-Seq dataset from UCI Repository.

The complete data set is divided into three disjoint sets: training, testing and validation. The training and test datasets are used in the training and test stages during the execution of the algorithm. During this time, the validation data is hidden for the algorithm. Once a stop condition is reached, the best solutions found are evaluated on the validation data set. The sizes commonly used for these sets are 80% of the data for the training and 20% for testing whether a validation set is not created. In our experiments a validation set has been used because it provides more veracity: 80% for training, 10% for testing and 10% for validation. The results do not lose generality because 10% of the total is large enough: 80 samples in case 1 and 39 samples in case 2.

The number of generations, the population size and the probability

Table 3
Stratified sampling of DNA microarray data from the UCI Repository, used in Section 4.2.

Class	Class ID	Number of samples
BRCA	1	78
PRAD	2	78
KIRC	3	78
LUAD	4	78
COAD	5	78
	Total	390

of mutation are established experimentally because the adaptation of the algorithm to the type of problem to be solved influences these values. A small number of simulations with different values of these parameters were executed, identifying the value ranges for which the GGA obtains a better fitness.

Both the number of generations and the population size influence mainly the convergence time [62], since with a greater variety of solutions it is more likely to find some with good aptitude and an acceptable time. The number of generations used in the literature can vary from 25 [63] to 2000 [64,65] or 1,000,000 [66]. In our experiments, the algorithm was initially executed with up to 100 generations, and it was found that from 60 generations there was hardly improvement in the suitability of the solutions.

The population size usually used in the literature varies between 20 and 100 [64,65]. The probability of mutation used in different applications ranges from 0.01 [41] to 0.1 [63,65].

The size of the offspring depends on the implementation of the parent selection function and the crossover function. The results described were obtained by allowing all individuals to match only once in each generation and each couple to generate a single offspring.

The number of simulations is also determined by experimentation because the adaptation of the algorithm to each type of problem influences its value. Researches carried out with simulations in the range of 1–50 can be found in the literature [64].

The use of an ELM as a classifier establishes by definition that the number of neurons matches the number of classes in the data set. The experiments described herein contain samples belonging to 5 types of tumors (BRCA, PRAD, KIRC, LUAD and COAD), therefore the ELM used is composed of 5 neurons.

The GGA was executed several times with independent simulations. Each simulation was run with its own randomly constructed data sets (training, testing and validation). The GGA run 3 simulations in the UCI data set to obtain excellent results. The parameter values chosen for the experiments are summarized below:

- Training data size = 80%
- Testing data size = 10%
- Validation data size = 10%
- Number of generations (G_{max}) = 60
- Population size (N_{ind}) = 50
- Offspring size = 25
- Mutation probability = .1
- Number of neurons in ELM = 5
- Number of simulations = 3

All experiments were performed with a 2.7 GHz Intel Core i7 processor. The results presented below are the average values obtained from 500 classification iterations using the ELM on the set of validation data, reserved for this phase. The validation data is unknown for the algorithm until this moment of the process. The 500 iterations are random and independent of each other.

The classification of the samples according to the totality of the features offers very poor results. Consequently, the GGA was applied to extract a limited set of features such as a classifier. Taking into account that the set of genes is very large (20,531 genes), the operation also spends too much computing time. More important is that the operation of classifying cancerous samples according to the total of 20,531 genes should make sense, since not all the genes are influential in the determination of a tumor. On the other hand, experts indicate that the combination of certain genes can be decisive for the appearance of specific tumors. In order to prove this fact in an algorithmically manner, the ELM fitness function is executed with all characteristics of the data indicated in Table 2, that is, without features selection, to classify the samples of a test dataset according to the 5 types of tumors: BRCA, PRAD, KIRC, LUAD and COAD. As expected, the average accuracy in the classification was unacceptable, being 37.37% with a standard deviation of 0.0518. This fact shows that it is necessary to extract a group of characteristics as classifier. The selection of these characteristics or genes is carried out by the GGA.

Salman et al. applied a genetic algorithm hybridized with an artificial neural network with 4 hidden layers (ANN + GA) to classify the RNA-Seq data, obtaining an average accuracy of 98.75 with a standard error of difference of 0.001 [22]. The GA algorithm adjusts the weights of the neural network according to the influence of the all characteristics. The big difference in the accuracies obtained by Salman et al. [22] and with the ELM without feature selection is due to the fact that the execution of the ELM is applied only once whereas [22] consists of an evolutionary algorithm that improves the last solution in each iteration. The improvement in classification accuracy will be obtained in our experiments after applying the GGA evolutionary algorithm.

The experiments were carried out with different subsets of gene expression cancer RNA-Seq dataset from UCI Repository:

- Case 1: the complete dataset from UCI Repository composed of 801 samples belonging to 5 types of tumors, see Table 2.
- Case 2: a subset with stratified sampling of 5 types and composed of 390 samples, summarized in Table 3.

4.1. Results of the classification in case 1 with the whole dataset of Table 2

The GGA selects the solutions with the best fitness found (accuracy classification). Each solution is composed of several features/genes among the 20,531 possible of the data set. Table 4 shows the average accuracy in the classification of the RNA-Seq in the UCI data using each of these different 5 classifiers (averaged over 500 independent and random iterations). This accuracy is > 92% in all 5 cases, with a standard deviation of 0.01. If the five classifiers are applied in a voting system, the average accuracy increases to 0.9812, with a standard deviation of 0.0148. The voting system used is a majority voting system with equal weights, which means that the weight of each of the N classifiers is $\omega_i = 1/N$ as defined in Eq. (9) with $N = 5$.

$$\sum_{i=1}^{N} \omega_i = 1,\ i = 1,\ ...,N \Rightarrow \omega_i = \frac{1}{N},\ i = 1,\ ...,N \tag{9}$$

The final decision is made based on the majority vote. These data are collected in the last row of Table 4. The GGA complemented with a voting system only needs 20 genes out of the 20,531 genes in the

Table 4
Composition of the 5 groups of genes selected by the GGA. The average accuracy in the classification of the RNA-Seq in the UCI data set is indicated for each group to the right of the table. The last row indicates the average accuracy in the classification with a majority voting system.

Group of genes	Feature ID	Average accuracy	Standard deviation
1	{gene_15900 gene_18636 gene_18746 gene_18810}	0.9409	0.0383
2	{gene_8032 gene_10428 gene_11250 gene_11550}	0.9327	0.0369
3	{gene_1216 gene_9176 gene_15089 gene_17770}	0.9285	0.0396
4	{gene_2352 gene_9652 gene_10290 gene_15900}	0.9236	0.0376
5	{gene_8616 gene_12986 gene_15899 gene_17906}	0.9175	0.0358
Majority voting system with the 5 groups		0.9812	0.0148

Table 5
Composition of the 8 groups of genes selected by the GGA. The average accuracy in the classification of the RNA-Seq in the UCI data set is indicated for each group to the right of the table. The last row indicates the average accuracy with a weighted majority voting system.

Group of genes	Feature ID	Average accuracy	Standard deviation
1*	{gene_15900 gene_18636 gene_18746 gene_18810}	0.9409	0.0383
2*	{gene_8032 gene_10428 gene_11250 gene_11550}	0.9327	0.0369
3*	{gene_1216 gene_9176 gene_15089 gene_17770}	0.9285	0.0396
4	{gene_2352 gene_9652 gene_10290 gene_15900}	0.9236	0.0376
5	{gene_8616 gene_12986 gene_15899 gene_17906}	0.9175	0.0358
6	{gene_9626 gene_11352 gene_14092 gene_19151}	0.8620	0.0386
7	{gene_9177 gene_17316 gene_18746}	0.8830	0.0370
8	{gene_15895 gene_16006 gene_17075 gene_18391}	0.8307	0.0434
Majority voting system with the 8 groups (*double weight)		0.9821	0.0156

database.

Table 5 shows 8 of the solutions generated by the GGA, with a total of 31 genes. The rows in the table show the average accuracy of each classifier when they act independently. The accuracy of the classification improves when a voting system is applied. The voting system assigns the first 3 classifiers a double weight in the voting to compensate for the fitness differences of the 8 classifiers (94.09% the first classifier against 83.07% of the last classifier). This distribution of weights in the vote is expressed in Eq. (10). In the first column of Table 5, asterisks identify classifiers with double weight in the voting system.

$$\sum_{i=1}^{N} \omega_i = 1,\ i = 1,...,N$$
$$\omega_i = 2x,\ i = 1,\ 2,\ 3$$
$$\omega_i = x,\ i = 4,...,8 \tag{10}$$

With this voting system an average accuracy of 98.21% and a standard deviation of 0.0156 is reached. The GGA method complemented with a voting system uses much fewer genes (31 of 20,531 total). It is much more direct than the classifier of [22], though both lead to very good accuracy: 98.21% for GGA + voting and 98.75% for ANN + GA.

The classifier defined as voting system of different weights 8 classifiers of Table 5 is evaluated through 500 independent random iterations *via* classification of the data collected in Table 2. The size of training and testing sets are 90% and 10%, respectively. Fig. 5 graphically represents one of the worst results obtained in all iterations. The figure marks the successes in classification that matches the red circle (fact) with the blue circle (prediction). The errors locate the blue circle on a tumor type different from the correct type in the same vertical. The errors are marked with vertical black lines that indicate the difference between the wrong prediction and the correct type. The 6 errors produced in 80 classified samples implicate a 92.5% accuracy classification, being this one of the worst cases of the 500 iterations.

4.2. Results of the classification in case 2 with stratified sampling of the dataset as Table 3

The GGA is applied in a balanced subset of the same database [6], randomly discarding samples of the four cancer types with the largest number of samples until leaving 78 samples of each type (see Table 3). The balanced subset is composed of 390 samples with the same number of characteristics (20,531 genes).

Table 6 collects 10 of the best solutions found by the GGA algorithm to classify a set of UCI Repository data with the stratified sampling referred to in Table 3. Each group works as an independent classifier. The size of the groups varies between 4 features/genes (in groups 3, 4, 5, 7, 8 and 10 of the Table 6) and 9 features/genes (in the group 9). The average accuracy and standard deviation have been calculated by performing 500 random iterations. Table 6 shows the groups of genes ordered from highest to lowest average accuracy.

The last row of the table shows the average accuracy (98.81%) and standard deviation (0.0174) in the classification by a voting system by majority of the 10 classifiers. The first three classifiers have a greater weight in the voting than the rest, given that they have the best average accuracy (> 91%). The voting system was evaluated with several weighting distributions in the classifiers. The combination described in Eq. (11) achieved the best results: the first classifier has a double vote and the next two have triple weight compared to the rest. Table 6 labels with a single asterisk the classifier with double weight and with double asterisk those that have a triple weight in the voting system.

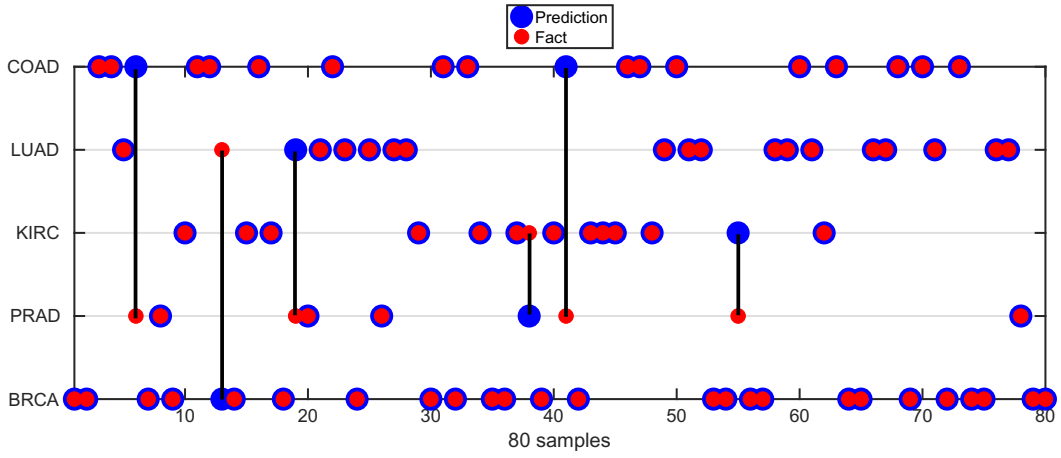

Fig. 5. The worst case of classification (accuracy of 92.50%) of the data collected in Table 2. Classification through a voting system of 8 classifiers of Table 5. The x-axis corresponds to the validation dataset (80 samples). The y-axis collects the 5 types of cancer: BRCA (breast), KIRC (kidney), COAD (colon), LUAD (lung) and PRAD (prostate).

Table 6
Composition of the 10 groups of genes selected by the GGA. The average accuracy in the classification of the RNA-Seq in the UCI data set is indicated for each group to the right of the table. The last row indicates the average accuracy with a weighted majority voting system.

Group of genes	Feature ID	# Genes	Average accuracy	Standard deviation
1*	{gene_5021 gene_6594 gene_8004 gene_8005 gene_16299}	5	0.9232	0.0552
2**	{gene_1795 gene_4447 gene_6399 gene_12995 gene_15897}	5	0.9206	0.0534
3**	{gene_9544 gene_11124 gene_15897 gene_19162}	4	0.9130	0.0539
4	{gene_6816 gene_7805 gene_16991 gene_18354}	4	0.8619	0.0622
5	{gene_9626 gene_11352 gene_14092 gene_19151}	4	0.8426	0.0640
6	{gene_7420 gene_8004 gene_8943 gene_16299 gene_20260 gene_20360}	6	0.8388	0.0172
7	{gene_1122 gene_6611 gene_10844 gene_11094}	4	0.8364	0.0755
8	{gene_220 gene_364 gene_9652 gene_16094}	4	0.7540	0.0734
9	{gene_659 gene_5377 gene_8868 gene_10111 gene_12808 gene_15894 gene_17184 gene_19553 gene_19760}	9	0.7468	0.0842
10	{gene_15895 gene_16006 gene_17075 gene_18391}	4	0.7323	0.0702
Majority voting system with the 10 groups (*double weight, **triple weight)		49	0.9881	0.0174

$$\sum_{i=1}^{N} \omega_i = 1,\ i = 1,...,N$$
$$\omega_1 = 2x,\ \omega_i = 3x,\ i = 2,\ 3$$
$$\omega_i = x,\ i = 4,...,10 \tag{11}$$

To compare the GGA + voting classifier with that proposed by Salman et al. [22], the classifier was applied to the 411 discarded data from the UCI Repository when obtaining the 390 samples from Table 3. The 411 samples are unknown data for the ELM training. The average accuracy on 500 random iterations offers very similar results to the last row of Table 6: average accuracy of 98.82% and standard deviation of 0.0046. The voting system offers a better performance (98.82%) in the classification of RNA-Seq in the UCI dataset than the ANN + GA algorithm with 98.75% of [22]. In addition, GGA + voting system works with only 49 genes against the total of 20,531 genes in [22].

Fig. 6 shows the worst case obtained in the 500 iterations, which had an accuracy of 97.08%. Only 12 samples from 411 were wrongly classified as another type of tumor. The figure highlights these cases with vertical black lines where red circles represent the correct type of tumor and the blue circles indicate the type of tumor misclassified.

5. Discussion

We have applied the GGA algorithm to the database from the UCI Repository previously used in [22] for the classification of DNA samples according to different types of cancer. The database taken from the UCI Repository collects a total of 801 samples belonging to 5 types of cancer: breast, kidney, colon, lung and prostate (Table 2). This database is not balanced because the number of samples of each type is quite different: the most numerous samples are those of breast cancer (300 in total) and the least frequent are those of the colon cancer (78 samples in total). With this unbalanced database, several groups of genes are obtained that allow a classification with acceptable accuracy. The groups act as individual classifiers whose decision is combined into a voting system. All numerical values presented in the results of the experiments have been averaged running 500 random and independent iterations.

Table 4 identifies 5 classifiers each formed by 4 genes. Each individual classifier has an average accuracy > 91%. The set of 20 characteristics, combined as 5 classifiers in a voting system 'one vote-one value', obtain an average accuracy in the classification of 98.12% and standard deviation of 0.0148.

Table 5 shows a solution with better result than the previous one. The solution consists of a total of 31 genes grouped into 8 groups or classifiers and working under a voting system with different counting values. The three classifiers with the highest average accuracy are counted with greater value than the rest of votes. This solution classifies with an average accuracy of 98.21% and standard deviation of 0.0156. The worst case found in the 500 random iterations reaches a success of 92.50% (Fig. 5). This result proves the validity of the proposed algorithm, in which the worst case misclassifies only 6 samples of 80 samples.

The GGA algorithm has also been executed in a balanced subset of the same database, randomly discarding samples of the four cancer types with the largest number of samples until leaving 78 samples of each type (Table 3). Discarded samples are added to the validation dataset with which the performance of the algorithm is evaluated. The GGA algorithm offers other combinations of genes as an accurate classifier. Table 6 shows a set of 49 genes grouped into 10 classifiers. The voting system with different counting values improves the average accuracy up to 98.81%, with a standard deviation of 0.0174. Fig. 6 shows the worst case found in the 500 iterations, with an accuracy of 97.08%, in which only 12 samples of the 411 ones are misclassified.

Table 7 summarizes the results of similar research in the literature. All of them reach a percentage of accuracy > 92%. Some of them differ markedly in the number of samples (from 62 in Mahata et al. [68] to 801 ones in the proposed work) or in the number of genes (from 32 in

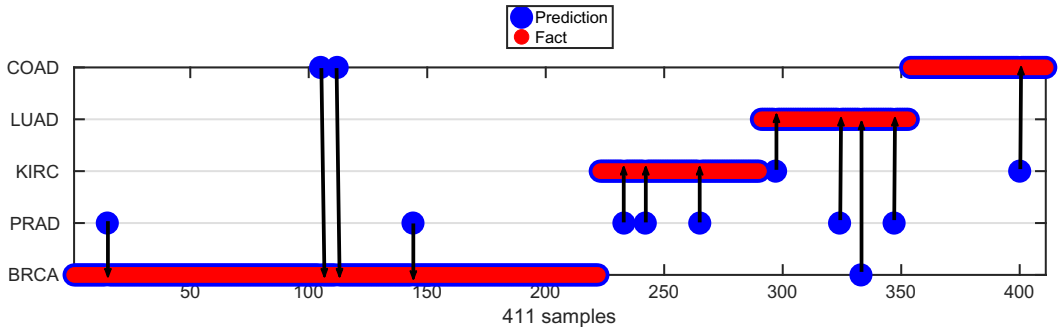

Fig. 6. Representation of the worst classification case (accuracy of 97.08%,) obtained by the voting system with the genes in Table 6. The validation dataset consisted of the 411. The x-axis corresponds to the validation dataset (411 samples) and the y-axis collects the 5 types of cancer.

Table 7
Achievements of accuracy in different research in the literature for cancer classification based on gene selection.

Author(s)	# Samples	# Features	# Classes	# Selected features	# Classifiers	Accuracy
Ding et al. (2005) [67]	96	4026	9	< 60	1	0.9730
Mahata et al. (2007) [68]	62	6000	2	15	1	0.9677
Liu et al. (2010) [69]a	97	24,481	2	7	7	0.9381
Liu et al. (2010) [69][b]	102	12,600	2	4	7	0.9706
Best et al. (2015) [72]	175	1072	2	–	110	0.9500
Piao et al. (2017) [73]	215	1047	4	–	20	0.9860
Saygılı (2018) [74]	569	32	2	24	1	0.9877
GGA	801	20,531	5	49	5	0.9881

Liu et al (2010)[a]with data from [70]*; Liu et al (2010)[b]with data from* [71].

Saygılı [74] to 24,481 in Liu et al. [69]). The presented method achieves the best accuracy of Table 7, even working a much higher number of features in most cases. Liu et al. [69] operate with 24,481 genes, similar number to the database used in our experiments, and obtain an accuracy of 0.9381 *versus* 0.9881, with the proposed algorithm.

6. Conclusions

The proposed GGA algorithm has been effective for the selection of features of a large database in a classification problem. The GGA has been successfully applied to a database from the UCI Repository composed of 801 samples and 20,531 characteristics that belong to 5 types of cancer (breast, kidney, colon, lung and prostate). There is no single nor simple solution. The objective is to find a solution close to the optimum, that is, a selection of features that works effectively as a high-accuracy classifier.

The fitness function of the algorithm is the ELM, which gives it speed in the computation time and accuracy in the classification. The GGA selects several candidate classifiers formed by a few characteristics (< 10 from the total of 20,531) that provide an average accuracy $>$ 90%. When several of these classifiers are combined in a voting system, the average accuracy of the classification is improved. < 50 genes among the 20,531 allowed to successfully classify 98.81% of the samples with a standard deviation of 0.0174. The worst case found in a series of 500 random and independent iterations had an accuracy of 97.08% in the classification process.

A limitation of the proposed algorithm is the possibly incomplete exploration of the solutions space. The number of features is massive and the selection of several tens of genes from 20,531 genes represents a huge search space. Proper space exploration is not guaranteed. One potential solution would be the parallel execution of the GGA on several computers that share subpopulations of individuals. Another idea could be the implementation of a metaheuristic algorithm to optimize the selection of gene groups in the voting center.

Conflicts of Interest: The authors declare that there is no conflict of interest. The authors alone are responsible for the content and writing of this article.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ygeno.2019.11.004.

References

[1] A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, et al., Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature 403 (6769) (2000) 503.

[2] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.

[3] H. Salem, G. Attiya, N. El-Fishawy, Classification of human cancer diseases by gene expression profiles, Appl. Soft Comput. 50 (2017) 124–134.

[4] R. Xu, G.C. Anagnostopoulos, D.C. Wunsch, Multiclass cancer classification using semisupervised ellipsoid ARTMAP and particle swarm optimization with gene expression data, IEEE/ACM Trans. Comput. Biol. Bioinforma. 4 (1) (2007) 65–77.

[5] G.F. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Trans. Inf. Theory 14 (1968) 55–63.

[6] M.W. Libbrecht, W.S. Noble, Machine learning applications in genetics and genomics, Nat. Rev. Genet. 16 (6) (2015) 321.

[7] A. Antoniadis, S. Lambert-Lacroix, F. Leblanc, Effective dimension reduction methods for tumor classification using gene expression data, Bioinformatics 19 (5) (2003) 563–570.

[8] G. Zhao, Y. Wu, Feature subset selection for cancer classification using weight local modularity, Sci. Rep. 6 (2016) 34759.

[9] J.C. Ang, A. Mirzal, H. Haron, H.N. Abdull Hamed, Supervised unsupervised and semi-supervised feature selection: a review on gene selection, IEEE/ACM Trans. Comput. Biol. Bioinforma. 13 (5) (2016) 971–989.

[10] D. Pavithra, B. Lakshmanan, Feature selection and classification in gene expression cancer data, 2017 International Conference on Computational Intelligence in Data Science, IEEE, 2017, pp. 1–6.

[11] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (1–3) (2002) 389–422.

[12] Y. Piao, K.H. Ryu, Detection of differentially expressed genes using feature selection approach from RNA-seq, 2017 IEEE International Conference on Big Data and Smart Computing, IEEE, 2017, pp. 304–308.

[13] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinforma. Comput. Biol. 3 (02) (2005) 185–205.

[14] C.-H. Zheng, D.-S. Huang, L. Shang, Feature selection in independent component subspace for microarray data classification, Neurocomputing 69 (16–18) (2006) 2407–2410.

[15] X. Zhu, N. Li, Y. Pan, Optimization performance comparison of three different group intelligence algorithms on a SVM for hyperspectral imagery classification, Remote Sens. 11 (6) (2019) 734.

[16] P. Maji, C. Das, Relevant and significant supervised gene clusters for microarray cancer classification, IEEE Trans.Nanobiosci. 11 (2) (2012) 161–168.

[17] J. Brimberg, N. Mladenović, R. Todosijević, D. Urošević, Solving the capacitated clustering problem with variable neighborhood search, Ann. Oper. Res. 272 (1–2) (2019) 289–321.

[18] G. Palubeckis, A. Ostreika, D. Rubliauskas, Maximally diverse grouping: an iterated Tabu search approach, JORS 66 (4) (2015) 579–592.

[19] K. Singh, S. Sundar, A new hybrid genetic algorithm for the maximally diverse grouping problem, Int. J. Mach. Learn. Cybern. (2019) 1–20.

[20] E. Bonilla-Huerta, A. Hernandez-Montiel, R. Morales-Caporal, M. Arjona-López, Hybrid framework using multiple-filters and an embedded approach for an efficient selection and classification of microarray data, IEEE/ACM Trans. Comput. Biol. Bioinforma. 13 (1) (2016) 12–26.

[21] Y. Zhang, Q. Deng, W. Liang, X. Zou, An efficient feature selection strategy based on multiple support vector machine technology with gene expression data, Hindawi. BioMed Res. Int. (2018), https://doi.org/10.1155/2018/7538204.

[22] I. Salman, O.N. Ucan, O. Bayat, K. Shaker, Impact of metaheuristic iteration on artificial neural network structure in medical data, Processes 6 (5) (2018) 57, https://doi.org/10.3390/pr6050057.

[23] A. Feitosa Neto, A.M. Canuto, J.C. Xavier-Junior, Hybrid metaheuristics to the automatic selection of features and members of classifier ensembles, Information 9 (268) (2018) 1–25, https://doi.org/10.3390/info9110268.

[24] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.

[25] G.B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (2008) 3460–3468.

[26] A. Akusok, K.M. Björk, Y. Miche, A. Lendasse, High-performance extreme learning machines: a complete toolbox for big data applications, IEEE Access 3 (2015) 1011–1025.

[27] G.B. Huang, H. Xiaojian, D. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (2010) 155–163.

[28] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern. B Cybern. 42 (2012) 513–529.

[29] S. Bashir, U. Qamar, F.H. Khan, Heterogeneous classifiers fusion for dynamic breast cancer diagnosis using weighted vote based ensemble, Qual. Quant. 49 (5) (2015) 2061–2076, https://doi.org/10.1007/s11135-014-0090-z.

[30] A. Ekbal, S. Saha, Weighted vote-based classifier ensemble for named entity recognition: a genetic algorithm-based approach, ACM Trans. Asian Lang. Inform. Process. 10 (2) (2011), https://doi.org/10.1145/1967293.1967296 Article 9, 37 pages.
[31] Ankit; Saleena, N, An ensemble classification system for twitter sentiment analysis, Proc. Comput. Sci. 132 (2018) 937–946.
[32] A. Onan, S. Korukoğlu, Bulut. H. a multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification, Expert Syst. Appl. 62 (15) (2016) 1–16.
[33] J. García-Gutiérrez, D. Mateos-García, M. Garcia, J.C. Riquelme-Santos, An evolutionary-weighted majority voting and support vector machines applied to contextual classification of LiDAR and imagery data fusion, Neurocomputing 163 (2015) 17–24.
[34] P. García-Díaz, J.A. Martínez-Rojas, M. Utrilla-Manso, L. Monasterio-Expósito, Analysis of water, ethanol, and fructose mixtures using nondestructive resonant spectroscopy of mechanical vibrations and a grouping genetic algorithm, Sensors (8) (2018) 1–19, https://doi.org/10.3390/s18082695 2695.
[35] C.-W.T. Yeu, M.-H. Lim, G.-B. Huang, A. Agarwal, Y.-S. Ong, A new machine learning paradigm for terrain reconstruction, IEEE Geosci. Remote Sens. Lett. 3 (3) (2006) 382–386.
[36] S.D. Handoko, K.C. Keong, Y.-S. Ong, G.L. Zhang, V. Brusic, Extreme learning machine for predicting HLA-peptide binding, Lect. Notes Comput. Sci 3973 (2006) 716–721.
[37] N.-Y. Liang, P. Saratchandran, G.-B. Huang, N. Sundararajan, Classification of mental tasks from EEG signals using extreme learning machine, Int. J. Neural Syst. 16 (1) (2006) 29–38.
[38] Y. Lan, Y. Soh, G. Huang, Extreme learning machine based bacterial protein subcellular localization prediction, IEEE Int. Jt Conf. Neural Networks (2008) 1859–1863.
[39] T. Helmy, Z. Rasheed, Multi-category bioinformatics dataset classification using extreme learning machine, IEEE Trans. Evol. Comput. (2009) 3234–3240.
[40] X. Luo, X. Chang, X. Ban, Regression and classification using extreme learning machine based on L1-norm and L2-norm, Neurocomputing 174 (Part A) (2016) 179–186.
[41] L. Cornejo-Bueno, J.C. Nieto-Borge, P. García-Díaz, G. Rodríguez, S. Salcedo-Sanz, Significant wave height and energy flux prediction for marine energy applications: a grouping genetic algorithm—extreme learning machine approach, Renew. Energy 97 (2016) 380–389.
[42] L. Duan, M. Bao, J. Miao, Y. Xu, J. Chen, Classification based on multilayer extreme learning machine for motor imagery task form EEG signals, Procedia Comput. Sci. 88 (2016) 176–184.
[43] D. Wicker, M.M. Rizki, L.A. Tamburino, The multi-tiered tournament selection for evolutionary neural network synthesis, Symposium on Combinations of Evolutionary Computation and Neural Networks, IEEE, 2000, pp. 207–215.
[44] H. Xie, M. Zhang, P. Andreae, M. Johnson, An analysis of multi-sampled issue and no-replacement tournament selection, Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, Atlanta, GA, USA, 2008, pp. 1323–1330.
[45] Ş.İ. Birbil, S.C. Fang, R.L. Sheu, On the convergence of a population-based global optimization algorithm, J. Glob. Optim. 30 (2–3) (2004) 301–318.
[46] W. Huyer, A. Neumaier, Global optimization by multilevel coordinate search, J. Glob. Optim. 14 (1999) 331–355.
[47] P. De Lit, E. Falkenauer, A. Delchambre, Grouping genetic algorithms: an efficient method to solve the cell formation problem, Math. Comput. Simul. 51 (2000) 257–271.
[48] E. Falkenauer, The grouping genetic algorithms: widening the scope of the GAs, Belgian Journal of Operations Research, Stat. Comput. Sci. 33 (1993) 79–102.
[49] E. Falkenauer, Genetic Algorithms for Grouping Problems, Wiley, New York, UK, 1998.
[50] T.L. James, E.C. Brown, K.B. Keeling, A hybrid grouping genetic algorithm for the cell formation problem, Comput. Oper. Res. 34 (2007) 2059–2079.
[51] E.C. Brown, R.T. Sumichrast, Evaluating performance advantages of grouping genetic algorithms, Eng. Appl. Artif. Intell. 18 (2005) 1–12.
[52] R.l. Kohavi, G.H. Jonh, Wrappers for features subset selection, Artif. Intell. 97 (1997) 273–324.
[53] G.B. Huang, L. Chen, Convex incremental extreme learning machine, Neurocomputing 70 (2007) 3056–3062.
[54] G.B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2011) 107–122.
[55] G. Huang, G.B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Netw. 61 (2015) 32–48.
[56] J.X. Xu, W. Wnag, J.C.H. Goh, G. Lee, Internal model approach for gait modeling and classification, The 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Shanghai, China, 2005, pp. 1–4.
[57] L.L. Chamara Kasun, Y. Yang, G.B. Huang, Z. Zhang, Dimension reduction with extreme learning machine, IEEE Trans. Image Process. 25 (8) (2016).
[58] M. Lichman, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, USA, 2013 Available online: http://archive.ics.uci.edu/ml (accessed on March 2019).
[59] Y. Yuan, E.M. Van Allen, L. Omberg, N. Wagle, A. Amin-Mansour, A. Sokolov, H. Liang, Assessing the clinical utility of cancer genomic and proteomic data across tumor types, Nat. Biotechnol. 32 (2014) 644–652.
[60] V. Cestarelli, G. Fiscon, G. Felici, P. Bertolazzi, E. Weitschek, CAMUR: knowledge extraction from RNA-seq cancer data through equivalent classification rules, Bioinformatics 32 (2016) 697–704.
[61] John N. Weinstein, et al., The cancer genome atlas pan-cancer analysis project, Nat. Genet. 45 (10) (2013) 1113–1120.
[62] G. Abu-Lebdeh, R.F. Benekohal, Convergence variability and population sizing in micro-genetic algorithms, Comput.-Aided Civ. Infrastruct. Eng. 14 (5) (1999) 321–334.
[63] E.C. Brown, R.T. Sumichrast, Evaluating performance advantages of grouping genetic algorithms, Eng. Appl. Artif. Intell. 18 (1) (2005) 1–12.
[64] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (2) (1999) 82–102.
[65] Quiroz-Castellanos, M.; Cruz-Reyes, L.; Torres-Jimenez, J.; Gómez S., C.; Fraire Huacuja, H. J.; Alvim, A. C. F. A grouping genetic algorithm with controlled gene transmission for the bin packing problem. Comput. Oper. Res., 2015, 55, 52–64, ISSN 0305–0548. https://doi.org/10.1016/j.cor.2014.10.010.
[66] T. Kucukyilmaza, H.E. Kizilozba, Cooperative parallel grouping genetic algorithm for the one-dimensional binpacking problem, Comput. Ind. Eng. 125 (2018) 157–170.
[67] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinforma. Comput. Biol. 3 (02) (2005) 185–205.
[68] P. Mahata, K. Mahata, Selecting differentially expressed genes using minimum probability of classification error, J. Biomed. Inform. 40 (6) (2007) 775–786.
[69] H. Liu, L. Liu, H. Zhang, Ensemble gene selection for cancer classification, Pattern Recogn. 43 (8) (2010) 2763–2772.
[70] L.J. van't Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, S.H. Friend, Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (6871) (2002) 530–536.
[71] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. D'Amico, J.P. Richie, E.S. Lander, M. Loda, P.W. Kantoff, T.R. Golub, W.R. Sellers, Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209.
[72] M.G. Best, N. Sol, I. Kooi, J. Tannous, B.A. Westerman, F. Rustenburg, et al., RNA-Seq of tumor-educated platelets enables blood-based pan-cancer, multiclass, and molecular pathway cancer diagnostics, Cancer Cell 28 (5) (2015) 666–676.
[73] Y. Piao, M. Piao, K.H. Ryu, Multiclass cancer classification using a feature subset-based ensemble from microRNA expression profiles, Comput. Biol. Med. 80 (2017) 39–44.
[74] A. Saygılı, Classification and diagnostic prediction of breast cancers via different classifiers, Int. Sci. Vocat. Stud. J. 2 (2) (2018) 48–56.