Saudi Journal of Biological Sciences 25 (2018) 932–946

Contents lists available at ScienceDirect

Saudi Journal of Biological Sciences

journal homepage: www.sciencedirect.com

Original article

DQB: A novel dynamic quantitive classification model using artificial bee colony algorithm with application on gene expression profiles

Hala M. Alshamlan *

Information Technology Department, King Saud University, Riyadh, Saudi Arabia
Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, United States

ARTICLE INFO

Article history:
Received 27 December 2017
Revised 30 January 2018
Accepted 31 January 2018
Available online 9 February 2018

Keywords:
Classification rule
Artificial Bee Colony Algorithm
Quantitive rule-based classification model
Microarray
Gene expression profile
ABC
Cancer gene selection

ABSTRACT

In the medical domain, it is very significant to develop a rule-based classification model. This is because it has the ability to produce a comprehensible and understandable model that accounts for the predictions. Moreover, it is desirable to know not only the classification decisions but also what leads to these decisions. In this paper, we propose a novel dynamic quantitative rule-based classification model, namely DQB, which integrates quantitative association rule mining and the Artificial Bee Colony (ABC) algorithm to provide users with more convenience in terms of understandability and interpretability via an accurate class quantitative association rule-based classifier model. As far as we know, this is the first attempt to apply the ABC algorithm in mining for quantitative rule-based classifier models. In addition, this is the first attempt to use quantitative rule-based classification models for classifying microarray gene expression profiles. Also, in this research we developed a new dynamic local search strategy named DLS, which is improved the local search for artificial bee colony (ABC) algorithm. The performance of the proposed model has been compared with well-known quantitative-based classification methods and bio-inspired meta-heuristic classification algorithms, using six gene expression profiles for binary and multi-class cancer datasets. From the results, it can be concludes that a considerable increase in classification accuracy is obtained for the DQB when compared to other available algorithms in the literature, and it is able to provide an interpretable model for biologists. This confirms the significance of the proposed algorithm in the constructing a classifier rule-based model, and accordingly proofs that these rules obtain a highly qualified and meaningful knowledge extracted from the training set, where all subset of quantitive rules report close to 100% classification accuracy with a minimum number of genes. It is remarkable that apparently (to the best of our knowledge) several new genes were discovered that have not been seen in any past studies. For the applicability demand, based on the results acqured from microarray gene expression analysis, we can conclude that DQB can be adopted in a different real world applications with some modifications.

1. Introduction

Efficiently comprehensible and accurate mining classifiers for large datasets are of growing interest in many applications. There is growing evidence that rule-based classification models can produce more efficient and accurate classification systems than traditional classification techniques (Ma, 1998). The main goal in this study is to discover knowledge that is not only correct but also comprehensible and simple for humans to understand. It is the nature of the human mind to be able to understand rules much better than any other data mining model. However, these rules need to be simple and comprehensive. The user can understand the results produced by the rule-based classifier and combine them with their own knowledge to make a well-informed decision, rather than blindly trusting results that are produced by the classifier. Moreover, in the medical domain, it is important to develop a *quantitative rule-based classification model.* The most common motivations for using quantitative rule extraction are to obtain a set of quantitative rules that can explain the black box

[*] Address: Information Technology Department, King Saud University, Riyadh, Saudi Arabia.
E-mail addresses: halshamlan@ksu.edu.sa, halaa@mit.edu

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.sjbs.2018.01.017
1319-562X/© 2018 The Author. Production and hosting by Elsevier B.V. on behalf of King Saud University.

classification model and to help specialists make accurate cancer diagnoses and provide effective chemotherapy treatment.

However, investigating classification rules is challenging due to the large size and noisy structure of the datasets, such as the microarray gene expression profile. The problem with mining classification rules in large databases involving both numerical and categorical attributes is that they are not straightforward (Kianmehr and Alhajj, 2008). As the size of microarray datasets is very large and high-dimensional, most classification rule mining algorithms generate an extremely large number of rules, often in the thousands. It is nearly impossible for a diagnosis to comprehend or validate such a large number of complex classification quantitative rules, which limits the usefulness of the data mining results. This is due to the great number of possibilities for discretizing numeric attributes into intervals; however, in categorical classification rules, all possible rules are considered. Actually, mining quantitative rule-based classification models is an optimization problem rather than a simple discretization one. Therefore, due to the difficulty of figuring out the meaningful and accurate quantitative rules, mining quantitative rule-based classification models is not a simple extension of mining categorical classification rules (Ma, 1998). In addition, using a bio-inspired evolutionary algorithm for mining quantitative rule-based classification models is, to our knowledge, a fairly new field of interest that requires more research. Thus, using the ABC algorithm to mine a quantitative rule-based classifier model is the challenging research area that we would like to explore in this paper.

Evolutionary meta-heuristic algorithms perform well in problems with vast search spaces and produce near optimal solutions, such as the ABC algorithm. Nevertheless, the use of the ABC algorithm for mining classification rules in the context of data mining is still a research area that few papers have tried to explore. Recently, only two research algorithms have been explored for mining rules using concepts based on the ABC algorithm, which are the ABCMiner algorithm (Celik et al., 2011) and the BeeMiner algorithm (Talebi and Abadi, 2014). The performance of these algorithms was compared with that of other classification data mining techniques, and the empirical results showed that the classification data mining models based on the ABC algorithm had comparative results in terms of predictive accuracy(Celik et al., 2011; Talebi and Abadi, 2014). These algorithms attempted to work by mining a discrete and simple dataset rather than a continuous and high dimensional dataset, such as a microarray dataset. Thus, to the best of our knowledge, this is the first attempt to apply the ABC algorithm as a quantitative rule-based classifier. In addition, this is the first attempt to explore and identify quantitative classification rules from a microarray gene expression profile.

In this research, we propose DQB mining dynamically a quantitative rule-based classification model by using the ABC algorithm. The model is applied on microarray gene expression profiles. The ABC algorithm dynamically discovers accurate intervals in the generated qualitative rule-based classification model that can lead to the best possible accuracy. In addition, this is the first attempt to use quantitative rule-based classification models for classifying microarray gene expression profiles. Also, in this research we developed a new dynamic local search strategy named DLS, which is improved the local search for artificial bee colony (ABC) algorithm. The proposed DQB algorithm is compared with other recently proposed classification algorithms applied on six gene expression profiles for binary and multi-class cancer datasets. The result shows that the DQB algorithm is consistently superior to other classification algorithms over all microarray datasets. The generated quantitative rules added a greater level of understanding regarding the classification model by using not only genes in classification but also the levels of their expression. This can reveal important biological discoveries and achieves superior improvement with respect to predictive accuracy, in addition to the simplicity of the rule generation process. Therefore, we can conclude that the DQB algorithm is a promising method for classification problems when applied to the classification of high dimensional data, such as cancer microarray gene expression profiles.

The rest of this paper is organized as follows. In Section 2, we briefly discuss the definition of a quantitative rule-based classification model. Subsequently, in Section 3, we present the application of evolutionary rule-based classification models that have recently been proposed in the state of the art. This is followed by an explanation of our proposed *DQB* algorithm in Section 4. After that, Section 5 outlines the experimental setup and provides results. Finally, Section 5.3 concludes this paper.

2. Quantitative rule-based classification model

Classification is an significant data mining problem that is widely applied in numerous real world applications. The primary task of a cancer microarray data classification approach is to determine a computational model from a given microarray data that determines the class of cancer for unknown samples. Accuracy, quality, and robustness are important elements of such models. The accuracy of the cancer microarray dataset classification depends on both the quality of the provided microarray data and the used classification approach. The Classification task or class prediction is the process of assignment correct labels to samples depend on their expression patterns, which is based on statistical or supervised machine learning approaches (Alshamlan et al., 2013) usually needs to identify which genes are informative and predictive for distinguishing the pre-defined classes. Then, the mathematical function used to estimate the accuracy of the classifier. Class classification is a widely applicable and beneficent data mining approach for medical diagnostic classification problems, prognostic prediction, and treatment selection. In addition, most cancer researches in microarray expression profiling foucs on class comparison or class prediction (i.e. classification).

In order to overcome the understandability problem of the classification task (Ma, 1998), rule-based classification techniques have recently been proposed and have received great consideration (Liu et al., 2004a). Thus, we aim to explore dataset instances to identify a rule set that can predict the class of each instance by using its attributes. These rule sets will then be used to build a classification model to determine the class of objects that have an unknown class. Many researchers have considered applying the machine learning algorithms (i.e., Neural Network (NN), Decision Tree(DT), Bayesian Network (BN), and Support Vector Machine (SVM)) in classification problems. However, despite their efficient performance in real world problems, machine learning algorithms have some disadvantages. First, they are based on mathematical and statistical techniques that use domain-independent biases to select and identify the rules. Therefore, they are not able to identify all of the important and comprehensible rules in the analyzed dataset. Second, the selected rules may not satisfy domain experts' expectations and interests. For example, some important rules might be needed to understand the classification task selected by machine learning algorithms.

The related literature indicates that quantitative rule-based classifiers have better results than machine learning classification algorithms (Salleb-aouissi et al., 2013). However, mining quantitative rule-based classification models for continuous (or numeric) attributes is still a major research issue (Salleb-aouissi et al., 2013). Furthermore, it is more difficult than mining classical rule-based classification models that have been widely used for discrete (categorical) datasets. In addition, in mining rule-based classification, there is only one predetermined target, while in mining quantitative rule-based classification models, the target of discovery is not predetermined (Ma, 1998). Therefore, it is a challenging problem, especially when applied to high dimensional datasets (with a large number of attributes and a small number of instances), such as in microarray gene expression profiles. The number of different possible combinations of the parameter is so high that algorithms based on exhaustive search become computationally unfeasible. Therefore, overfitting is considered a vital problem in generating quantitative classification rules (Liu et al., 2004b; Ma, 1998).

Let $I = (I_1, I_2, \ldots, I_n)$ be a set of instances in the dataset D. Each Instance $I_k \in I$ follows the scheme $(A_1, A_2, \ldots, A_n, C)$, where $A_1, A_2, \ldots, A_n$ are continuous attributes and C is a set of discrete class labels. An instance of the microarray dataset has the form $(a_1, a_2, \ldots, a_n, c)$, where each a_i equals the gene expression the value of gene A_i $(i \in 1, 2, \ldots, n)$ and c is a class label $\in C$.

In general, a classification rule consist of two parts: the antecedent and the consequent, which are an implications in the form of $X \rightarrow Y$, where X is called antecedent while Y is called consequent. This means that X implies Y. However, this rule can be used for classification if and only if the consequence Y is a single attribute-value pair (i.e., class label).

In quantitative rule mining, the antecedent part (IF) contains quantitative conditions with conjunctions in the form of:

$$IF\ (MaxValue_1 > attribute_1 > MinValue_n)\ \text{and}$$
$$\ldots.\ \text{and}\ (MaxValue_n > attribute_n > MinValue_n).$$

Each condition is associated with an specific attribute and examines whether its value belongs to the instance attribute value interval: minimum value *MinValue* and maximum value *MaxValue*.

Whereas, the consequent part (THEN) is in the form of:

$$THEN\ InsClass = PClass,$$

where *InsClass* refers to the instance class and *PClass* refers to the predicted class. If the conditions match the instance's attributes in the antecedent part, then the classifier assigns the predicted class in the consequent part for the instance. This process is repeated until a new predictive and accurate rule is found. There are different fitness measures in the literature, such as entropy and gene index, that can be used (Talebi and Abadi, 2014).

3. Evolutionary rule-based classification models

In the literature, meta-heuristic evolutionary algorithms have been applied to discover rule-based classification models. As shown in Fig. 1, the evolutionary algorithms is applied for desired dataset in many iteration. It generating one rule in each iteration (best individual) discovered with the highest fitness. If the number of target rules is not reached, evolutionary algorithm allows punishment in the previously covered instances, with the aim of discovering rules that cover those instances that have not been covered yet in previous iterations. The major benefit of the evolutionary algorithm is that aims and seeks to cover all regions in the solutions domain, which is the set of rules will cover all the consequent domain. The iterative operation ends when it discovers the target number of rules.

For instance, in Yan et al. (2009), the authors used a Genetic Algorithm (GA) to discover the classification rules without minimum support. A more recent algorithm presented in Salleb-aouissi et al. (2013) called the QuantMiner is a genetic-based algorithm for mining quantitative association rules. This algorithm discovered good intervals in association rules by optimizing both the support and confidence. Recently, a new algorithm that aimed to find all the frequent item sets from given datasets using a genetic algorithm has been proposed (Jaiswal and Dubey, 2013). Obviously, both of these genetic-based algorithms have been used for an optimization task rather than a classification one. meanwhile, the algorithms proposed in Al-maqaleh and Shahbazkia (2012) were designed to discover classification rules using a genetic algorithm.

Applying a Particle Swarm Optimization (PSO) algorithm as a new tool for data mining to discover classification rules has been proposed in Liu et al. (2004a) and Sousa et al. (2004). Consequently, these algorithms were evaluated and tested against a GA and other data mining algorithms. From the generated results, the PSO algorithm proved to be a efficient approach for classification tasks. In addition, Ant Colony Optimization (ACO) algorithms have been successfully applied for classification rule mining problems. Ant-Miner, the first ACO-based algorithm for the classification task of data mining, was presented in Parpinelli et al. (2002a) and Parpinelli et al. (2002b). In Otero et al. (2008), the authors explored an extension to the Ant-Miner algorithm (named the cAnt-Miner algorithm) which was applied to continuous datasets to find discrete interval rules for continuous attributes. The cAnt-Miner algorithm has been compared to the Ant-Miner algorithm, and the experiment results show that creating discrete interval (quantitative) rules facilitates the discovery of more accurate and significantly simpler classification rules (Otero et al., 2008). A hybrid particle swarm optimization and ant colony optimization (PSO-ACO) algorithm for the discovery of classification rules was explored in Holden and Freitas (2008). The authors compared the algorithm to an industry standard algorithm, Pruning rule based classification tree (PART) algorithm, with continuous data. The results showed that the proposed algorithm was very competitive in terms of accuracy and rule set simplicity. It is worth mentioning that PSO showed a good convergence speed and was faster than ACO with the same performance when they were used for mining classification rules. The hybrid (PSO-ACO) algorithm was faster than other evolutionary algorithms but had slightly more memory usage. However, all of these efforts provided a good insight into the efficiency of applying evolutionary algorithms for classification rule mining problems.

It is obvious that the ABC algorithm has been successfully applied to several optimization problems. Nevertheless, the use of the ABC algorithm for mining classification rules in the context of data mining is still a research area that few papers have tried to explore. Only two research algorithms have been explored for mining rules using concepts based on the ABC algorithm, which are the ABCMiner algorithm (Celik et al., 2011) and the BeeMiner algorithm (Talebi and Abadi, 2014). The performances of these algorithms were compared with other classification data mining techniques, and the empirical results showed that classification data mining based on the ABC algorithm had a comparative result in terms of predictive accuracy (Celik et al., 2011; Talebi and Abadi, 2014). However, all of these algorithms attempted to mine a discrete and simple dataset rather than a continuous and high dimensional one, such as a microarray dataset.

Therefore, As far as we know, this is the first attempt to apply the ABC algorithm as a quantitative rule-based classifier. In addition, this is the first attempt to explore and identify mining quantitative rule-based classification models for microarray gene expression profiles. In the following section, we propose a novel quantitative rule based classifier named the *DQB* algorithm, which is based on the ABC algorithm, to discover quantitative classification rules (i.e., discrete interval rules for continuous attributes) from microarray gene expression profiles with higher predictive accuracy and a smaller and comprehensive rule set. In particular, the *DQB* algorithm does not require users to specify the minimum-support threshold. Instead of generating an unknown number of interesting rules, as in traditional mining models, only the important rules are generated based on a defined

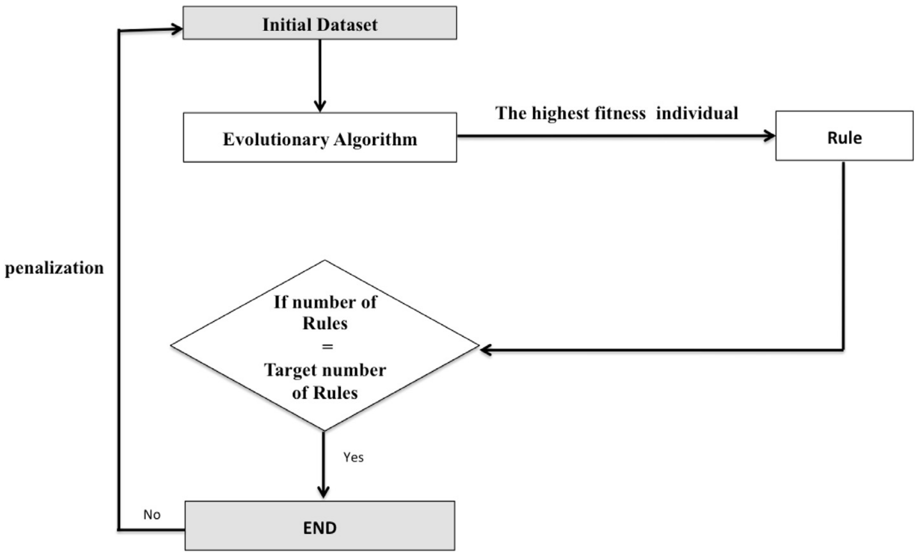

Initial Dataset
Evolutionary Algorithm
The highest fitness individual
Rule
penalization
If number of Rules = Target number of Rules
Yes
No
END

Fig. 1. Using evolutionary algorithms to discover rule based classification model.

measurement, which achieves classification accuracy by using a specific fitness function. Thus, *DQB* leads to effectiveness and efficiency for global search and system automation.

In addition, the most recently proposed evolutionary rule-based classification models suffer from efficiency problems. First, the rule-based classification algorithm generate a huge number of classification rules, and it is very hard to save the rules, retrieve the related classification rules, and prune and order the predictive classification rules. Second, it is challenging to identify the best subset of classification rules to build the most robust and accurate classifier (Ma, 1998). Therefore, in this paper, we address this challenge by developing an efficient quantitative rule-based classification model that is suitable for numerical datasets, such as microarray gene expression profiles. Our proposed *DQB* algorithm is built by using a subset of quantitative classification rules, namely QCRs, where the consequent of each rule is a class attribute.

4. The proposed algorithm (DQB) dynamic quantitive bee

In this section, we describe the approach we have taken to accomplish our primary goal of building a classification model. We develop a novel classification algorithm called *DQB*, which generates dynamically a quantitative classification rules from the ABC algorithm and carries out post-pruning to reduce the cardinality of the set of generated rules and building models from the pruned rules. *DQB* is a rule-based classification model that aims to make the classification task more understandable and efficient by integrating quantitative association rule mining techniques and the ABC algorithm.

The classic Artificial Bee Colony (ABC) algorithm usually looking for the optimal solution, the main steps of ABC algorithm illustrated in Fig. 2. In the ABC algorithm, there are three main phases for each cycle of the search of t: (1) the employed bees phase: each employed bees are sent to specific food sources to evaluate the nectar amount in each food source; (2) the onlookers phase: after collecting the nectar information for the food sources, the onlookers bee choose the food source location and evaluate the nectar amount in the food sources; and (3) the scouts bee phase: only one bee are designated as scout. the colony size divided by two groups contain equal number of bees. The first group of the colony includes of the employee bees and the second group contains the onlookers bee. Any employed bee of an abandoned food source becomes a scout bee. The onlookers and employed bees control the exploitation process in the search space, while the scouts bees carry out the exploration process (Alshamlan et al., 2016).

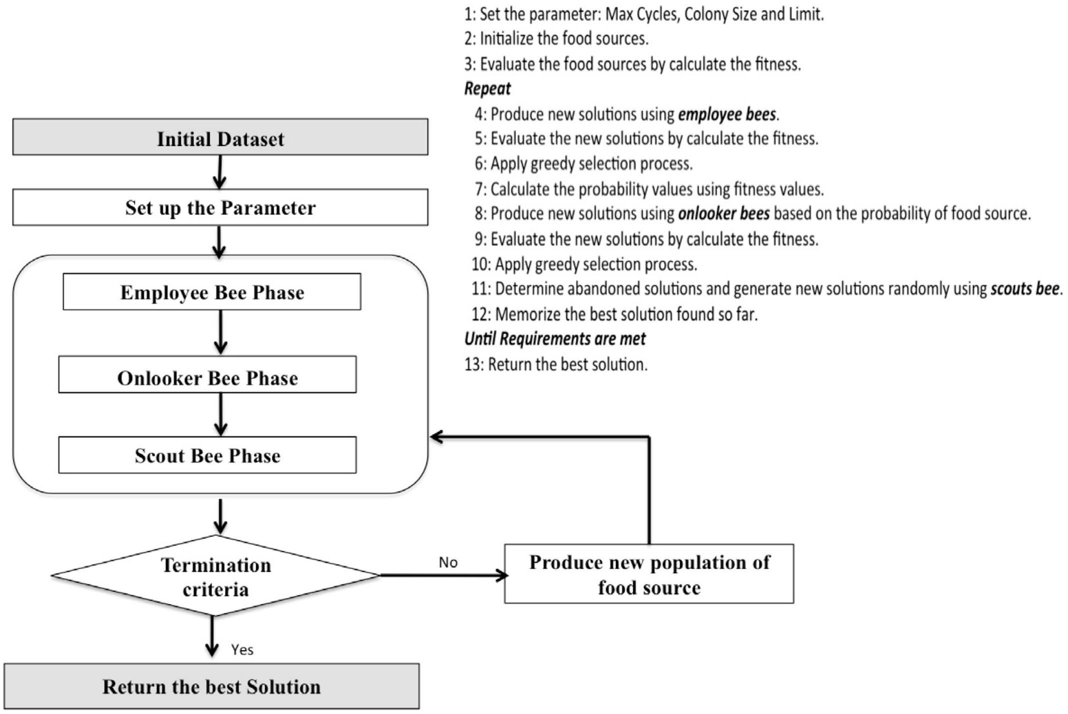
Based on previous researches mentioned previously, we investigate that the ABC algorithm has been used for optimization purposes. In addition, a few papers used it for classifying a discrete (categorical) dataset. Therefore, in this paper, we aim at applying this novel ABC-based algorithm for classifying continuous (numerical) datasets, such as a microarray gene expression profile. Our objectives are (1) to generate a set user-defined number of quantitative classification rules (*QCRs*) that acquire high classification accuracy, (2) to use the QCRs to build a classification system, and (3) to investigate the accurate, biologically meaningful quantitative rules and show their classification accuracy.

The *DQB* algorithm consists of four major phases: (1) *preprocessing phase,* (2) *rule generating phase*, (3) *rule pruning phase*, and (4) *lcassification and prediction phase*. The flowchart of phases and main steps for the *DQB* algorithm has been shown in Fig. 3. Below are the explanations of each phase.

4.1. Preprocessing phase

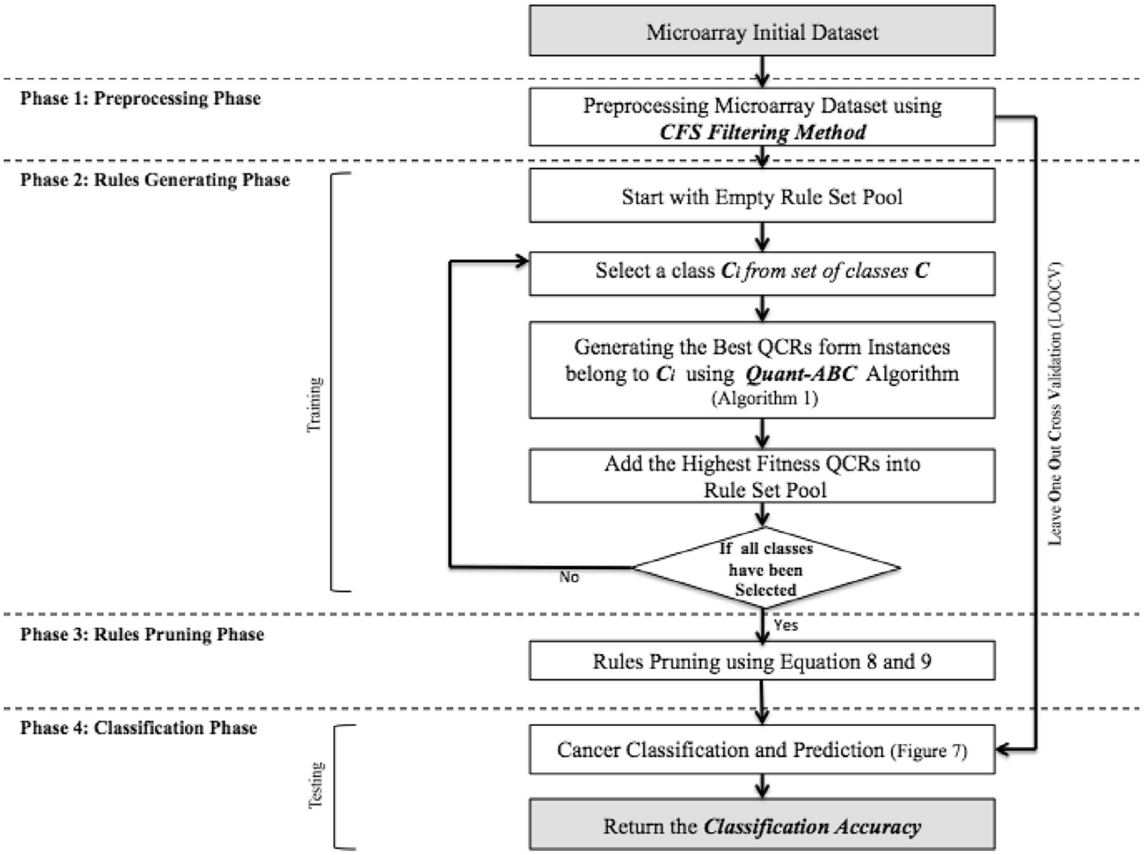
The microarray dataset is considered a dirty and high dimensional dataset. Therefore, as a first step, we reduced the dimensionality of the microarray dataset and elected the high informative genes by using a filter feature selection approach, named the correlation-based feature selection (CFS). The CFS algorithm scores (and ranks) the worth of subsets of features according to a correlation-based heuristic evaluation function, rather than scoring (and ranking) individual features (Yvan et al., 2007). As the space of the microarray feature (genes) is usually huge, CFS uses a best-first-search heuristic that takes into account the usefulness of individual features for predicting the class. Therefore, CFS selects the subset that has maximal correlation to the class and minimal correlation between features (Yvan et al., 2007). CFS first calculates a matrix of (feature to class) and (feature to feature) correlations from the training data. Then, a score of a subset of features assigned by the heuristic is calculated using Eq. (1),

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



```
1: Set the parameter: Max Cycles, Colony Size and Limit.
2: Initialize the food sources.
3: Evaluate the food sources by calculate the fitness.
Repeat
  4: Produce new solutions using employee bees.
  5: Evaluate the new solutions by calculate the fitness.
  6: Apply greedy selection process.
  7: Calculate the probability values using fitness values.
  8: Produce new solutions using onlooker bees based on the probability of food source.
  9: Evaluate the new solutions by calculate the fitness.
  10: Apply greedy selection process.
  11: Determine abandoned solutions and generate new solutions randomly using scouts bee.
  12: Memorize the best solution found so far.
Until Requirements are met
13: Return the best solution.
```

Fig. 2. The main steps of ABC algorithm.



Microarray Initial Dataset
Phase 1: Preprocessing Phase
Preprocessing Microarray Dataset using CFS Filtering Method
Phase 2: Rules Generating Phase
Start with Empty Rule Set Pool
Select a class Ci from set of classes C
Training
Generating the Best QCRs form Instances belong to Ci using Quant-ABC Algorithm (Algorithm 1)
Add the Highest Fitness QCRs into Rule Set Pool
If all classes have been Selected
No
Yes
Leave One Out Cross Validation (LOOCV)
Phase 3: Rules Pruning Phase
Rules Pruning using Equation 8 and 9
Phase 4: Classification Phase
Testing
Cancer Classification and Prediction (Figure 7)
Return the Classification Accuracy

Fig. 3. The main phases and steps of the proposed *DQB* Dynamic Quantitive Bee algorithm.

where $Merit_S$ is the heuristic merit of a feature subset S containing k features, $\overline{r_{cf}}$ is the average correlation between features and class, and $\overline{r_{ff}}$is the average correlation between features.

CFS starts from the empty set of features. Then, the subset with the highest merit found during the search is selected. In this study, we used Weka (University of Waikato, 1997) to implement CFS, and the selected informative and highly correlated gene subsets were used for the next phases.

4.2. Rule generating phase

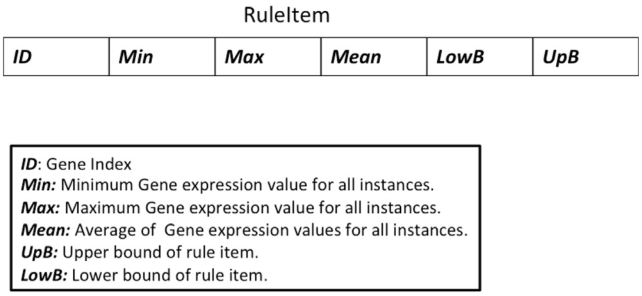
In this section, we discuss how we apply the ABC algorithm for generating QCRs. Our proposed *DQB* algorithm differs from the original ABC algorithm presented in our previous researches (Alshamlan et al., 2016; Alshamlan et al., 2015a). The pseudo code of rule discovery phase for the proposed *DQB* algorithm is presented in Algorithm 1. In this phase, we adopt four main components: (1) rule format, (2) rule discovery, (3) rule fitness evaluating, and (4) the enhanced local search (ELS) method. These main components are described below.

4.2.1. Rule format

Let $R(ri_1, ri_2, \ldots c, ri_n)$ be the quantitative classification rule (QCR) that consists of different attributes or items ri, namely rule item, where n is the number of *rule items* in each rule that are dependent on the number of genes defined by the user for a cancer classification task. For instance, if the user intended to use three genes for classification, this means that every classification rule R must contain three *rule items*. The number of genes or *rule items* appearing in the rule R is fixed (user-defined) in order to ensure a start with enough diversity in the initial population that can model accurate and efficient QCR rules (Salleb-aouissi et al., 2013).

Furthermore, Fig. 4 shows the representation of the solution space population (foods) for the *DQB* algorithm. The individual *rules* of the initial population are randomly generated. Each row represents a $Rule(rule\ item_1 \ldots crule\ item_D)$ that is a component for every food source or candidate solution (i.e., *QCR*).

In the rule-based classification model, a classification rule contains two parts: the antecedent and the consequent. Thus, a quantitative classification rule (QCR) has the form $(R : ri \rightarrow C)$, where C is a class label. Each *rule item* has its identification *id*, which is the gene index in the microarray dataset. There are also three values associated with the classification rule item(em-dash) max, min, and mean(em-dash which are (respectively) the maximum, minimum, and average gene expression values of the gene domain and has the gene index *id* for all instances belonging to the desired class *C*. The last two values are the most important in the classification task, which are the lower bound (lowb) and upper bound (upb) values. In our algorithm, we aim to identify the best match interval by generating the best *lowb* and *upb* values for each *rule item* $\in R$ using the enhanced local search ELS method. Consequently, the rule R can classify the instance correctly if the instance's gene expression values belong to R's *rule items* interval (i.e., in between the rule item's *upb* and *lowb* values) and it also has the same class label *C*. Otherwise, it misclassifies an instance. From the above explanation, every quantitative classification *rule item* designed as the structure is presented in Fig. 5.



ID: Gene Index
Min: Minimum Gene expression value for all instances.
Max: Maximum Gene expression value for all instances.
Mean: Average of Gene expression values for all instances.
UpB: Upper bound of rule item.
LowB: Lower bound of rule item.

Fig. 5. The structure of the rule item.

4.2.2. Rule discovery

The purpose of classification rule mining is to identify a set of rules that can predict the specific cancer class for different instances in a mircoarray dataset. Thus, rule discovery is the most important step in our classification model since the rule sets are the outcome of this phase.

In our proposed *DQB* algorithm, at the initialization stage, we generate the rule interval by setting the lower bound value *lowb* and upper bound value *upb* for every *rule item ri* for each rule *R* (food source positions) in the solution space (food) using the following equations:

$$ri.lowb = ri.mean - k_1 \times (ri.max - ri.min), \tag{2}$$

$$ri.upb = ri.mean + k_2 \times (ri.max - ri.min). \tag{3}$$

For these two equations, *ri.max* and *ri.min* are the maximum and minimum values of the *rule item ri*. The difference between them is the range of the *rule item ri*, where *ri.mean* is the average gene expression value of the gene domain and has the gene index ri.d for all instances belonging to the desired class *C* and *k*1 and *k*2 are two random values between 0 and 1. In addition, we make sure that the *ri.upb* value does not equal the *ri.lowb* value and, given a

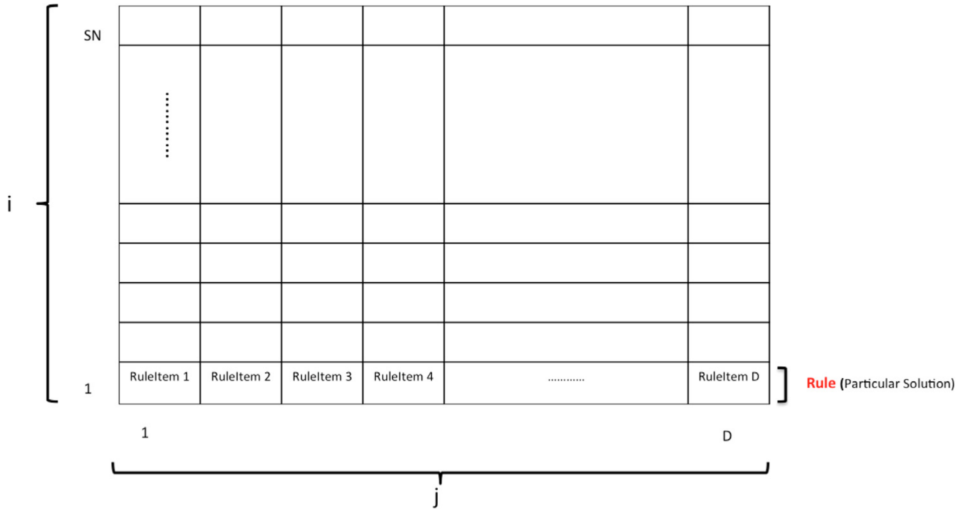


Fig. 4. The solution space for the *DQB* algorithm. *SN* represents the number of particular solutions or rules (food sources), and *D* represents the number *rule items* to be optimized for each solution or quantitative classification rule *QCR*. Each cell represent different *rule items*.

granularity, that the *lowb* of the interval has to be less than the *upb* of the interval. If the obtained values do not match this requirement, the *lowb* and *upb* are swapped. In addition, the *lowb* and *upb* of the interval have to be greater and less than the *min* and *max* limits of the gene domain of the gene, respectively. Differently, the corresponding *min* and *max* limits of the gene domain are assigned.

Our proposed dynamic rule based classification algorithm *DQB* can automatically discover the classification rules for each class. For the selected class, it can determine the rules iteratively until the rule set can cover all instances belonging to that class.

4.2.3. Rule fitness evaluating

To evaluate the fitness value accurately, the fitness function presented in Eq. (4) will be used for training dataset classification to measure the quality of the food source (rule) rather than of measuring the nectar amount. Its representation is defined below (Shukran et al., 2011):

$$Fitness\ value = \frac{TP}{TP + FN} \times \frac{TN}{TN + FP}, \tag{4}$$

where *TP* represents the true positives, *FN* represents the false negatives, *FP* represents the false positives, and *TN* represents the true negatives associated with the rule. In addition, before we illustrate the meaning of *TP*, *FN*, *FP*, and *TN*, there are two important concepts, as demonstrated below:

- When the algorithm examines the instance, it will measure every gene index specified in the *rule item* for an evaluated instance. If the gene expression value of this gene index is between the lower bound *lowb* and upper bound *upb* for this *rule item*, this means that the gene index for the evaluated instance can be covered by the *rule item*. If all gene indices specified in the *rule items* for an evaluated instance can be discovered by the rule, this means that the evaluated instance can be discovered by the rule as well.
- The evaluated instance is denoted as (class predicted) by the rule when the class of the instance is equal to the predictive class of the rule.
  - *TP*: the number of instances that have their (class predicted) by the rule and are covered by the rule;
  - *FN*: the number of instances that have their (class predicted) by the rule and are not covered by the rule;
  - *FP*: the number of instances that have their (class not predicted) by the rule and are not covered by the rule;
  - *TN*: the number of instances that have their (class not predicted) by the rule and are not covered by the rule.

4.2.4. Dynamic Local Search (*DLS*) Method

After the initialization of solutions (generating rules), the *DQB* algorithm starts looking for the optimal solution (accurate *QCRs*). In the *DQB* algorithm, in each cycle, the employee and onlooker bees need to move to a new food source (solution) or search for new rules to optimize the classification result followed by the local search method. In our previous papers Alshamlan et al. (2015a,b, 2016), we implemented the original ABC's local search method. However, in this paper, we propose a new local search method called the dynamic local search (*DLS*) method to replace the original local search strategy.

In the *DLS* method, we change the position of food source (solution), which represents a rule in our classification problem by changing two *rule items* r_1 and r_2 in each rule, which are randomly selected. The enhanced local search method is shown in Fig. 6.

In the first *rule item* r_1, we update the position of employee and onlooker bee food source (rule) by changing the *rule item id*, which represents the gene index, by using the following equation:

$$V_{ir_1}.id = X_{kr_1}.id - \theta(X_{ir_1}.id - X_{kr_1}.id), \tag{5}$$

where $V_{ir_1}.id$ represents the new *rule item*'s gene index in the r_1 dimension for the new food source position V_i, $X_{kr_1}.id$ represents the *rule item*'s gene index in the r_1 dimension for the neighbor of the current food source position X_k, and $X_{ir_1}.id$ represents the *rule item*'s gene index in the r_1 dimension for the current food source position X_i. r_2 is a randomly chosen parameter between 1 and D, where θ is a random real value between 0 and 1. The value of i and k are between 1 and SN, but k has to be a different value from i. In addition, r_1 is the number of dimensions. In our classification problem, the dimension D equals the number of rule items in each rule, and $k \in [1, 2, \ldots SN]$ and $r_1 \in [1, 2, \ldots, D]$ are randomly chosen parameters.

In the second *rule item* r_2, we update the position of employee and onlooker bee food source (rule) by changing the *rule item*'s boundary parameters(em-dash)upper bound upb and lower bound lowb(em-dash)using the following equations:

$$V_{ir_2}.upb = X_{ir_2}.max - \alpha, \tag{6}$$

$$V_{ir_2}.lowb = X_{ir_2}.min + \alpha, \tag{7}$$

where $V_{ir_2}.upb$ and $V_{ir_2}.lowb$ represent (respectively) the new *rule item*'s upper bound and lower bound in the r_2 dimension for the new food source position V_i. In addition, $X_{ir_2}.max$ and $X_{ir_2}.min$ represents (respectively) the *rule item*'s maximum and minimum gene expression values in the r_2 dimension for current food source position X_i. r_2 is a randomly chosen parameter between 1 and D, where α is a random real value between 0 and 1.

4.3. Rule pruning phase

In the rule generating phase, for each class, we select the best rules which have the highest fitness value. Then, we add them to a rule set pool. However, the number of rules generated by previous phases can be excessive. Thus, to make the classification effective and efficient, we need to prune rules to delete redundant and

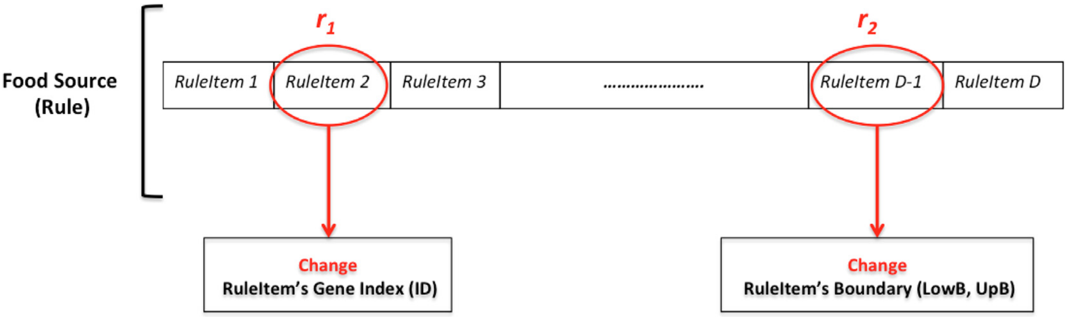


Fig. 6. The dynamic local search (*DLS*) method.

inefficient rules. The major goal of rule pruning is to select the most accurate quantitative classification rule *QCR* that has the highest prediction value for each class. According to the facility of rules' support and confidence on classification, a global order of rules is composed.

Given a training microarray sample *T*, let *C* be a class label. For rule $R : RI \rightarrow C$, the ratio of the number of instances in *T* matching rule items *RI* and having class label *C* is called the support of *R*, denoted as $Sp(R)$, which is computed as follows:

$$Sp(R) = \frac{TP}{N}. \tag{8}$$

The ratio of the number of instances matching rule items *RI* and having class label *C* versus the total number of instances matching rule items *RI* is called the confidence of *R*, denoted as $Cnf(R)$. The confidence can then be computed as follows:

$$Cnf(R) = \frac{TP}{TP + FP}. \tag{9}$$

DQB adopts the following steps for rule pruning. First, it uses higher confidence rules to prune lower confidence ones. Second, DQB prunes R2 if R1 has a higher rank. *DQB* prunes R_2 if R_1 also has higher rank than R_2. Given two rules R_1 and R_2, R_1 has a higher rank than R_2, denoted as $R_1 > R_2$, if and only if:

(1) $Cnf(R_1) > Cnf(R_2)$,
(2) $Cnf(R_1) = Cnf(R_2)$ but $Sp(R_1) > Sp(R_2)$.

After a set of *QCRs* is selected for classification, *DQB* is ready to classify new testing microarray samples. In the next phase, we discuss how to predict the class label based on the pruning *QCRs* rules.

4.4. Classification and prediction phase

In this phase, we will test the efficiency of *QCRs* generated in the previous phase. The pruned *QCR* classiction rule set will be used to classify the new testing microarray samples, which have unknown classes. However, sometimes, one testing sample will be discovered by more than one classification rule for a different class. When this occurs, the classification and prediction strategy will identify which class should be used for prediction. There are four main steps for the classification approach, which are specified as follows:

1. Determine the cover percentage for all QCRs that cover the testing sample, which defines the proportion of the samples covered by the QCR that have their class predicted by the rule (TP). This is calculated by the expression shows below:

   $$Cover\ percentage = \frac{TP}{Nc}, \tag{10}$$

   where *Nc* is the total number of instances which belong to the predicted class by the rules.
2. Compute the prediction value for all QCRs, which cover the testing sample. This is defined in Eq. (11) as shown below:

   $$Prediction\ value = (\alpha \times rule\ fitness\ value) + (\beta \times rule\ cover\ percentage), \tag{11}$$

   where α and β are two weighted parameters associated with rule fitness value and rule cover percentage, respectively, α is a randomly real value between 0 and 1, and $\beta = (1 - \alpha)$. Eq. (4) can calculate the fitness values for each rule.
3. Discriminate the class which has the highest prediction value as the predictive class.
4. Calculate the average accuracy for the given microarray dataset, which is defined as the proportion of correctly classified *CC* test samples:

   $$Classification\ accuracy = \frac{CC}{N}, \tag{12}$$

   where N is the total number of instances in the initial microarray dataset, and *CC* correctly classified the testing sample (i.e., CC = TP + TN), which is initially set by 0.

Finally, the *QCR* will be displayed with its classification accuracy. The summary of the main steps of the classification and prediction phase for the proposed *DQB* algorithm is shown in Fig. 7.

Algorithm 1. DQB Algorithm

```
Represent the solution space of DQB algorithm for a microarray dataset, as shown in Fig. 4.
For each employee bee
    Generate random initial solution (rule) consisting of $D$ rule items for each employee bee by using Eqs. (2) and (3)
    Calculate the fitness value using fitness function in Eq. (4).
    Reset the abandonment counter.
Do
    For each employee bee
        Randomly select neighbour food source (rule) for current employee bee
        Randomly select the first rule item to be change $r_1$
        Update the position of employ bee food source (rule) by changing the selected rule item's ID (gene index) using Eq. (5).
        Randomly select the second rule item to be change $r_2$
        Update the position of employee bee food source (rule) by changing selected rule item's boundary, upb and lowb, using Eqs. (6) and (7)
        Calculate the fitness value using fitness function in Eq. (4).
        IF fitness value of new solution > fitness value of old solution THEN
            Replace the old solution with the new one and reset the abandonment counter of the new solution.
        ELSE
            Increase the abandonment counter of the old solution by 1.
    For each onlooker bee
        Choose a food source (rule) depending on its probability to be chosen (roulette wheel).
        Randomly select the first rule item to be change $r_1$
        Update the position of food source (rule) by change the selected $r_1$'s ID (gene index) using Eq. (5), it should be >=1
        Randomly select the second rule item to be change $r_2$
        Update the position of food source (rule) by changing selected $r_2$'s boundary: upb and lowb, using Eqs. (6) and (7)
        Calculate the fitness value using fitness function in Eq. (4).
        IF fitness value of new solution > fitness value of old solution THEN
            Replace the old solution with the new one and reset the abandonment counter of the new solution.
        ELSE
            Increase the abandonment counter of the old solution by 1.
    Set the abandonment bee Limit $L$ by 5.
    Search for abandonment bee.
    IF the abandonment counter of bee > $L$ THEN
        Reset the abandonment counter of bee.
        Generate a new solution (rule) for the employee bee randomly.
Until termination condition is met
Return the generated rules.
```

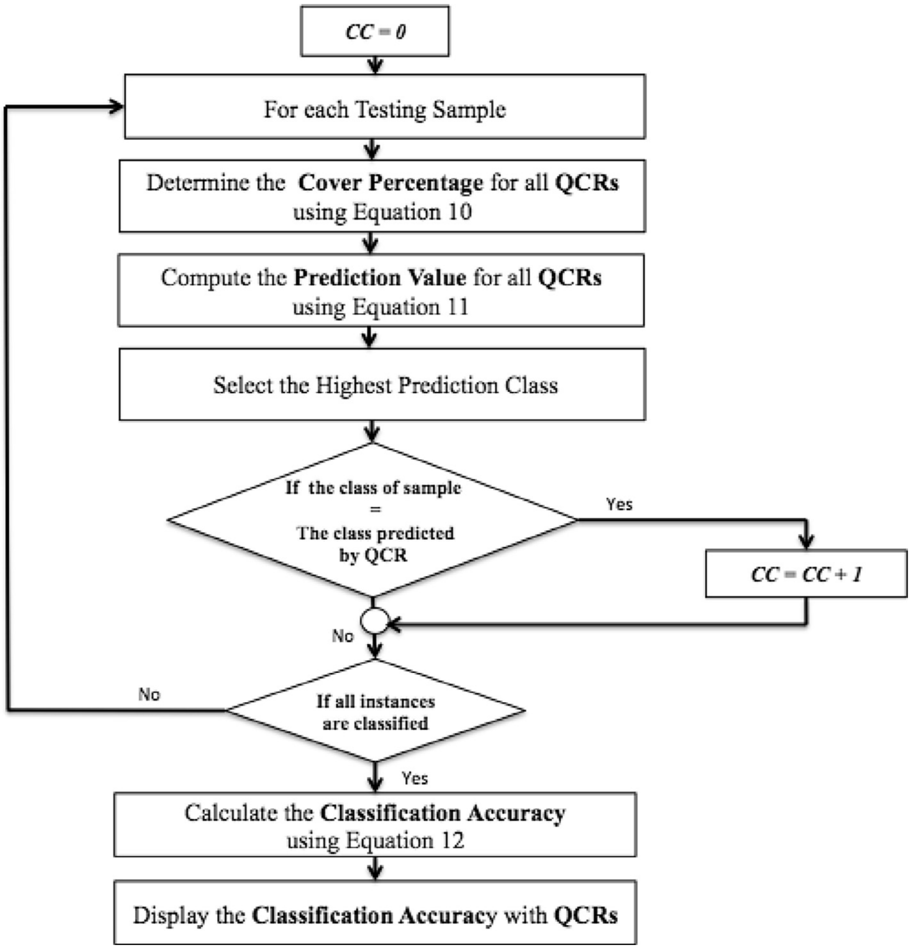

CC = 0
For each Testing Sample
Determine the Cover Percentage for all QCRs using Equation 10
Compute the Prediction Value for all QCRs using Equation 11
Select the Highest Prediction Class
If the class of sample = The class predicted by QCR
Yes
CC = CC + 1
No
If all instances are classified
No
Yes
Calculate the Classification Accuracy using Equation 12
Display the Classification Accuracy with QCRs

Fig. 7. The flowchart of classification and prediction phase.

5. Experimental setup and results

5.1. Experiential setup

In this section, we evaluate the overall performance of the *DQB* methods using six useful binary and multi-class microarray cancer datasets which we used to evaluate our previously proposed algorithms. The binary-class microarray datasets are *colon* (Alon et al., 1999), *leukemia* (Golub et al., 1999), and *lung* (Beer et al., 2002), where the multi-class microarray datasets are *SRBCT* (Khan et al., 2001), *lymphoma* (Alizadeh et al., 2000), and *leukemia* (Armstrong et al., 2001). In Table 1, we present a detailed description of these six benchmark cancer microarray gene expression datasets with related to the number of classes, samples, genes, and a brief description of each dataset construction.

Table 2 shows the control parameters for the *DQB* algorithm that was used in our experiments. The first control parameter is the *bee colony size* or population, with a value of 200. The other control parameter is the *maximum cycle*, which is mean the maximum number of generations. We used 2500 as value for this parameter. the third control parameter is the *number of runs* that was applied as a stopping condition. For this, A value of 30 was used for this parameter in our experiments, which has been shown to be acceptable. The last parameter is the *limit*, which equals to the maximum number of iterations allowed when the specific food source is not improved (exhausted). If the any food source exceeds this limit, it will be replaced by the scout bee. A value of 10 iterations was used for this parameter.

In this paper, we tested the performance of the proposed *DQB* algorithm by comparing it with other more populare bio-inspired algorithms, including Artificial Bee Colony (ABC) algorithm, Genetic Algorithm (GA), and Particle Swarm Optimaization (PSO) algorithm. We implemented GA, PSO, and Support Vector Machine (SVM) algorithms using the Waikato Environment for Knowledge Analysis (WEKA version 3.6.10), which is an open source data mining tool (University of Waikato, 1997). We compared the classification performance of each algorithm based on two parameters: the classification accuracy and the number of predictive genes that have been used for cancer class predication (classification).

In addition, we applied leave-one-out cross-validation (LOOCV) (Ng, 1997) in order to evaluate the performance of our proposed algorithm and the other related methods in comparison. We ran the DQB algorithm 30 times, and in each run, we applied LOOCV to calculate the classification accuracy. For each iteration, the generated accuracy was reported, which is the mean of the testing phase of all folds. In each fold, in the rule generating phase, our algorithm generated different rules (highest fitness rules), and we put them in the rule pool. In the rule pruning phase, we observed that the same rule had been discovered for each class in each fold. Thus, in each iteration after the rule pruning phase, we only had a single rule for each class. Furthermore, in order to make the experiments more statistically valid, we conducted each experiment 30 times on each dataset. Then, after we finished all iterations, we reported the best, which is the best accuracy result across all 30 runs, the mean, which is the average (mean) accuracy

Table 1
Statistics for cancer gene expression profiles.

Gene expression profiles	No of classes	No of samples	No of genes	Description
Colon (Alon et al., 1999)	2	62	2000	40 Cancer samples and 22 normal samples
Leukemia1 (Golub et al., 1999)	2	72	7129	25 AML samples and 47 ALL samples.
Lung (Beer et al., 2002)	2	96	7129	86 Cancer samples and 10 normal samples.
SRBCT (Khan et al., 2001)	4	83	2308	29 EWS cancer samples, 18 NB cancer samples, 11 BL cancer samples, and 25 RMS cancer samples.
Lymphoma (Alizadeh et al., 2000)	3	62	4026	42 DLBCL cancer samples, 9 FL cancer samples, and 11 B-CLL cancer samples.
Leukemia2 (Armstrong et al., 2001)	3	72	7129	28 AML sample, 24 ALL sample, and 20 MLL samples.

Table 2
The control parameters for DQB algorithm.

Parameter	Value
Colony$_{S}$*ize*	200
Max$_{C}$*ycle*	2500
Number$_{o}$*f*$_{R}$*un*	30
Limit	10

Table 3
The performance of the CFS with SVM classifier for cancer gene expression profile.

Microarray datasets	Number of genes	Classification accuracy
Colon	25	91.94%
Leukemia1	80	100%
Lung	71	100%
SRBCT	110	100%
Lymphoma	184	100%
Leukemia2	103	100%

across all 30 runs, and the worst, which is the worst accuracy across all 30 runs. And the *worst*, which is the worst accuracy of 30 runs.

5.2. Experimental results

In this section, we present and evaluate the results that are obtained by our proposed *DQB* algorithm. In order to overcome the basic artificial bee colony algorithm converges slowly and prematurely. And because any quantitive classification algorithm is very sensitive with noise. In our proposed algorithm, we employed the CFS filter method as first step in order to identify the high correlative and informative genes that give accurate accuracy with an SVM classifier. Also, to reduce the dimensionality and noise level of the microarray dataset. Then, we used these high correlative genes as inputs for our proposed *DQB* algorithm to generate meaningful quantitative rules that can be used for cancer classification and achieve accurate performance. From Table 3, we can see that the most correlated 80 genes from leukemia1 dataset achieve100% classification accuracy. For the colon dataset, we can generate 91.94 % classification accuracy using 25 genes. While in the lung dataset, we got 100% class-action accuracy with 71 genes and 110 genes to achieve the similar classification accuracy percentage for the SRBCT dataset. Also, with 184 highly correlated genes from the lymphoma dataset and 103 genes from the leukemia2 dataset, we generated 100% classification accuracy.

To highlight the understandability of the quantitative rules in the *DQB* model, the explanation for the most accurate (i.e., best) and most frequent QCRs in $IF - THEN$ form for all microarray cancer datasets are shown in Tables 4–9. It can be observe that the reported rules are very understandable by a domain expert because the antecedents of the rules are simply conjunctions of genes shown by their upper and lower pounds. Rules that are shorter in length (i.e., have a smaller number of *rule items*) are more effective since they are easy to comprehend. Thus, we changed the number of *rule items* (genes) that have been used for each rule in order to achieve an accurate and comprehensive result. To the best of our knowledge, *DQB* produces accurate results by producing quantitative classification rules that achieve 100% classification accuracy and a small number of *rule items* for all datasets. These rules have been examined by medical experts are considered to be interesting and adequate information.

By analyzing the classification rules generated from the colon dataset, as shown in Table 4, we observed four genes (249, 917, 682, 1671) as very significant to build the quantitative rule (QCR) that achieved 100% classification accuracy. This result is partially in agreement with the best result for colon cancer in the literature, which is reported by Lee and Leu (Lee and Leu, 2011), who showed that Gene 1671, Gene 286, Gene 1836, Gene 1058, Gene 1485, Gene 765, Gene 249, and Gene 1771 are the most significant genes that achieved 100% classification accuracy. There are two similar genes, which are Gene 249 and Gene 1671. However, *DQB* discovered fewer genes than GADP.

As seen in Table 5, the discovered rules from the leukemia1 dataset showed that the genes M31523_at and U46499_at are the most predictive genes in AML. Notably, by analyzing rules discovered from ALL samples, we observed that gene M27891 is very significant for building all generated quantitative rules with two rule items that achieved 98.61% classification accuracy. The CARSVM algorithm (Kianmehr and Alhajj, 2008), which is an association rule-based classification method (i.e., not quantitative), reported six genes (namely, M27891, M19507, L20941, X04085, Y007876, and M63138) as very significant for building the rules with 95.83% classification accuracy. Thus, this proves that M19507 is the most significant. This is also in agreement with

Table 4
The best quantitative classification rules for colon dataset.

Number of rule items	QCR	Accuracy
1	$IF\ (163.60875 \leqslant Gene[249] \leqslant 5832.3174)\ THEN$ $Class = Normal$	87.09%
2	$IF\ (62.7375 \leqslant Gene[1671] \leqslant 2076.9026)\ AND$ $(108.2 \leqslant Gene[682] \leqslant 998.61)\ THEN$ $Class = Tumor$	93.54%
3	$IF\ (103.42625 \leqslant Gene[249] \leqslant 2787.0425)\ AND$ $(62.7375 \leqslant Gene1[1671] \leqslant 2076.9026)\ AND$ $(108.2 \leqslant [Gene682] \leqslant 998.61)\ THEN$ $Class = Tumor$	98.38%
4	$IF\ (103.42625 \leqslant Gene[249] \leqslant 2787.0425)\ AND$ $(13.692857 \leqslant Gene[917] \leqslant 200.19167)\ AND$ $(108.2 \leqslant Gene[682] \leqslant 998.61)\ AND$ $(62.7375 \leqslant Gene[1671] \leqslant 2076.9026)\ THEN$ $Class = Tumor$	100 %

Table 5
The best quantitative classification rules for leukemia1 dataset.

Number of rule items	QCR	Accuracy
1	$IF\ (543.0 \leqslant Gene[X95735\_at] \leqslant 7133.0)\ THEN\ Class = ALL$	95.83%
2	$IF\ (110.0 \leqslant Gene[M31523\_at] \leqslant 671.0)\ AND\ (163..0 \leqslant Gene[U46499\_at] \leqslant 3166.0)\ THEN\ Class = AML$	100%
2	$IF\ (326.95 \leqslant Gene[D83776\_at] \leqslant 1180.27)\ AND\ (37.0 \leqslant Gene[M23197\_at] \leqslant 395.0)\ THEN\ Class = ALL$	98.61%

Table 6
The best quantitative classification rules for lung dataset.

Number of rule items	QCR	Accuracy
1	$IF\ (424.7 \leqslant Gene[J02874\_at] \leqslant 2245.7)\ THEN\ Class = Tumor$	100%

Table 7
The best quantitative classification rules for SRBCT dataset.

Number of rule items	QCR	Accuracy
1	$IF\ (0.6426 \leqslant Gene[123] \leqslant 1.3896)\ THEN\ Class = NP$	100%
2	$IF\ (0.1103 \leqslant Gene[1003] \leqslant 0.4008)\ AND\ (0.1792 \leqslant Gene[972] \leqslant 1.1354)\ THEN\ Class = BL$	100%
2	$IF\ (4.201 \leqslant Gene[509] \leqslant 32.6601)\ AND\ (0.36 \leqslant Gene[1003] \leqslant 3.68)\ THEN\ Class = RMS$	100%
3	$IF\ (0.0564 \leqslant Gene[2] \leqslant 0.252)\ AND\ (0.2893 \leqslant Gene[1159] \leqslant 1.9317)\ AND\ (0.0097 \leqslant Gene[2050] \leqslant 0.5527)\ THEN\ Class = EWS$	100%

Table 8
The best quantitative classification rules for lymphoma dataset.

Number of rule items	QCR	Accuracy
1	$IF\ (1.24 \leqslant Gene[2368X] \leqslant 2.34)\ THEN\ Class = BCLL$	100%
2	$IF\ (0.26 \leqslant Gene[2405X] \leqslant 1.43)\ AND\ (0.96 \leqslant Gene[2108X] \leqslant 2.49)\ THEN\ Class = FL$	100%
1	$IF\ (0.04 \leqslant Gene[2368X] \leqslant 1.03)\ THEN\ Class = DLBCL$	96.96%

Table 9
The best quantitative classification rules for leukemia2 dataset.

Number of rule items	QCR	Accuracy
1	$IF\ (64 \leqslant Gene[X04145\_at] \leqslant 2608)\ THEN\ Class = ALL$	100%
2	$IF\ (8156 \leqslant Gene[X00274\_at] \leqslant 25969)\ AND\ (0.0 \leqslant Gene[M27891\_at] \leqslant 553.0)\ THEN\ Class = AML$	100%
2	$IF\ (163 \leqslant Gene[U46499\_at] \leqslant 3166)\ AND\ (110 \leqslant Gene[M31523\_at] \leqslant 670)\ THEN\ Class = MLL$	100%

the result reported by Golub et al. (1999), who presented that M27891, X04085, Y007876, and M63138 are down-regulated in ALL. Bijlani et al. (2003) showed that M27891 is in ALL discriminators; this is consistent with our findings. Moreover, Dudoit et al. (2002) presented that samples 66, 67, and 70 are the most difficult samples for many models to correctly classify by many models. Here, it is worth mentioning that *DQB* could correctly classify those samples and all other ALL and AML samples.

Table 6 presents the discovered quantitative rule from lung cancer; J02874_at is the most significant. This result is in agreement with the result generated by CFS-PART and CFS-OneR algorithms, but the *DQB* algorithm achieved the highest classification accuracy.

By analyzing the classification rules discovered from the SRBCT dataset, as shown in Table 7, we observed that gene 123 is the most significant gene in NB, while gene 1003 is a very predictive gene to classify BL and RMS samples; this is consistent with CFS-PART and CFS-OneR findings. In addition, from Table 8, we note that gene 2368X is a discriminator gene for lymphoma cancer, especially in the BCLL sample. For the leukemia2 dataset, as shown in Table 9, gene X04145 is the most significant for building a QCR that can classify all ALL samples correctly.

In order to prove the efficiency of the proposed *DLS* method, we evaluated and compared *DQB* classification performance using the original local search method and the *DLS* method. One of the most significant criteria in evaluating any new developed algorithm is the performance quality of the algorithm and its ability to generate and discover similar (identical) result when executed several times. This factor is very critical for meta-heuristics algorithms, which is the case in our work. To examine the robustness of the two approaches, in some instances, in all 30 runs, both algorithms managed to find the same answer or similar ones (not identical). The comparison results of running the *DQB* algorithm using the *DLS* method and the original local search method in terms of statistical results and reporting the best, worst, and average solutions found for the binary-class microarray datasets for colon, leukemia1, and lung are shown in Tables 10–12, respectively. Meanwhile, Tables 13–15, show the comparison results for multi-class microarray datasets for SRBCT, lymphoma, and leukemia2, respectively. From the point of view of the classification accuracy in all independent runs, the *DQB* algorithm using the DLS method obtained a better performance, although the difference with regard to the *DQB* algorithm using the original local search method (as shown in these tables) is significant in each single case (i.e., all cancer datasets using a different number of selected genes).

In this paper, we compared our proposed algorithm with two quantitative rule-based classification methods, OneR and PART algorithms. We implemented CFS with OneR (CFS-OneR) and CFS with PART (CFS-PART) in order to compare the performance of the *DQB* algorithm with the same parameters and datasets. We have also compared the performance of the *DQB* algorithm against our previously proposed classification algorithms, namely *ABC-SVM* (Alshamlan et al., 2016), *mRMR-ABC* (Alshamlan et al., 2015a), and *Genetic-ABC* (Alshamlan et al., 2015b). In addition, we compared our proposed algorithm with published results for recent bio-inspired meta-heuristic classification algorithms with respect to classification accuracy and simplicity (i.e., number of genes used in classification). Notably, all bio-inspired meta-heuristic approaches have been combined with the SVM as a classification approach.

In order to make the comparison between algorithms more robust,we applied statistical analysis to the results using Kruskal-Wallis (Kruskal and Wallis, 1952), which yields a p-values of 0.009, which indicates the significance of the shown results.

Computational complexity is an important aspect in algorithm assessment. Therefore, we tested and compared the time complexity of *DQB* with other quantitative rule-based classification methods, CFS-OneR and CFS-PART, and our proposed bio-inspired meta-heuristic algorithms, ABC-SVM, mRMR-ABC, and Genetic-ABC. Table 16 shows the average runtime in seconds for the *DQB* algorithm and other classification algorithms under comparison; the *DQB* algorithm has a faster execution time.

Table 10
The performance of the *DQB* algorithm with the proposed *DLS* method as compared to the original local search method for colon dataset.

Number of rule items	Classification accuracy for the DQB algorithm					
	Dynamic local search method			Original local search method		
	Best	Worst	Mean	Best	Worst	Mean
1	87.09%	85.48%	86.28%	72.58%	69.35%	70.96%
2	93.54%	91.93%	92.74.%	88.70%	85.48%	87.09%
3	98.38%	96.77%	97.58%	90.32%	85.48%	87.09%
4	100%	98.38%	99.19%	87.09%	82.25%	85.48%

Table 11
The performance of the *DQB* algorithm with the proposed *DLS* method as compared to the original local search method for leukemia1 dataset.

Number of rule items	Classification accuracy for the DQB algorithm					
	Dynamic local search method			Original local search method		
	Best	Worst	Mean	Best	Worst	Mean
1	95.83%	94.44%	95.13%	84.72%	80.55%	82.63%
2	100%	97.22%	98.61.%	84.72%	81.94%	83.33%
3	100%	100%	100%	91.66%	90.27%	90.96%

Table 12
The performance of the *DQB* algorithm with the proposed *DLS* method as compared to the original local search method for lung dataset.

Number of rule items	Classification accuracy for the DQB algorithm					
	Dynamic local search method			Original local search method		
	Best	Worst	Mean	Best	Worst	Mean
1	100%	98.95%	99.22%	93.75%	89.58%	90.22%
2	100%	100%	100%	98.95%	95.33%	96.87%
3	100%	100%	100%	100%	91.66%	97.91%

Table 13
The performance of the *DQB* algorithm with the proposed *DLS* method as compared to the original local search method for SRBCT dataset.

Number of rule items	Classification accuracy for the DQB algorithm					
	Dynamic local search method			Original local search method		
	Best	Worst	Mean	Best	Worst	Mean
1	100%	96.38%	98.79%	92.71%	73.99%	83.13%
2	100%	97.59%	99.8%	95.18%	80.72%	90.36%
3	100%	100%	100%	95.18%	81.92%	91.56%
4	100%	100%	100%	96.38%	85.54%	92.77%
5	100%	100%	100%	96.38%	79.51%	90.36%
6	100%	100%	100%	93.97%	78.31%	85.54%

Table 14
The performance of the *DQB* algorithm with the proposed *DLS* method as compared to the original local search method for lymphoma dataset.

Number of rule items	Classification accuracy for the DQB algorithm					
	Dynamic local search method			Original local search method		
	Best	Worst	Mean	Best	Worst	Mean
1	100%	92.42%	96.96%	92.42%	68.18%	89.39%
2	100%	90.90%	95.16%	92.42%	61.24%	86.36%
3	100%	90.90%	95.16%	87.87%	59.67%	74.19%
4	100%	90.90%	95.16%	86.36%	59.67%	72.58%

Table 15
The performance of the *DQB* algorithm with the proposed *DLS* method as compared to the original local search method for leukemai2 dataset.

Number of rule items	Classification accuracy for the DQB algorithm					
	Dynamic local search method			Original local search method		
	Best	Worst	Mean	Best	Worst	Mean
1	100%	95.83%	97.22%	91.66%	72.22%	83.33%
2	100%	98.61%	99.3%	93.05%	68.05%	80.55%
3	100%	100%	100%	91.66%	55%	69.44%
4	100%	100%	100%	90.27%	55%	72.22%

Table 16
Average runtime (in *s*) for the *DQB* algorithm and other classification algorithms.

Algorithms	Preprocessing time	Average classification time	Total
DQB	26.37 s	30.51 s	56.88 s
CFS + OneR	26.37 s	36.86 s	63.23 s
CFS + PART	26.37 s	40.22 s	66.59 s
mRMR-ABC with SVM (Alshamlan et al., 2015a)	25.17 s	72.13 s	97.3 s
Genetic-ABC with SVM (Alshamlan et al., 2015b)	25.17 s	90.26 s	115.43 s
ABC with SVM (Alshamlan et al., 2016)	0.0 s	134.74 s	134.74 s

Table 17
The classification accuracy performance of the related algorithms under comparison for binary-class cancer microarray datasets. Numbers in parentheses mean the numbers of selected genes.

Algorithms	Colon	Leukemia1	Lung
DQB	99.19(4)	100(3)	100(2)
CFS-PART	93.54(3)	97.22(2)	98.95(1)
CFS-OneR	88.70(1)	95.83(1)	98.95(1)
ABC-SVM (Alshamlan et al., 2016)	92.44(20)	91.89(5)	93.75(8)
mRMR-ABC (Alshamlan et al., 2015a)	93.60(15)	92.82(5)	98.53(7)
Genetic-ABC (Alshamlan et al., 2015b)	93.60(9)	96.43(5)	99.11(7)
CARSVM (Kianmehr and Alhajj, 2008)		95.83(4)	
Association rule-based (Wang and Gotoh, 2009)	90(2)		82(2)
BSTC (Iwen et al., 2008)		82.35 (866)	100 (2173)
PSO (Qi et al., 2007)	85.48(20)	94.44(23)	
PSO (Javad and Giveki, 2013)	87.01 (2000)	93.06 (7129)	
mRMR-PSO (Javad et al., 2012)	90.32(10)	100(18)	
GADP (Lee and Leu, 2011)	100(8)	100(5)	100(8)
mRMR-GA (Amine et al., 2009)			100(15)
IGA (Huang and Chang, 2007)		100(4)	95.75(7)
MLHD-GA (Huang et al., 2007)			97.1(10)
GA (Peng et al., 2003)	93.55(12)	100(6)	
mAnt (Yu et al., 2009)	91.5(8)	100(9)	

The first criterion used to evaluate the performance of the various classification algorithms is classification accuracy, which is defined in terms of the cross-validation accuracy rate, which in turn, equals the quotient between the number of test cases correctly classified and the total number of test cases. A LOOCV cross-validation was used to evaluate the classification performance of all algorithms. The other criterion for performance evaluation is the number of genes (i.e., *rule item* in our algorithm) that have been used for classification. In order to achieve a comprehensive and useful classification result, a minimum number of genes is required. Based on our study we urged that by increasing the rule item size the speed of convergence and the classification accuracy will and adhere to it.

Tables 17 and 18, summarizes the classification accuracy of the *DQB* algorithm and other classification algorithms under comparison for binary and multi-class microarray datasets, respectively. Each entry in the tables shows the mean value of the accuracy obtained via the cross-validation procedure followed by the number of selected genes used in classification between parentheses. An entry in the *DQB* column is shown in bold font; for the corresponding dataset, the accuracy achieved with *DQB* was significantly greater than the accuracy achieved with other classification algorithms under comparison.

For binary-class microarray datasets, as shown in Table 17, *DQB* achieved 100% classification accuracy for the colon cancer dataset using four *rule items* (genes). Meanwhile, for the leukemia1 dataset, it acquired 100% classification accuracy by using two rule items (genes), and for the lung cancer dataset, it achieved 100% classification accuracy using only one rule item (gene). And for the lung cancer dataset, *DQB* achieves 100% classification accuracy using only one *rule items* (genes).

Regarding the multi-class microarray datasets, as shown in Table 18, *DQB* achieved 100% classification accuracy for the SRBCT dataset using only one *rule items* (gene) to classify NP class samples, two rule items to classify BL and RMS class samples, and three rule items to classify EWS class samples. For the lymphoma dataset, it acquired 100% classification accuracy to classify BCLL and FL class samples by using one and two *rule items* (genes), respectively, and it achieved 96.96% classification accuracy to classify DLBCL class samples by using one gene. In addition, for the leukemia2 dataset, *DQB* achieved 100% classification accuracy using only one *rule item* to classify ALL class samples and two *rule items* to classify AML and MLL class samples.

The results show that the *DQB* algorithm was significantly more accurate and remained consistently superior to other bio-inspired meta-heuristic classification algorithms over all six benchmark microarray datasets. It can also be seen that the number of genes of rule length was reduced in our algorithm. This is because each individual rule is pruned based on a quality measure in every

Table 18
The classification accuracy performance of the related algorithms under comparison for multu-class cancer microarray datasets. Numbers in parentheses mean the numbers of selected genes.

Algorithms	SRBCT	Lymphoma	Leukemia2
DQB	**EWS:** 100(3) **NB:** 100(1) **BL:** 100(2) **RMS:** 100(3)	**DLBCL:** 100(1) **FL:** 100(2) **B-CLL:** 96.96(1)	**AML:** 100(1) **ALL:** 100(3) **MLL:** 100(3)
CFS-PART	**EWS:** 96.96(2) **NB:** 66.26(6) **BL:** 61(1) **RMS:** 96(2)	**DLBCL:** 91.30(1) **FL:** 93.54(3) **B-CLL:** 90.90(1)	**AML:** 85.71(1) **ALL:** 71.79(3) **MLL:** 84.72(3)
CFS-OneR	74.69(1)	96.77(1)	86.11(1)
ABC-SVM (Alshamlan et al., 2016)	87.99(6)	92.42(5)	90.27(8)
mRMR-ABC (Alshamlan et al., 2015a)	91.56(6)	95.69(5)	91.66(8)
GBC (Alshamlan et al., 2015b)	96.38(6)	96,96(5)	95.83(8)
GADP(Lee and Leu, 2011)	100(8)	100(6)	
mRMR-GA (Amine et al., 2009)		95(5)	
IGA (Huang and Chang, 2007)	98.75(6)		
MLHD-GA (Huang et al., 2007)	100(11)	100(6)	100(9)
CFS-IBPSO (Yang et al., 2008)		100(6)	98.57(41)
mAnt (Yu et al., 2009)		100(7)	
MIDClass (Giugno et al., 2013)	100(3)		

iteration. Therefore, we can conclude that the *DQB* algorithm is a promising approach for solving classification problems.

5.3. Conclusion

In this paper, we proposed a new quantitative rule-based classification model mining algorithm. It integrates quantitative rule mining and the ABC algorithm in order to provide users with more convenience in terms of understandability and interpretability. To the best of our knowledge, this is the first attempt that uses the ABC algorithm for mining a quantitative rule-based classifier. In addition, this is also the first attempt that uses a quantitative rule-based classification model for a classifying microarray cancer dataset.

The performance of the developed model has been compared with well-known quantitative rule-based classification methods and bio-inspired meta-heuristic classification algorithms using six binary and multi-class microarray datasets.

From the results, it can be concluded that a considerable increase in classification accuracy can be obtained when the quantitative rule-based (*DQB*) algorithm is integrated in the learning process. This confirms that these rules achieve a highly qualified knowledge extracted from the training dataset. All subsets of quantitative rules are reported to be close to 100% classification accuracy with a minimum number of genes. It is remarkable that several newly discovered genes emerged which have not been seen in past studies. It worthily noted that one of main limitation of DQB is number of pruning rules with some gene expression datasets. The DQB generate large number of rules when is it applied on non correlated gene expression profile. So, In future we need some improvement or preprocessing step to solve this issue.

In the context of applicability, according to the results obtained from the microarray gene expression analysis, we can conclude that *DQB* can also be adopted in a various real world problems with some modification.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

Hala Alshamlan conceived, designed, implemented, tested, analyses the results, and drafting of the manuscript, and critically revised the final manuscript.

Acknowledgements

This research project was supported by a grant from the Research Center of the Center for Female Scientific and Medical Colleges, Deanship of Scientific Research, King Saud University.

References

Al-maqaleh, B.M., Shahbazkia, H., 2012. Article: a genetic algorithm for discovering classification rules in data mining. Int. J. Comp. Appl. 41 (18), 40–44 (full text available).

Alizadeh, A., Eisen, M., Davis, M., Rosenwald, A., Boldrick, J., Sabet, T., Powell, Y., Yang, L., Marti, G., Moore, T., Hudson, J., Lu, L., Lewis, D., Tibshirani, R., Sherlock, G., Chan, W., Greiner, T., Weisenburger, D., Armitage, J., Warnke, R., Levy, R., Wilson, W., Grever, M., Byrd, J., Botstein, D., Brown, P., Staudt, L., 2000. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403 (6769), 503–511.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., Levine, A., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Nat. Acad. Sci. 96 (12), 6745–6750.

Alshamlan, H., Badr, G., Alohali, Y., 2015a. mRMR-ABC: a hybrid gene selection algorithm for cancer classification using microarray gene expression profiling. BioMed Res. Int.

Alshamlan, H.M., Badr, G.H., Alohali, Y., 2013. A study of cancer microarray gene expression profile: objectives and approaches. In: Proceedings of the World Congress on Engineering, vol. 2.

Alshamlan, H.M., Badr, G.H., Alohali, Y.A., 2015b. Genetic bee colony (gbc) algorithm: a new gene selection method for microarray cancer classification. Comput. Biol. Chem. 56, 49–60.

Alshamlan, H.M., Badr, G.H., Alohali, Y.A., 2016. Abc-svm: artificial bee colony and svm method for microarray gene selection and multi class cancer classification. Int. J. Mach. Learn. Comput. 6 (3), 184.

Amine, A., El Akadi, A., El Ouardighi, A., Aboutajdine, D., 2009. A new gene selection approach based on minimum redundancy-maximum relevance (mrmr) and genetic algorithm (ga). In: IEEE/ACS international conference on computer systems and applications, 2009. AICCSA 2009, pp. 69–75.

Armstrong, S.A., Staunton, J.E., Silverman, L.B., Pieters, R., den Boer, M.L., Minden, M. D., Sallan, S.E., Lander, E.S., Golub, T.R., Korsmeyer, S.J., 2001. Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature Genet. 30 (1), 41–47.

Beer, D.G., Kardia, S.L., Huang, C.-C., Giordano, T.J., Levin, A.M., Misek, D.E., Lin, L., Chen, G., Gharib, T.G., Thomas, D.G., et al., 2002. Gene-expression profiles predict survival of patients with lung adenocarcinoma. Nature Med. 8 (8), 816–824.

Bijlani, R., Cheng, Y., Pearce, D.A., Brooks, A.I., Ogihara, M., 2003. Prediction of biologically significant components from microarray data: Independently consistent expression discriminator (iced). Bioinformatics 19 (1), 62–70.

Celik, M., Karaboga, D., Koylu, F., 2011. Artificial bee colony data miner (abc-miner). In: 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA), June. pp. 96–100.

Dudoit, S., Fridlyand, J., Speed, T.P., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. J. Am. Statist. Assoc. 97 (457), 77–87.

Giugno, R., Pulvirenti, A., Cascione, L., Pigola, G., Ferro, A., 2013. Midclass: Microarray data classification by association rules and gene expression intervals. PLoS ONE 8 (8), e69873. 08.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, L., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286 (5439), 531–537.

Holden, N., Freitas, A.A., 2008. A hybrid pso/aco algorithm for discovering classification rules in data mining. J. Artif. Evol. App. (January), 2:1–2:11.

Huang, H.-L., Chang, F.-L., 2007. Esvm: evolutionary support vector machine for automatic feature selection and classification of microarray data. Biosystems 90 (2), 516–528.

Huang, H.-L., Lee, C.-C., Ho, S.-Y., 2007. Selecting a minimal number of relevant genes from microarray data to design accurate tissue classifiers. Biosystems 90 (1), 78–86.

Iwen, M., Lang, W., Patel, J., 2008. Scalable rule-based gene expression data classification. In: IEEE 24th International Conference on Data Engineering, 2008. ICDE 2008, pp. 1062–1071.

Jaiswal, A., Dubey, G., 2013. Identifying best association rules and their optimization using genetic algorithm. Int. J. Emerg. Sci. Eng. 1 (7), 91–95.

Javad, A.M., Giveki, D., 2013. Automatic detection of erythemato-squamous diseases using pso-svm based on association rules. Eng. Appl. Artif. Intell. 26 (1), 603–608.

Javad, A.M., Mohammad, H.S., Rezghi, M., 2012. A novel weighted support vector machine based on particle swarm optimization for gene selection and tumor classification. Comput. Math. Meth Med., 2012

Khan, J., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C., et al., 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Med. 7 (6), 673–679.

Kianmehr, K., Alhajj, R., 2008. CARSVM: a class association rule-based classification framework and its application to gene expression data. Artif. Intell. Med. 44, 7–25.

Kruskal, W.H., Wallis, W.A., 1952. Use of ranks in one-criterion variance analysis. J. Am. Statist. Assoc. 47 (260), 583–621.

Lee, C.-P., Leu, Y., 2011. A novel hybrid feature selection method for microarray data analysis. Appl. Soft Comput. 11 (1), 208–213.

Liu, Y., Qin, Z., Shi, Z., Chen, J., 2004a. Rule Discovery with Particle Swarm Optimization. Springer Berlin Heidelberg, pp. 291–296. https://doi.org/10.1007/978-3-540-30483-8_35.

Liu, Y., Qin, Z., Shi, Z., Chen, J., 2004b. Rule discovery with particle swarm optimization. Content Comput., 291–296.

Ma, B.L.W.H.Y., Liu, B., 1998. Integrating classification and association rule mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining.

Ng, A.Y., 1997. Preventing overfitting of cross-validation data. In: ICML, vol. 97. pp. 245–253.

Otero, F., Freitas, A., Johnson, C., 2008. cant-miner: an ant colony classification algorithm to cope with continuous attributes. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A. (Eds.), Ant Colony Optimization and Swarm Intelligence of Lecture Notes in Computer Science, vol. 5217. Springer Berlin Heidelberg, pp. 48–59. https://doi.org/10.1007/978-3-540-87527-7_5.

Parpinelli, R.S., Lopes, H.S., Freitas, A.A., 2002a. An ant colony algorithm for classification rule discovery. Data mining: a heuristic approach 208, 191–132.

Parpinelli, R.S., Lopes, H.S., Freitas, A.A., 2002b. Data mining with an ant colony optimization algorithm. IEEE Trans. Evolution. Comput. 6 (4), 321–332.

Peng, S., Xu, Q., Ling, X.B., Peng, X., Du, W., Chen, L., 2003. Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. FEBS Lett. 555 (2), 358–362.

Qi, S., Shi, W.-M., Wei, K., Ye, B.-X., 2007. A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification. Adv. Comp. Sci. 71 (4), 157–162.

Salleb-aouissi, A., Vrain, C., Nortet, C., 2013. Quantminer: a genetic algorithm for mining quantitative association rules. J. Mach. Learn. Res., 3153–3157.

Shukran, M.A.M., Chung, Y.Y., Yeh, W.-C., Wahid, N., Zaidi, A.M.A., 2011. Artificial bee colony based data mining algorithms for classification tasks. Mod. Appl. Sci. 5 (4), p217.

Sousa, T., Silva, A., Neves, A., 2004. Particle swarm based data mining algorithms for classification tasks. Paral. Comput. 30 (5?6), 767–783 (parallel and nature-inspired computational paradigms and applications).

Talebi, M., Abadi, M., 2014. Beeminer: a novel artificial bee colony algorithm for classification rule discovery. In: 2014 Iranian Conference on Intelligent Systems (ICIS), February, pp. 1–5.

University of Waikato, N.Z., 1997. Waikato Environment for Knowledge Analysis <http://www.cs.waikato.ac.nz/ml/weka/downloading.html> (accessed: 2014-06-12).

Wang, X., Gotoh, O., 2009. Microarray-based cancer prediction using soft computing approach. Cancer Informat. 7, 123–139.

Yan, X., Zhang, C., Zhang, S., 2009. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. Expert Syst. Appl. 36 (2), 3066–3076.

Yang, C.-S., Chuang, L.-Y., Ke, C.-H., Yang, C.-H., 2008. A hybrid feature selection method for microarray classification. Int. J. Comp. Sci. 35, 285–290.

Yu, H., Gu, G., Liu, H., Shen, J., Zhao, J., 2009. A modified ant colony optimization algorithm for tumor marker gene selection. Genom., Proteom. Bioinformat. 7 (4), 200–208.

Yvan, S., aki, I., Pedro, L., 2007. A review of feature selection techniques in bioinformatics. Bioinformatics 23 (19), 2507–2517.