Gene expression

A genetic programming-based approach to the classification of multiclass microarray datasets
Kun-Hong Liu1,2,3,∗,† and Chun-Gui Xu2,4,†

1School of Software, Xiamen University, Xiamen, Fujian, 361005, china, 2Intelligent Computing Lab, Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, P.O. Box 1130, Hefei, Anhui, 230031, China, 3Department of Automation and 4School of Life Science, University of Science and Technology of China, Hefei, Anhui, 230026, China

Received on March 23, 2008; revised on November 14, 2008; accepted on December 11, 2008

Associate Editor: David Rocke

ABSTRACT

Motivation: Feature selection approaches have been widely applied to deal with the small sample size problem in the analysis of microarray datasets. For the multiclass problem, the proposed methods are based on the idea of selecting a gene subset to distinguish all classes. However, it will be more effective to solve a multiclass problem by splitting it into a set of two-class problems and solving each problem with a respective classification system.

Results: We propose a genetic programming (GP)-based approach to analyze multiclass microarray datasets. Unlike the traditional GP, the individual proposed in this article consists of a set of small-scale ensembles, named as sub-ensemble (denoted by SE). Each SE consists of a set of trees. In application, a multiclass problem is divided into a set of two-class problems, each of which is tackled by a SE first. The SEs tackling the respective two-class problems are combined to construct a GP individual, so each individual can deal with a multiclass problem directly. Effective methods are proposed to solve the problems arising in the fusion of SEs, and a greedy algorithm is designed to keep high diversity in SEs. This GP is tested in five datasets. The results show that the proposed method effectively implements the feature selection and classification tasks.

Contact: lhqz@163.com; khliu1977@gmail.com

Supplementary information: Supplementary data are available at Bioinformatics online.

1 INTRODUCTION

With the development of microarray technology, it is possible to diagnose and classify some particular cancers directly based on DNA microarray datasets. Up to now, more and more new prediction, classification and clustering techniques have been used for analysis of the data, and these efforts lead us to a more complete understanding of the molecular variations among tumors (Dettling, 2004; Golub et al., 1999; Li et al., 2004; Pochet et al., 2004). However, one feature of the microarray dataset is that the number of samples collected tends to be much smaller than the number of genes. That is, the number for the former tends to be on the order of tens or hundreds, while microarray datasets typically contain thousands of genes on each chip. The high dimensionality of microarray datasets unavoidably worsens the generalization ability of machine learning methods. At the same time, a large number of genes do not contribute to cancer diagnosis. A widely deployed solution for this problem is the feature selection approach, with which a set of biologically significant genes will be found and used to improve the cancer diagnosis accuracy (Choudhary et al., 2006).

The evolutionary algorithm (EA)-based feature selection methods achieve better performances compared with other methods (Kudo and Sklansky, 2000). The search mechanism imbedded in EA is powerful and can lead it to an optimal solution. So far many researchers have proposed different evolutionary-based methods for the analysis of microarray dataset with great success (Lin et al., 2006; Yu et al., 2007).

Genetic programming (GP) has been widely applied to deal with classification problems because it can discover underlying data relationships. Among the evolutionary approaches, GP is a promising solution for the discovery of potentially important gene by generating comprehensible rules for classification (Hong and Cho, 2006; Langdon and Buxton, 2004; Yu et al., 2007). GP-based methods have been successfully applied to analyze two-class microarray datasets because a typical individual in traditional GP is a tree, which can produce a ‘yes/no’ answer for a classification problem. But these methods cannot solve multiclass problems directly due to the limit of individual structure. And to the best of our knowledge, no effective GP-based approaches have been proposed to analyze multiclass microarray datasets.

In this article, we focus the discussion on the analysis of multiclass microarray datasets. When dealing with a multiclass problem, most of methods use EA to find an optimal gene subset, which is then used to classify all classes (Liu et al., 2005; Ooi and Tan, 2003). However, when there are a large number of classes along with a small sample size in each class, it is difficult to obtain an optimal gene subset. In contrast, it is easier to discover key genes for discriminating one class from others. So we design a new GP individual structure to solve the classification task for multiclass microarray datasets. In application, a multiclass problem is split to multiple two-class problems, and each problem is tackled by a respective ensemble system, which consists of a set of trees. Then the ensemble systems are fused to construct an individual. In this way, this new individual
structure can be used to represent a multiclass classifier. And we also propose effective methods to solve the problems arising in the fusion of multiple ensemble systems, and design a greedy algorithm to maintain the diversity among the ensembles. The experimental results show that this GP-based scheme can tackle the classification and feature selection tasks successfully.

2 METHODS

GP was introduced by Koza (1992). GP is a branch of genetic algorithm (GA), and the main difference between GP and GA is the structure of individuals: GA has string-structured individuals, while GP’s individuals are trees. Due to the structure, GP can produce classification rules by formulating important features.

Generally the terminal set consists of features and constants, and the function set consists of arithmetical or logical functions. The leaf nodes and non-leaf nodes of trees are chosen from the terminal and function sets, respectively. Let $F$ be the set of functions, and $T$ be the set of terminals. The trees built in the evolution process are the set of all possible compositions of functions and terminals selected from $F$ and $T$. When used for the classification task, a tree is evolved from a training dataset and validated against an independent test dataset. Take the third rule for distinguishing colon cancer on the NCI60 dataset as an example, which is listed in Table S2 in the Supplementary Material 1. This rule is: ‘if times(X203,X942)<0.296 then colon cancer’. Here, the serial number of a gene is indicated by the letter ‘X’ followed by a number, so X203 and X942 refers to the 203th and 942th gene in the NCI60 dataset, respectively. The ‘if’ clause is generated by the GP that is, the function ‘times’, two genes and the constant 0.296 are selected by GP. The target class is predefined as the colon cancer. According to the function settings shown in Table 1, function ‘times’ performs multiplication on the expression levels of these two genes. Only when the result is smaller than 0.296, the tree produces an answer ‘yes’ (+1) and assigns this sample to the colon cancer; otherwise, it produces ‘no’ (−1). All trees can be comprehended in this way, regardless of their depths. As a tree can only produce a ‘yes/no’ answer, the traditional GP cannot tackle a multiclass problem directly.

So far, some authors proposed different methods to apply GP to multiclass problems (Bojarczuk et al., 2004; Kishore et al., 2000; Muni et al., 2004). In these papers, a multiclass problem is split into multiple two-class problems, each of which is handled by a tree. In details, for the $i$-th two-class problem, all training samples are divided into two groups, denoted by $G_{i1}$ and $G_{i2}$. If a sample $X$ belongs to the $i$-th class, then $X \in G_{i1}$; otherwise, $X \in G_{i2}$. In this way, the classification of $n$-class problem is divided into $n$ two-class problems, and the $i$-th problem is the classification task of separating the $i$-th class from all other classes. A tree is used to distinguish the two respective sample groups, so a multiclass classifier contains $n$ trees for an $n$-class problem. These proposed schemes work well for many classification tasks, but they are not effective for the microarray dataset problem. Due to the small sample size problem, it is hard to obtain an accurate tree for each two-class problem. Consequently, the combination of trees may lead to poor performance avoidably.

In addition, some problems will arise in the fusion of trees. Assuming there are equal samples in each class. After a multiclass problem is divided into $n$ two-class problems, there are $S_{j1}$ samples in $G_{ij1}$, and $S_{j2}$ samples in $G_{ij2}$ for the $j$-th two-class problem, which makes $S_{j1}/S_{j2} = 1/(n-1)$. As the sample size in the second group is larger than the first group, the classification task is accompanied by the unbalanced data problem. The larger the $n$ is, the more skewed the dataset is. What is more, after obtaining $n$ trees for an $n$-class problem, a solution is required to fuse their outputs. For a new sample, if only the $i$-th tree returns +1 and others return −1, then this sample will be assigned to the $i$-th class. However, two situations should be taken into consideration. First, in the case of data distribution overlapping, two or more trees will return +1. Then it is impossible to determine to which class the input sample should be assigned. Second, it is possible that all trees return −1. Then none of trees recognizes the sample as its class, and the corresponding sample will be rejected. However, it is still reasonable because some samples may not follow the data distribution of their group in a feature subspace, especially when the training sample size is quite small. These two cases prevent the classification system from achieving high accuracies, and result in a slowdown of evolutionary speed. Although some methods have been proposed to solve these problems, such as strength association (SA) measure along with heuristic algorithms (Kishore et al., 2000) or $Z$-value score (Chien et al., 2004), they all require a large number of training samples for the reliability estimation of each classifier. So they are not applicable to microarray datasets. Here, we propose some new methods to tackle these problems with taking the small sample size of microarray datasets into consideration. The framework of the proposed GP is described as follows.

2.1 The structure of individuals

An ensemble system has been proved to be more accurate and robust than an excellent single classifier in many fields (Kuncheva, 2004). As the output of an ensemble is based on all trees in the ensemble, when a new tree fails to distinguish a ‘hard’ sample, other trees in the ensemble still have a chance to correct it. Then the final ensemble can produce a correct output. So instead of applying a tree to a two-class problem, an ensemble of $k$ trees is deployed in this study. For an $n$-class microarray dataset, $n$ ensembles are needed to solve the respective two-class problems. Based on this consideration, a new individual structure for GP is proposed, as illustrated in Figure 1. In this scheme, an individual is a multiclass classifier and can deal with a

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal set ($T$)</td>
<td>All gene expression values and constant values</td>
</tr>
<tr>
<td>Function set ($F$)</td>
<td>Boolean and floating point operators: gt (&gt;), le (&lt;), times (*), minus (−), plus (+), max, min.</td>
</tr>
<tr>
<td>Maximum generation</td>
<td>100</td>
</tr>
<tr>
<td>Population size</td>
<td>500</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.7</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.5</td>
</tr>
<tr>
<td>Termination criteria</td>
<td>Fitness score reaches 1 or running 100 generations</td>
</tr>
<tr>
<td>Dynamic maximum tree</td>
<td>5</td>
</tr>
<tr>
<td>depth limit</td>
<td>10</td>
</tr>
</tbody>
</table>

![Fig. 1. The new individual structure for the GP](https://academic.oup.com/bioinformatics/article-abstract/25/3/331/244785/327x103-to-530x239)
multiclass problem directly. In an individual, there are \( n \) ensemble systems, which are named as sub-ensemble (SE) systems for clarifying their roles. As an individual is composed of \( n \times k \) trees in all, the size of SEs should not be too large for building an efficient and compact classifier.

Due to the small training sample size, 10-fold cross-validation (CV) is applied to evaluate the generalization ability of trees in each SE system. It was proved that the weighted majority vote is more effective than majority vote (Kuncheva, 2004). So here, CV accuracies are used to calculate trees’ weights, and the output of a SE is based on the weighted majority vote of \( k \) trees in it. Assuming that for the \( m \)-th individual in a population, the \( i \)-th SE is used to deal with the \( i \)-th two-class problem, denoted by \( SE_i \). Assuming that tree \( a \) in \( SE_i \) is denoted by \( T_{a,i} \), and its 10-fold CV accuracy is \( Ac_{a,m}^i \). Its weight \( w_{ai}^m \) is calculated by \( w_{ai}^m = Ac_{a,m}^i / \sum_{p=1}^{m} Ac_{p,m}^i \). Let its output be \( O_{a,i,m} \), and the final output of \( SE_i \) is \( O_{i,m} = \sum_{p=1}^{m} w_{ai}^m O_{a,i,m}^p \). For each sample, only when \( O_{i,m} \) is larger than 0, \( SE_i \) claims that it belongs to the \( i \)-th class.

We design some effective methods to address the problems in the fusion of SEs. As proved in Tang et al. (2006), when ‘hard’ samples are classified correctly by as many base classifiers as possible, an ensemble system can achieve high generalization ability. Due to the data unbalance problem, the number of samples in \( G \) is much smaller than that in \( G_2 \) in the \( i \)-th two-class problem. Then it is more difficult to correctly classify a sample in \( G_1 \) than in \( G_2 \). So we propose a weight assignment scheme to encourage SEs to distinguish more samples in the first group. For the \( i \)-th problem, the weights are:

\[
W_i = \frac{S_i}{S_i + S_2} \quad \text{and} \quad W_2 = \frac{S_2}{S_i + S_2}
\]

(1)

Then a covering score is designed to evaluate the generalization ability of each SE. If \( SE_i \) correctly classifies \( l_1 \) samples in \( G_1 \) and \( l_2 \) samples in \( G_2 \), its covering score is:

\[
C_m^i = \left( W_1 \times l_1 + W_2 \times l_2 \right) / \left( W_1 \times S_i + W_2 \times S_2 \right)
\]

(2)

When a SE gets a high covering score, its generalization ability can be guaranteed. As the generalization ability of an individual is directly connected with the performance of all SEs in it, it is necessary to guide SEs to evolve towards a balanced covering in respective two-class problems. So for the \( m \)-th individual, the fitness value \( F_m \) is designed as:

\[
F_m = \frac{\sum_{i=1}^{n} C_i}{n}
\]

(3)

With this design, an individual’s fitness value can indicate its performance, and the individual with high fitness value performs better. Due to the huge search space, only the individuals with top fitness values are kept so as to accelerate the evaluation process. When two or more individuals get the same fitness score, the one containing fewer features will be selected first.

When some SEs in an individual positively classify a sample at the same time, the conflicting situation occurs. Then the covering scores of these SEs will be compared, and only the output of the SE obtaining the highest score will be selected as the final decision.

2.2 The initialization of GP

The Ramped Half-and-Half method proposed in Koza (1992) is applied here to generate the first generation. In this procedure, an equal number of trees are initialized for each depth between 2 and the initial maximum tree depth value. For each depth level considered, half of the trees receive non-leaf nodes from \( F \) until trees are fully grown; the other half are allowed to receive nodes from both \( F \) and \( T \) randomly except for the root node, which produces a set of heavily unbalanced trees. Then this method results in balanced and unbalanced trees with several different depths.

Bloat means that trees keep growing without the corresponding improvements in fitness. As it makes the final results complex without any benefit, the dynamic maximum tree depth technique (Silva and Almeida, 2003) is used to control it here. When using this technique, there are two important parameters: strict depth limit and dynamic maximum tree depth limit. The depth of tree is initially set to be no deeper than the dynamic depth limit. If a tree does not exceed the dynamic maximum depth, it participates in the current population. When a tree is deeper than the dynamic maximum depth but does not exceed the strict maximum depth, it is evaluated by 10-fold CV. In the case that it is the best tree for the respective two-class problem currently, the dynamic maximum depth is increased and the new tree is allowed to join the population; otherwise, the new tree is rejected. If a tree is deeper than the strict maximum depth, it is rejected and one of its parents enters the population. Once the dynamic maximum depth is increased, it would never be lowered again. The initial maximum dynamic depth limit is set to be small so as to force the GP to look for simpler trees first before accepting complex solutions.

An effective SE consists of accurate and diverse trees. That is, the trees employed in a SE should be of high classification accuracy and avoid making coincident errors. In this way, the fused outputs can be more accurate than that of the best tree. And no gains will be achieved when fusing trees producing the same outputs. Many different diversity measures were proposed based on different theories, and most of them are based on the difference among the classifier outputs (Kuncheva, 2004). Here, we define a new diversity measure based on the difference in the feature subsets among trees, named as diversity in features (DIF).

**Definition.** Assuming that two trees, \( T_{a,m}^a \) and \( T_{b,m}^b \), are in \( SE_m \). Their feature subsets are denoted by \( f_{a,m}^a \) and \( f_{b,m}^b \), respectively. Then the \( p \)-th feature in \( T_{a,m}^a \) (denoted by \( j_{a,m}^a \)), \( j_{a,m}^a \in f_{a,m}^a \), is not included in \( f_{b,m}^b \), DIF for this tree-pair scores 1, or 0 otherwise. As the size of \( f_{a,m}^a \) is different from that of \( f_{b,m}^b \), usually, it is necessary to take both feature sets into consideration at the same time. Let the normalized DIF in this tree-pair be denoted by \( DIF_{a,b} \), then it is:

\[
DIF_{a,b} = \frac{\sum_{p=1}^{m} \left( s_{a,m}^p \neq s_{b,m}^p \right)}{s_a + s_b}
\]

According to (4), when \( DIF_{a,b} \) equals 1, these two trees contain completely different features, which guarantees high diversity between them. Let the sum of DIF for all tree-pairs including \( T_{a,m}^a \) in \( SE_m \) be denoted by \( S_{DIF}(T_{a,m}^a) \), which is calculated by

\[
S_{DIF}(T_{a,m}^a) = \sum_{p=1, p \neq a}^{m} DIF_{a,p}
\]

(5)

Then for \( T_{a,m}^a \), \( S_{DIF}(T_{a,m}^a) \) is a proper estimator of its contribution to the diversity in \( SE_m \). When \( S_{DIF}(T_{a,m}^a) \) reaches the highest score, \( k - 1 \), \( T_{a,m}^a \) is different from all other trees in \( SE_m \). Furthermore, when all trees in a SE get the highest score, this SE is composed of totally different trees. In this case, if all the trees in a SE are accurate enough, the diversity among trees guarantees a good performance of this SE. Let \( S_{DIF}(SE_m) \) represent the overall DIF scores in \( SE_m \), and is calculated by

\[
S_{DIF}(SE_m) = \frac{\sum_{a=1}^{k} S_{DIF}(T_{a,m}^a)}{2}
\]

(6)

When the maximum value of \( S_{DIF}(SE_m) \) is equal to \( (k^2 - k)/2 \), the highest diversity in \( SE_m \) is achieved. But even when some features are shared among different trees in a SE, the trees will still be diverse if different functions are deployed to combine the features. So it is unnecessary to always construct SEs with trees using completely different features. In addition, if important genes are permitted to appear more than once in a SE, the genes with great biological significance will be given more chances to be selected, and then the appearance frequency of a gene will more closely connect with its importance. So, to achieve a tradeoff, the threshold is set to 0.4 \( \times (k^2 - k) \) here, which allows a small part of features to be shared among different trees.
For $i=1$ to $n$
    Put all SEs for the $i$-th two-class problem into the exchange-list;
    For $m=1$ to $P$
        If $S_{Im} (SE_{Im}^m)$ satisfies the threshold
            Delete $SE_{Im}^m$ from the exchange-list;
        End
    End
    While $E_i > 2$
        $E_i = E_i - 1$
        For $m=1$ to $E_i$
            Find $T_{i}^{m\uparrow}$ with the smallest $S_{Im}^{m\uparrow}$ in $SE_{Im}^m$;
            For $q=1$ to $E_i$
                Find $T_{i}^{q\downarrow}$ with the smallest $S_{Im}^{q\downarrow}$ in $SE_{Im}^q$;
                If after exchanging $T_{i}^{q\downarrow}$ with $T_{i}^{m\uparrow}$, $S_{Im}^{q\downarrow} (SE_{Im}^q)$ or $S_{Im}^{m\uparrow} (SE_{Im}^m)$ satisfies the threshold
                    Exchange $T_{i}^{q\downarrow}$ with $T_{i}^{m\uparrow}$;
                    Delete $SE_{Im}^q$ from the exchange-list;
                    $E_i = E_i - 1$;
                End
            End
            If $S_{Im}^{m\uparrow} (SE_{Im}^m)$ satisfies the threshold
                Delete $SE_{Im}^m$ from the exchange-list;
                $E_i = E_i - 1$;
            End
        End
        Break; % End of this loop
    End
    End
    If $E_i = E_i$; % no exchange occurs in this loop
    Break; % so it is unnecessary to run some more loops
End
End

Fig. 2. Heuristic Algorithm I.

in a SE. For $SE_{Im}^m$, if $S_{Im} (SE_{Im}^m)$ is higher than this threshold, it is regarded as a diverse one.

This diversity measure is used to drive the GP to evaluate more genes for each two-class problem in the evolution. So the frequently selected genes are of greater importance compared with those seldom selected. And it should be noted that the evaluation of diversity is only carried out among the trees in a SE. And different SEs in an individual could contain the same feature subset without any constraint. In this way, when a gene is involved in different disease processes, it may appear in respective SEs.

Assuming that there are $P_1$ individuals in each generation, and then $P_1 \times n \times k$ trees will be randomly generated after initialization. In the first generation, the generated trees are randomly assigned to different SEs, so the diversity among SEs may not be greater than the threshold. Then it is necessary to adjust the assignment of trees among different SEs before evaluating individuals. So the Heuristic Algorithm I, a greedy algorithm for local improvement, is proposed to execute this adjustment process, as shown in Figure 2.

The Heuristic Algorithm I tries to maximize the diversity in all individuals by adjusting trees among different SEs. The tree dealing with the $i$-th two-class problem cannot be applied to the $j$-th problem because of different data distributions in different problems. So the exchange of trees should be limited between the SEs tackling the same two-class problem in different individuals. First, this algorithm extracts the SEs dealing with the same two-class problem from different individuals, and puts them into the exchange list. Then the SEs satisfying the threshold are deleted from the list. For the remaining SEs, the algorithm searches the trees with the least contributions to the diversity in respective SEs, and exchanges them if the exchange can improve the diversity of the corresponding SEs. The SEs satisfying the threshold is/are deleted from the list. When the number of SEs in the list is not larger than 2, or no exchange occurs in a loop, this algorithm stops because no gains will be obtained in the following loops. When the algorithm stops, there are more SEs satisfying the threshold. So the Heuristic Algorithm I helps to boost diversity among individuals.

The rejection case occurs only when all SEs in an individual cannot recognize a sample as its own class. If each SE contains diverse trees, the probability of the rejection case will be very small even when the size of SE is small. Assuming that the rejection case occurs in the probability $P_r$ when using a tree for a two-class problem. Assuming that $k$ trees in a SE are completely independent. Then only when more than $\lceil k/2 \rceil$ trees in a SE cannot recognize a sample, the tree rejects it. So the probability of rejection case is $P_r^{\lceil k/2 \rceil}$ for a SE, which is far smaller compared with $P_r$ even when the ensemble scale is only set to 3. As we apply the Heuristic algorithm I to keep trees in a SE as independent as possible, the rejection case occurs in a small probability, and is simply ignored here.

2.3 The genetic operators

The crossover and mutation operators should be operated on trees instead of individuals or SEs. So before applying the operators, a selection process is needed to pick up trees first. In detail, to select the trees for the $i$-th two-class problem, the first step is to select an individual from the current population in a probability proportional to the individual’s fitness value. Then, the next step is to select a tree in a probability proportional to the tree’s CV accuracy rate in the $i$-th SE of the selected individual. Both steps are based on the principle of roulette, which allows all trees in the current population getting a chance. Two/one trees will be selected as parent(s) for crossover/mutation.

After the selection process, the standard crossover and mutation operators are deployed. In the crossover process, two random nodes are chosen from both parents, and then the respective branches are swapped to create two offspring. The mutation operator randomly chooses a node from a parent tree and replaces it with a new randomly generated sub-tree. In our GP, the newly generated sub-tree only contains the features excluded in the parent tree. In this way, more features will be evaluated in the evolution. As the task of the GP is to discover potentially important genes in a huge number of candidates, the mutation operator is important to enhance the exploration in the huge search space.

For each offspring, $k$ trees will be generated in each SE using crossover and mutation operators. However, due to the random mechanism, the newly generated SEs in new offspring may not always be diverse enough. So the Heuristic Algorithm I is also applied to adjust the tree assignment for the new individuals, just as in the initialization process.

2.4 The validation of classifier

A classifier is validated in an independent test set. The validation of an individual’s performance takes three steps. First, a new sample will be checked by all trees in an individual. The trees classify a sample using the procedure described above. Second, the trees’ outputs are combined to form the respective SEs’ outputs based on the weighted majority vote. Here, the weights of SEs’ outputs are indicated by $+1/−1$. Finally, the respective SEs’ outputs are fused to produce the final results for an individual. If only a SE returns $+1$, the sample will be assigned to the respective class. When the conflicting situation occurs, the covering scores of the conflicted SEs are compared, and only the output of the SE with the highest score will be chosen as the final decision.

As there are many feature subsets generated in parallel during evolution, a set of globally optimal or at least near optimal trees can be obtained for each two-class problem. What is more, owing to the Heuristic Algorithm I, high diversity in SEs can be easily achieved, so the GP is promising in obtaining predictive classifiers.
3 RESULTS

Five microarray datasets are deployed here: NCI60 (Ross et al., 2000); lung (Bhattacharjee et al., 2001); prostate (Dhanasekaran et al., 2001); small round blue cell tumors (SRBCT) of childhood (Khan et al., 2001) and leukemia (Golub et al., 1999).

The original NCI60 dataset is composed of 68 samples and 9712 genes. A widely adopted preprocessing method is to exclude some categories with few samples, and to keep a smaller gene subset (Lin et al., 2006; Liu et al., 2005; Ooi and Tan, 2003). In the experiments, the samples belonging to the unknown, normal breast, lymph node and prostate cancers are excluded, and only the genes with high SD are kept. So only the truncated data with 61 samples and 1000 genes are used, and the preprocessing is exactly identical as described in Liu et al. (2005).

The original lung cancer dataset contains expression values of 12 600 genes in 203 samples. A SD threshold of 50 expression units is used to select the 3312 most variable transcripts sequences from the original dataset. The prostate cancer dataset contains 14 benign prostatic hyperplasia, 14 localized prostate cancer and 20 metastatic prostate cancer cases after excluding two categories with six samples. Each of the missing values of the dataset is substituted with the average of existing values of the respective gene in the category to which the sample belongs to. Then 1000 genes with the highest variations in these samples are selected. The preprocessing of the remaining two datasets are the same: transforming the raw data to natural logarithmic values, and then all of the 2308 and 7129 genes of the remaining two datasets are the same: transforming the raw data to natural logarithmic values, and then all of the 2308 and 7129 genes are kept for evaluation, respectively. After these steps, these four datasets are standardized to have zero mean and unit variation. The details of all datasets are given in Table S1 in the Supplementary Material 1.

In all experiments, we run GP 10 times with random initialization, and the parameter settings of the GP are listed in Table 1. Here, the initial maximum dynamic depth limit is set to 5 to force the GP to search simple trees at the first stage. And the strict depth limit is set to 10 to restrict the complexity of the generated trees. The parameters for all datasets are the same in our experiments.

In Table 2, we list the average of the best results obtained by the GP in each run with setting the size of SE to 3. Other tree-based classification systems, including CART (Breiman et al., 1984), decision forest (DF) (Ho, 1998) and random forests (RF) (Breiman, 2001) are also employed for comparisons. CART is provided in the Statistics Toolbox in Matlab 7.5. DF and RF are two widely deployed ensemble systems. Here, CART is deployed as the base classifier for them. In all experiments, the ensemble sizes for DF and RF are set to 30, and the corresponding results are based on 10 independent runs. For evaluation of the performance of different classification systems, the runs on independent test datasets and 10-fold CV on entire data are listed in Table 2.

From Table 2, it can be found that the results of CART are the worst usually. CART selects key features to build a tree based on an exhaustive search in the whole feature space. When the feature dimension of a dataset is huge, such as the leukemia dataset, it cannot search the optimal feature subsets for classification. As its learning algorithm lacks the ability to search global optimal results, it cannot perform well in experiments. RF and DF can achieve better results compared with CART. It is consistent with the observation that the ensemble system performs better than a single classifier when the classification accuracies obtained by base classifiers are higher than 50%. But when the base classifiers cannot satisfy this condition, the ensemble of these classifiers will not benefit the final results. So when dealing with the NCI60 and leukemia datasets, neither DF nor RF has a good performance. In general, RF can produce higher accuracies compared with DF.

On the contrary, the classifiers generated by our GP are much more effective compared with other methods. From Table 2, it can be found that the GP produces the highest average accuracies and the lowest SDs in most cases, and always achieve the best results. The success of the GP lies in the powerful search mechanism embedded in the evolutionary process, which can pick up global optimal or at least near optimal feature subsets to construct trees. In addition, the method of dividing a multiclass problem into multiple two-class problems reduces the difficulty in the search of optimal gene subsets.

Among all datasets, NCI60 is the hardest one to deal with. Using the traditional classification models or even ensemble systems, for example, diagonal linear discriminant analysis (DLDA) and bagging, the mean classification accuracies on the NCI60 dataset are lower than 50% even with only six classes kept (Lee et al., 2005). EAs have been widely deployed to tackle the NCI60 dataset analysis. Due to the small sample size in each class, some authors did not split the dataset into training and test datasets, and only tested the methods on the whole dataset using Leave-One-Out CV (LOOCV) to evaluate the performance of their methods (Jirapech-Umpai and Aitken, 2005; Liu et al., 2005). For the NCI60 dataset, the results obtained by other researchers are listed in Table 3, which shows that our method takes obvious advantages in both independent test set accuracy and the LOOCV accuracy. Although in Ooi et al. (2003) and Lin et al. (2006), the preprocessing methods for the NCI60

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Method</th>
<th>A1(%)</th>
<th>A2 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Amean</td>
<td>Asd</td>
</tr>
<tr>
<td>NCI60</td>
<td>GP</td>
<td>84.76</td>
<td>3.37</td>
</tr>
<tr>
<td></td>
<td>CART</td>
<td>47.54</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>DF</td>
<td>54.76</td>
<td>2.21</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>58.69</td>
<td>3.26</td>
</tr>
<tr>
<td>SRBCT</td>
<td>GP</td>
<td>99.64</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td>CART</td>
<td>79.52</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>DF</td>
<td>89.52</td>
<td>1.51</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>94.70</td>
<td>1.72</td>
</tr>
<tr>
<td>Leukemia</td>
<td>GP</td>
<td>99.44</td>
<td>0.97</td>
</tr>
<tr>
<td></td>
<td>CART</td>
<td>50.00</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>DF</td>
<td>50.56</td>
<td>1.49</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>52.78</td>
<td>2.50</td>
</tr>
<tr>
<td>Lung</td>
<td>GP</td>
<td>99.80</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>CART</td>
<td>85.71</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>DF</td>
<td>93.68</td>
<td>1.48</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>95.14</td>
<td>0.53</td>
</tr>
<tr>
<td>Prostate</td>
<td>GP</td>
<td>100.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>CART</td>
<td>97.92</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>DF</td>
<td>98.80</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>99.38</td>
<td>1.01</td>
</tr>
</tbody>
</table>

Here, A1 represents the results performing the 10-fold CV accuracies using entire data, and A2 represents the accuracies on the independent test set. All results are based on 10 independent runs. Amean and Asd are the mean and standard deviation of the accuracies, and Abest is the best results.
Table 3. Comparisons of the best accuracies on the NCI60 datasets

<table>
<thead>
<tr>
<th>Research</th>
<th>$A_{cv}$</th>
<th>$A_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jirapech-Umpai and Aitken</td>
<td>76.23</td>
<td>N/A</td>
</tr>
<tr>
<td>Liu et al.</td>
<td>88.52</td>
<td>N/A</td>
</tr>
<tr>
<td>Ooi and Tan</td>
<td>85.37</td>
<td>95</td>
</tr>
<tr>
<td>Lin et al.</td>
<td>87.80</td>
<td>85</td>
</tr>
<tr>
<td>This study</td>
<td>91.80</td>
<td>95.24</td>
</tr>
</tbody>
</table>

Here, $A_{cv}$ is the LOOCV accuracy, and $A_i$ is the accuracy on independent test set. N/A means not available.

dataset is similar to ours with keeping 61 samples and 1000 genes, they used 41 samples for training and 20 for testing; on the contrary, the sizes of our training and test sets are 40 and 21, respectively. So our results cannot be compared with theirs directly. However, the best LOOCV and independent test accuracies of our method are higher than theirs with smaller training set. At the same time, it should be pointed out that the results obtained by Ooi et al. (2003) may be too optimistic because their error rate estimator is different from the widely deployed ones, and the test set was involved in the fitness evaluation. So our method achieves the best performance in this case.

It has been proved that when the ensemble size is infinite, and the base classifiers in the ensemble system are independent and more accurate than random guess, the ensemble can reach 100% accuracy (Hansen and Salamon, 1990). So although by setting the SE size to 5, we cannot obtain higher classification accuracy on the NCI60 dataset, it can still be expected that with the increasing of SE size, the classification accuracy on the NCI60 dataset can reach 100% accuracy. And there is another method to further improve the classification accuracy: fuse the outputs of some individuals. This method will be our future research direction.

When setting the scale of SE to 1, the best result obtained is 89.95% on the NCI60 dataset. So using a tree to deal with each two-class problem cannot guarantee a good performance for such a hard problem. And only by increasing the size of SE to 3, the performance is improved greatly, and the best individual obtained is much better than other studies. For this individual, as listed in Table S2 in the Supplementary Material 1, there are 199 genes (131 of them are unique) in the 27 trees. Although it requires more genes than the results obtained by other studies, it is necessary because the trees obtained by GP are based on the mathematical or logical relationships among genes. Due to the difficulty of the NCI60 dataset, simple trees are not capable of distinguishing all classes, and the most complex tree requires a depth of 9 in this individual.

On the contrary, from Tables S3–S6 in the Supplementary Material 1, it can be found that the best classifiers dealing with other datasets only require simple trees, whose depth level are usually <5. For the SRBCT and prostate datasets, even when setting the SE scale to 1, the highest accuracy on the test set can reach 100%. For the lung and leukemia datasets, such a high accuracy cannot be achieved in this case. But by increasing the size of SE, better results can be easily achieved. For example, when setting the size of SE to 1, the best result is 97.06% on the leukemia dataset. And it is found that for the first and third two-class problem, none of the obtained trees can distinguish all samples. But when setting the SE size to 3, we can obtain 100% accuracy without requiring more accurate trees because the trees in the same SE do not wrongly distinguish a sample simultaneously. It proves that the proposed individual structure of the GP is effective. The details about these individuals are listed in Tables S2–S6.

We use $Z$-score (Li et al., 2001) to evaluate the frequency of different genes. Let $Z = \frac{X_i - E(X_i)}{\sigma}$, where $X_i$ is the frequency that gene $i$ is selected, $E(X_i)$ is the expected number of times that gene $i$ is selected, and $\sigma$ is the SD of variance. Let $p_i$ denote the probability of gene $i$ being selected randomly, which is approximately equal to the total counts of frequency in a population divided by the number of individuals in the population, then divided by the total number of genes. Let $A$ denote the number of individuals, then $E(X_i) = A \times p_i$, and $\sigma = \sqrt{A \times p_i \times (1-p_i)}$.

In this experiment, the size of SE is set to 3. All individuals in the final generations of 10 runs are kept for evaluation. Figure 3 shows the $Z$-score of genes selected by the individuals in the final generation on the NCI60 dataset. All genes have been selected more than once in 10 runs. The reason lies in that there are nine SEs in each individual for the nine categories, which require 27,000 trees in the experiments, so all genes stand a chance to be selected. But from Figure 3, it is apparent that there are always some dominant genes in the gene selection process. In contrast, for all other datasets, some genes are never selected by the final classifiers. For the SRBCT, leukemia, lung and prostate datasets, 1378 out of 2308 genes (59.7%), 5270 out of 7129 genes (73.9%), 1449 out of 3312 (43.8%) and 740 out of 1000 (74.0%) are selected in all runs, respectively. And in different datasets, the range of $Z$-score values change greatly. The $Z$-scores of these four datasets can be found in Figure S1 in the Supplementary Material 1.

In addition, for all datasets, the selected genes are fully investigated, as are listed in the Supplementary Materials. Thirty genes with the highest $Z$-scores are checked on NCBI, and it is found that lots of these genes are related to specific tumors. Some of them are even made biological markers of cancer cell clinically. And from Supplementary Figure S2–S6, it can be found that unsupervised hierarchical clustering of the selected genes’ expression patterns readily separate and largely agree with clinically observed cancer types. The validation on the Gene Ontology (GO) database also indicates that the selected genes are of great biological significance. The details about the results and corresponding discussions are given in the Supplementary Materials 1 and 2.

![Fig. 3. The genes' $Z$-score value on NCI60 datasets. X-axis represents the gene index, and Y-axis represents the corresponding $Z$-scores.](https://academic.oup.com/bioinformatics/article-abstract/25/3/331/244785)
4 CONCLUSIONS

In this article, we proposed a new GP-based approach to deal with the gene selection and classification tasks for multiclass microarray datasets. Here, the multiclass problem is divided into multiple two-class problems, and a set of sub-ensemble systems are deployed to deal with respective two-class problems. Then by fusing these ensembles, an individual is built to deal with a multiclass problem directly. The experimental results prove that our approach can reach high accuracies even when analyzing the datasets with very small sample sizes. At the same time, as trees are constructed with different genes, important genes can be selected as important references for clinic diagnosis or cancer development. For each dataset, the biological significance of the selected genes are validated on the NCBI and GO database.

In conclusion, we hope that our GP-based method could present useful alternatives in the analysis of complex multi-class microarray datasets.

Funding: National Science Foundation of China (30570368, 30700161, 60772130 & 60805021); National Basic Research Program of China (973 Program, 2007CB311002); National High Technology Research and Development Program of China (863 Program, 2007AA01Z167 & 2006AA02Z309); Guide Project of Innovative Base of Chinese Academy of Sciences (KSCX1-YW-R-30); the China Postdoctoral Science Foundation (No.20060390180 and 200801231), and the grants of Natural Science Foundation of Fujian Province of China (No.A0740001 and A0810010).

Conflict of Interest: none declared.

REFERENCES