Multiple Feature Construction for Effective Biomarker Identification and Classification using Genetic Programming

Soha Ahmed[1], Mengjie Zhang[1], Lifeng Peng[2] and Bing Xue[1]
[1]School of Engineering and Computer Science
[2] School of Biological Sciences
Victoria University of Wellington , PO Box 600, Wellington 6140, New Zealand
{soha.ahmed, mengjie.zhang, bing.xue}@ecs.vuw.ac.nz and lifeng.peng@vuw.ac.nz

ABSTRACT

Biomarker identification, i.e., detecting the features that indicate differences between two or more classes, is an important task in omics sciences. Mass spectrometry (MS) provide a high throughput analysis of proteomic and metabolomic data. The number of features of the MS data sets far exceeds the number of samples, making biomarker identification extremely difficult. Feature construction can provide a means for solving this problem by transforming the original features to a smaller number of high-level features. This paper investigates the construction of multiple features using genetic programming (GP) for biomarker identification and classification of mass spectrometry data. In this paper, multiple features are constructed using GP by adopting an embedded approach in which Fisher criterion and p-values are used to measure the discriminating information between the classes. This produces nonlinear high-level features from the low-level features for both binary and multi-class mass spectrometry data sets. Meanwhile, seven different classifiers are used to test the effectiveness of the constructed features. The proposed GP method is tested on eight different mass spectrometry data sets. The results show that the high-level features constructed by the GP method are effective in improving the classification performance in most cases over the original set of features and the low-level selected features. In addition, the new method shows superior performance in terms of biomarker detection rate.

1. INTRODUCTION

The discovery of fingerprints in transcproteomics, proteomics and metabolomics samples is attracting much research in the life sciences [5]. Identifying the variables (genes, proteins, metabolites) that distinguish different populations with certain groups is highly interesting. Such variables are commonly referred to as biomarkers [7].

GECCO'14, Jul 12–16, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2576768.2598292.

Mass spectrometry (MS) offers high throughput analysis of the biological samples by determining the elemental compositions of these samples [1, 8]. The molecules in the mass spectrometer are usually ionized to facilitate the process of measuring the molecules, therefore, each of the molecules is associated with a mass to charge ratio (m/z). Mass spectrometer produces spectra which are composed of m/z values of the charged molecules in the sample and their corresponding abundances (intensities). The mass spectrometer is often accompanied by a liquid chromatography (LC) device. LC separates the molecules and elute them into the mass spectrometer at different retention times. This helps in reducing the complexity of the sample, and hence, increasing the identification numbers. The produced LC-MS spectra is composed of m/z values, retention times and intensities of the compounds in the sample. The number of features of the spectra is usually very large, exceeding thousands, and at the same time the number of samples is very small [27]. This makes biomarker identification extremely difficult [18, 28].

Feature construction can provide a means for dimensionality reduction and producing new high-level features from the original low-level features. The new constructed features will also benefit in discovering the relationship between the original features, and therefore, improve the classification performance [22]. Existing feature construction methods can be classified into three main categories [4, 17], which are wrapper approaches, filter approaches, and embedded approaches. Wrapper approaches use a classification algorithm to evaluate the goodness of the constructed features while filter approaches are independent of any classification algorithm. Embedded approaches construct features during the learning process of a classification algorithm.

Genetic programming (GP) can dynamically build models for classification. Due to the nature of GP which automatically produces high-level features (construction) by combining the original features through the functions used in the evolved models, GP can be a good choice for feature construction. This capability of automatic feature construction can also help in discovering the hidden relationship between features [26], and therefore, increase the classification performance. GP is not limited to a specific transformation model like principle component analysis for example [12]. Moreover, GP can build different transformations without any predefined templates.

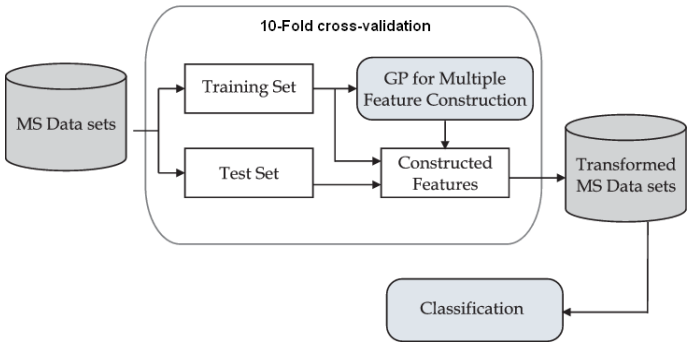

Figure 1: Overview of the GP-multiple feature construction system.

GP has been widely used for feature construction [12, 17, 21, 22] with promising results in terms of improving classification accuracy. There have been different scenarios for the use of GP for feature construction, for example, in [9] GP was used as an embedded approach to construct features, while in [11, 12], the authors used a wrapper approach to construct features. In addition, a filter approach was used in [22] to construct a single feature per class depending on the entropy measure.

Most of the GP based feature construction approaches were based on constructing a single feature and either using this single feature for classification or using this feature along with the original set of features. Using the single constructed feature alone might not achieve acceptable classification accuracy and using the combination of a single constructed feature along with the original set of features will increase the dimensionality [10, 22]. Therefore, the second approach is completely inappropriate for high dimensional data like MS data, where the number of features exceeds thousands. However, none of these methods investigates the effect of constructing multiple features from a single tree during the evolutionary process of GP.

In this paper, we present a new GP approach to constructing multiple features, which uses GP to select a good subset of features and automatically construct new features. The new approach is expected to further decrease the dimensionality of the selected features and improve the classification performance. This method is also evaluated through its performance for biomarker detection on MS data.

Goals

The goal of the paper is to investigate the performance of the features constructed by GP in terms of the classification accuracy and biomarker identification. The new GP method works by taking an embedded approach, where the features are constructed by automatically generating high-level features from the combination of the original low-level features and the functions from the function set. The sub-trees and root nodes are used as the constructed features. Fisher criterion and p-values are used to measure the discriminating information between different classes. Specifically, we will investigate the following questions:

1. How can multiple features be automatically constructed from the evolved GP tree?

2. How can Fisher criterion and the p-values be used to construct a new fitness measure?

3. What is the effect of mixing several compounds of peptides or metabolites in terms of classification accuracy?

4. Whether the constructed features perform better than the low-level selected features?

5. How well can the new method detect the actual biomarkers?

Organisation

The rest of the paper is organised as follows. Section 2 describes the new GP approach. The experiment set-up, the data sets description and preprocessing are presented in section 3. Section 4 reports the experimental results along with the discussions. The conclusions of the paper are presented in section 5.

2. THE NEW GP ALGORITHM FOR MULTIPLE FEATURE CONSTRUCTION

2.1 Overall Structure

GP can automatically produce multiple outputs from its sub-trees and root nodes [32]. The use of subtree outputs (internal nodes) has shown to be effective for classification problems [32] which encourages us to use the internal nodes outputs for constructing new features. Unlike other approaches which use only the output of root node of the evolved tree as the constructed feature, the sub-tree nodes' output are also used as high-level features here. This will help in construction of more features from a single evolved tree and not from multiple trees (runs) and therefore, reduce the computational cost. The more high-level features can also improve the classification accuracy. The proposed GP method uses the original low-level features to construct multiple features. The constructed features are the outputs of the functions that are calculated using the original features. For example, if two original features from the terminal set are mixed with a multiply function, the constructed feature is the output of the multiplication of those two features. The constructed multiple features are used to transform the original data. Finally, the projected data is used for classification. The overview of the GP multiple feature construction system is shown in Figure 1.

The process is as follows: divide the data sets into training and test sets using ten-fold cross-validation. Use the training set with GP to construct new features, where the goodness of the features are measured using their discriminating power between the classes, which is calculated using Fisher

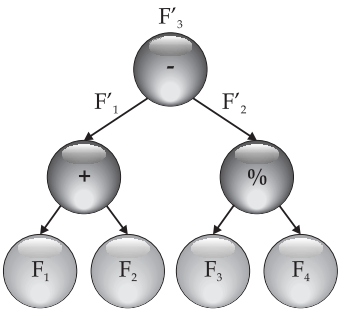

Figure 2: Example of how the features are constructed.

criterion and the p-values. The features are constructed by taking the output of the function on the original features in the evolved program. The new constructed features are used to project both the training and test sets, where different classification algorithms can be used to evaluate new features. Figure 2 shows an example of how the features are constructed from an evolved GP tree. As shown in Figure 2, the two features F_1 and F_2 construct a new feature F'_1, while the two features F_3 and F_4 construct the new feature F'_2. Finally, the new feature F'_3 which represents the final output of the tree is constructed from the new features F'_1 and F'_2. Therefore, this evolved tree will construct three new features from the four original selected features.

2.2 New Fitness Function

The fitness function determines how well a GP tree performs, which is one of the key components in a GP system. Usually, using a wrapper based fitness measure in GP for feature construction can achieve better classification performance than a filter based fitness measure [12], but the computational cost is higher as it requires training a classifier for each individual of the population. Meanwhile, the classification performance success depends more on the discrimination power of the classifier. Therefore, designing a fitness function as an embedded method can avoid those disadvantages.

The Fisher criterion [9] works by maximizing the *between-class* scatter and minimizing the *within-class* scatter. For a two-class problem, the Fisher criterion is defined as

$$\text{Fisher criterion} = \sum_{n=1}^{N} |\frac{\mu_i - \mu_j}{\sigma_i^2 - \sigma_j^2}| \quad (1)$$

where μ_i and μ_j are the means of the samples which belong to class i and class j, respectively. σ_i^2 and σ_j^2 are the variances of the samples which belong to class i and class j, respectively. N is the number of samples in the training set.

For c classes where $c > 2$, the Fisher criterion is firstly calculated for each adjacent pair of classes based on Equation (1) and the summation of those pairs is the final value of Fisher criterion.

In addition to the Fisher criterion, minimizing the p-value between the classes helps in the significant maximization of the distance between the classes. The p-values are calculated using the one way analysis of variance (one-way ANOVA) test which also measures the *between-class* and *within-class* separability. The new fitness function F_p is given by:

$$F_p = \frac{\text{Fisher criterion}}{P_{value}} \quad (2)$$

In Equation (2), the Fisher criterion is the measured distribution of between-class scatter over the within-class scatter of the GP program outputs. The P_{value} ensures that the degree of separation of the GP program outputs of different classes is significantly large. The objective is to maximize the fitness. Therefore, during the evolution, the p-value is minimized and the Fisher criterion is maximized (i.e. the between-class distance is maximized and the within-class distance is minimized).

3. EXPERIMENT SETUP

This section explains the design of the experiments including the data sets that were used in the experiments and the preprocessing of the data sets. The terminal set, the function set and the GP parameters are also explained in this section.

3.1 Data Sets and Preprocessing

In order to test the effectiveness of the new GP approach, eight MS data sets were used. In this section, the data sets characteristics and the preprocessing will be explained. Table 1 summarizes the characteristics of the data sets.

Preprocessing of the MS data involves several steps which is necessary for successful analysis of the data. The MS data sets include binary and multi-class classification problems which are described as follows.

- Pancreatic cancer data set [14]: This data set is acquired using a time-of-flight (TOF) mass spectrometer and the samples were analyzed using surface-enhanced laser desorption/ionization (SELDI). It is composed of 101 healthy samples and 80 cancerous samples. The preprocessing steps includes baseline subtraction where piecewise linear interpolation is used for regression of the baseline. Afterwards, filtering and normalization are performed using Gaussian filter and area under the curve respectively.

- Ovarian cancer low and high resolution data sets [23]: Both of these data sets were analyzed using SELDI-TOF technology. Although the high resolution mass spectra can generate more distinguishable sets of diagnostic features, the high resolution data is more complex than the low resolution data. Similar to the preprocessing of the Pancreatic Cancer data set, the preprocessing of these two data sets involves baseline adjustment, filtering and normalization. The final step performed is the alignment in order to remove the fluctuation in the m/z values. The ovarian cancer high resolution data set contains 121 cancer and 95 healthy samples, while the low resolution data set contains 162 cancerous samples and 91 healthy samples.

- Prostate cancer data set [24]: Samples of three different stages of Prostate Cancer and healthy samples were analyzed using low resolution SELDI-TOF mass spectrometer. It is composed of four classes which are: Healthy (63 samples), Benign (stage_1) (190 samples), Prostate Cancer stage_2 (26 samples) and Prostate Cancer stage_3 (43 samples).

- Toxpath data set [25]: Serum samples with toxicity-related biomarkers were analyzed using SELDI-TOF mass spectrometer. The data set consists of four classes

Table 1: Data sets characteristics

Data Set	# Features	# Samples	#Classes
Pancreatic cancer	6771	181	2
Ovarian cancer low resolution	15,154	253	2
Ovarian Cancer high resolution	15,000	216	2
Prostate cancer	15,000	322	4
Toxpath	7105	115	4
Arcene	10,000	200	2
Apple-plus	773	40	4
Apple-minus	365	40	4

which are: definite positive (34 samples), definite negative (28 samples), probable positive (10 samples) and probable negative (43 samples). The prostate and Toxpath data sets were already baseline adjusted. Therefore, both of the data sets were only filtered and normalized.

The above five data sets were downloaded from FDA-NCI Clinical Proteomics Program[1]. Those data sets are already binned therefore the number of features remains the same after preprocessing. Matlab [20] bioinformatics toolbox was used to perform the preprocessing of the data.

- Arcene data set [3]: Three different MS data sets were combined to produce the Arcene data set which contains 100 samples of cancer patients and 100 healthy samples. The data set is available after preprocessing and it is downloaded from the UCI machine learning repository [3].

- Apple extract data sets [8]: These two data sets are metabolomics data sets where twenty apples were analyzed using LC-MS technology. Four classes are created from the twenty apples where each class contains ten samples. Three classes contain a mixture of known compounds (biomarkers) while the fourth class is not spiked-in with those compounds. The negative and positive ion modes form the two different data sets. The total number of biomarkers is five and twelve in the negative and positive ion modes, respectively. The data sets are available in NetCDF format and it is preprocessed using XCMS [29] with the settings described in [8].

3.2 GP Settings

The standard tree-based GP is used in the experiments where each node outputs a single floating point [6, 26]. The initial population is generated using the ramped half-and-half method [16].

The m/z and retention time variables represent the feature identities of the compounds and the corresponding intensity is the feature value [31]. Therefore, the terminal set is composed of the intensity variable which represents the abundance of the compound in the data. For each sample in a data set, a single floating-point value is produced by the program at the root of its evolved tree [16]. The function set is composed of the four mathematical operators $+, -, \times, \%$ in addition to the operators max, min and if then else ($max, min, IFTE$). The % is a protected division which returns zero for dividing by zero. All the function set members take two arguments except for $IFTE$, which takes three argument and it returns the second argument if the first argument is negative or it returns the third argument otherwise. The evolution terminates at a maximum number of generations of 50. This number is selected as there was no further improvement in increasing the number of generations. The size of population is set to 2000. The tree-depth has been set between 2 and 10. The crossover and mutation rates are set to 0.8 and 0.2, respectively. The tournament selection method is used here and the size is set to 7. An elitist method is taken to ensure the best individual in the next generation is not worse than the current generation and, therefore, keeps the performance monotonically increasing during the evolution [30]. The ECJ [19] package was used in our experiments for running GP. Table 2 shows the various settings of the new method.

[1] `http://home.ccr.cancer.gov/ncifdaproteomics/`

Table 2: GP settings

Function set	$+, -, \times, \%, max, min, IFTE$
Variable terminals	Intensity features
Initialization method	Ramped Half-and Half
Tree Depth	2-10
#Generations	50
Mutation rate	20%
Crossover rate	80%
Elitism	Yes%
Population Size	2000
Selection type	Tournament
Tournament Size	7

3.3 Benchmark Classification Algorithms

To evaluate the classification performance of the constructed features, various linear and non-linear classifier algorithms are used in the experiments. The WEKA package [13] is used to run the classification algorithms. The classification algorithms used are as follows.

1. Multi-layer perceptron (MLP) classifier: It is the implementation of artificial neural networks (ANN) which is a non-linear classifier where the input space is transformed into layers of networks.

2. Naive Bayes Tree (NB-tree): Uses Naive Bayes classifiers at the leave nodes of a decision tree.

3. Random Forest (RF): constructs a multitude of decision trees for training.

4. K- Nearest Neighbors (K-NN): it is the implementation of the nearest neighbors algorithm where the output class is the class of the nearest training example. K is set to 1.

5. Naive Bayes (NB): is a probabilistic method based on Bayes theorem.

6. J-48: The C4.5 decision tree classifier.

7. Decision table (DT): The possible subset of features are used to construct the decision tables. The test set samples are mapped to cells in the decision table. The samples in the test set are then classified according to the label of the majority of training samples of the cell they are mapped to in the table [15].

3.4 Comparison Methods

The performance of the proposed GP method is compared with several methods. Firstly, the original set of features of each data set are used with the seven classifiers for classification. Secondly, the proposed method firstly selects low-level features and through its operators form another set of high-level features. The objective here is to test whether the high-level features can perform better than the low-level features selected by the same method. Therefore, the features selected by the proposed method are compared against the features constructed by it. This method is annotated as Method_1. Finally, another method is used for comparison which is a GP-based feature selection method (Method_2) [2] for MS data. The reason for selecting Method_2 is its previous good performance on MS data. The settings and parameters of Method_1 and Method_2 are set to be the same as the proposed method on the eight data sets. Method_1 and Method_2 select the features which are used in the terminal nodes of the best individual. Both Method_1 and Method_2 are used with the same seven classifiers.

The classification performance of the new GP method for feature construction is compared to that of using all the original features, the low-level features selected by Method_1 and the low-level features selected by Method_2 [2]. For each set of the GP experiments, the GP process is repeated for 30 independent runs with 30 random seeds. A significance test (Z-test) is with 90% significance level is performed to compare the classification performances of the three methods.

4. RESULTS AND DISCUSSIONS

In Table 3, the new GP method is annotated as *GP-Constructed*. The mean ($\overline{x}$), best and the standard deviation (s) of the 30 runs for using the selected and the constructed features with the seven classifiers are reported in Table 3. "Avg#" shows the average number of selected or constructed features by each method. The evaluation of the seven classifiers is done through ten-fold cross validation. The accuracy of using all the original features is also reported in the same table and shown by "All".

In Table 3 the sign $^{\intercal}$ means that the proposed method is significantly better than using all the features, while the sign † means that the proposed method is significantly better than Method_1. The sign * means the new method is significantly better than Method_2. The experiments were run on a machine with an Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, running Ubuntu 4.6 and Java 1.7.0_25 with a total memory of 8GByte.

4.1 Comparison of the Constructed Features with All the Original Features

As shown in Table 3, for all the data sets except for Apple-plus and Apple-minus, using the original set of features with MLP and NB-tree were both running out of memory and did not manage to produce the results due to the huge search space.

The best classification performance of the GP constructed features is better than using the original set of features on all the data sets except for Apple-plus and Apple-minus data sets, where their performance were both ideal. The average classification performance of the GP constructed features is significantly better than using all the original features on almost all the MS data sets excluding the LC-MS data sets (Apple-plus and Apple-minus). This suggests that GP can benefit in both selecting a good set of features and at the same time in discovering the hidden relationship between the features by constructing the new features that can perform better.

For all the seven classifiers, the features constructed by GP managed to improve the classification accuracy over using all the original features. On the Ovarian Low, Ovarian High, Apple-plus and Apple-minus data sets, the constructed features achieved 100.0% accuracy with most of the classifiers. For other data sets, the improvement of the accuracy of the seven different classifiers is 25.97-41.44% on Pancreatic Cancer, 14.5-27.5% on Arcene, and 2.55-27.79% on Prostate Cancer and Toxpath.

In addition to improving the classification performance, the proposed GP approach also helps in reduction of dimensionality. For example in Prostate Cancer data set, the mean number of the constructed features is 26.03, which means that GP reduced around 99.82% of the original dimensionality. The only exception is the Toxpath data set with J-48 classifier where the original features are slightly better than the average performance of the constructed features, but the best performance of the constructed features achieved the same performance. This is mainly due to the imbalance between the number of samples in each class and the embedded feature selection capability of J-48.

4.2 Comparison of the New Constructed Features with the Low-Level Selected Features

The features constructed by the new approach are also compared with the features selected by GP Method_1 (low-level features of the proposed method) and GP Method_2. The objective is to test whether the new smaller set of high-level features constructed by GP can perform better than the selected original low-level features.

Comparing the proposed method with Method_1: In most cases, the classification performance of the features constructed by the new approach (i.e. notated as GP-Constructed) is significantly better than that of the features selected by Method_1 (low-level features of *GP-Constructed*) for most classifiers. For example, On the Toxpath data set, GP-Constructed is significantly better than Method_1 with all the seven classifiers except for NB-tree, where their results are similar.

In terms of the dimension reduction, GP-Constructed further decreased the number of features over Method_1 on all the eight data sets. The average number is reduced by 7.9-29.26 in different data sets. Meanwhile, GP-Constructed either significantly improved or kept the same performance as the low-level selected features in almost all cases.

Comparing the proposed method with Method_2: In almost all data sets, the classification performance of the new approach is significantly better or similar to that of Method_2 for most classifiers. For example, GP-Constructed is significantly better than Method_2 on the Ovarian Low, Ovarian High, and Apple-plus data sets with almost all the seven classifiers, and on the Toxpath and Apple-minus data set with five of the seven classifiers.

The average number of the constructed features is smaller or much smaller than the average number of the features selected by Method_2. The new method reduces the number of features on average from 1 to 221 features over Method_2 on different data sets. With the smaller set of constructed

Table 3: Results of using the constructed, selected and original set of features with seven classifiers.

Data set	Classifier	All		GP-Constructed			Method_1			Method_2		
		Best	Avg#	Best	$\overline{x} \pm s$	Avg#	Best	$\overline{x} \pm s$	Avg#	Best	$\overline{x} \pm s$	Avg#
Pan Cancer	MLP	-	6770	95.55	**88.48±4.97**†*	36.20	94.44	86.17±4.40	65.46	82.60	74.56±6.12	257.66
	NB-tree	-		96.66	89.57 ±3.77*		96.66	90.92 ±3.38		96.73	**92.94 ±2.87**	
	RF	58.56		100.0	**95.38 ±2.14**$^{\intercal\star}$		98.88	95.13 ±2.00		97.83	94.11 ±1.51	
	K-NN	55.80		98.88	95.56 ±1.80$^{\intercal}$		98.88	95.70 ±2.16		100.0	**95.73 ±1.61**	
	NB	51.38		77.77	**62.50 ±5.76**$^{\intercal\dagger\star}$		64.44	57.26 ±3.44		56.52	54.57 ±0.91	
	J-48	50.82		92.77	87.82 ±2.77$^{\intercal}$		94.44	**88.48 ±2.41**		92.93	88.42 ±2.16	
	DT	61.32		81.15	**72.36 ±5.33**$^{\intercal\dagger\star}$		80.44	71.17 ±4.40		75.61	64.57 ±6.12	
Ovarian Low	MLP	-	15154	100.0	99.97±0.14*	27.20	100.0	**99.98±0.07**	46.10	100.0	99.23±0.66	62.03
	NB-tree	-		100.0	**99.68 ±0.45**†*		100.0	99.55 ±0.43		100.0	99.06 ±0.68	
	RF	93.28		100.0	99.67±0.26$^{\intercal\star}$		100.0	**99.73±0.37**		100.0	99.21 ±0.46	
	K-NN	92.09		100.0	**99.95 ±0.20**$^{\intercal\dagger\star}$		100.0	99.76 ±0.47		100.0	99.01 ±0.68	
	NB	76.28		99.21	**96.91 ±1.42**$^{\intercal\dagger\star}$		97.22	94.48 ±2.22		96.87	91.12 ±2.78	
	J-48	95.65		100.0	**98.76 ±1.06**$^{\intercal\dagger\star}$		100.0	98.20 ±1.06		99.22	97.28 ±0.93	
	DT	92.49		100.0	97.97 ±1.39$^{\intercal\star}$		100.0	**97.98 ±0.49**		100.0	97.23 ±2.43	
Ovarian High	MLP	-	15000	100.0	**99.93±0.28**†*	27.26	100.0	98.69±0.80	48.23	100.0	97.00±1.63	63.00
	NB-tree	-		100.0	**99.55 ±0.90**†*		100.0	98.10 ±1.12		100.0	96.12 ±2.01	
	RF	87.04		100.0	**99.71 ±0.51**$^{\intercal\dagger\star}$		100.0	98.33±0.92		100.0	97.17 ±1.08	
	K-NN	86.57		100.0	**99.85 ±0.43**$^{\intercal\dagger\star}$		100.0	98.97 ±0.83		99.10	96.10 ±1.38	
	NB	83.79		100.0	**94.16 ±4.38**$^{\intercal\star}$		98.13	93.97 ±2.36		93.11	88.22 ±3.34	
	J-48	86.57		100.0	**96.76 ±2.72**$^{\intercal\dagger\star}$		98.59	95.28 ±2.00		97.71	93.93 ±1.77	
	DT	82.87		97.93	**94.74 ±2.16**$^{\intercal\dagger\star}$		97.00	93.69 ±3.20		96.21	92.00 ±3.45	
Arcene	MLP	-	10000	99.00	95.48±2.98†	32.50	100.0	96.15±1.50	58.56	99.00	**95.78±1.70**	102.1
	NB-tree	-		98.00	91.57 ±3.10†*		99.00	94.03 ±2.65		99.00	**94.73 ±3.01**	
	RF	72.50		100.0	97.28 ±0.51$^{\intercal\star}$		100.0	**97.50±1.27**		100.0	96.68 ±1.53	
	K-NN	84.50		100.0	**96.73 ±1.48**$^{\intercal}$		100.0	96.70 ±1.49		99.00	96.33 ±1.58	
	NB	70.0		85.50	**72.75 ±7.53**		88.5	72.00 ±6.56		77.50	69.95 ±3.17	
	J-48	81.00		95.50	**90.43 ±2.76**$^{\intercal\star}$		93.50	90.15 ±2.59		94.50	88.65 ±2.45	
	DT	71.50		92.00	83.51 ±4.64$^{\intercal}$		93.67	84.15 ±3.50		94.00	**85.78 ±2.35**	
Pros. Cancer	MLP	-	15154	100.0	96.47±2.62†	26.03	99.68	**97.39±1.54**	41.76	99.39	96.69±1.59	40.83
	NB-tree	-		98.58	95.09 ±2.04*		98.12	94.59 ±1.76		98.78	**96.29 ±1.43**	
	RF	98.75		100.0	**98.83 ±0.90**†*		98.75	97.82±0.76		100.0	98.80 ±0.80	
	K-NN	97.45		100.0	**98.83 ±0.95**$^{\intercal\dagger\star}$		99.37	97.72 ±0.98		100.0	97.74 ±1.04	
	NB	58.13		84.91	**75.37 ±6.13**$^{\intercal\dagger\star}$		82.18	70.34 ±5.25		80.79	69.85 ±6.91	
	J-48	95.00		94.33	**88.55 ±2.86**		90.62	87.71 ±2.13		92.07	87.75 ±2.46	
	DT	72.21		82.25	73.49 ±4.92†		81.25	72.39 ±5.54		83.39	**73.65 ±4.59**	
Toxpath	MLP	-	7105	99.12	**94.42±3.03**†*	37.1	98.25	93.07±2.95	59.40	96.72	91.45±4.56	177.80
	NB-tree	-		99.12	89.56 ±4.82		97.36	**89.94 ±4.80**		96.72	89.84 ±5.75	
	RF	97.36		100.0	**97.92 ±1.39**†*		100.0	97.05±1.75		97.54	93.67 ±2.10	
	K-NN	97.75		100.0	**98.65 ±1.10**$^{\intercal\dagger\star}$		100.0	97.75 ±1.14		96.72	92.57 ±1.54	
	NB	58.12		82.45	**61.99 ±8.89**†*		60.52	51.23 ±4.93		54.91	49.72 ±2.72	
	J-48	89.47		89.47	**83.59 ±3.48**†*		89.47	81.46 ±4.78		88.53	80.19 ±3.68	
	DT	64.91		76.12	**67.42 ±3.03**$^{\intercal\dagger\star}$		78.25	65.07 ±4.45		71.72	62.45 ±2.56	
Apple plus	MLP	100.0	773	100.0	**100.0±0.0**†*	32.30	100.0	99.25±2.38	46.73	96.72	91.01 ±3.68	33.26
	NB-tree	100.0		100.0	**100.0 ±0.0**†*		100.0	98.83 ±2.38		96.72	87.67 ±5.04	
	RF	100.0		100.0	**99.85 ±0.83**†*		100.0	92.65 ±2.33		97.54	91.01 ±3.68	
	K-NN	100.0		100.0	**100.0±0.0***		100.0	**100.0 ±0.0**		98.36	95.24 ±1.94	
	NB	100.0		100.0	95.83 ±1.75*		100.0	**95.93 ±1.87**		71.31	55.57 ±6.62	
	J-48	100.0		100.0	**93.29 ±2.28**†*		100.0	92.58 ±3.23		88.52	80.71 ±4.58	
	DT	100.0		100.0	**97.25 ±2.38**†*		100.0	96.35 ±3.23		96.72	91.01 ±3.68	
Apple minus	MLP	100.0	365	100.0	**99.71±1.66***	28.43	100.0	99.58±1.87	36.33	100.0	98.26±3.63	41.33
	NB-tree	100.0		100.0	99.03 ±2.27*		100.0	**99.75 ±1.01**		100.0	98.96 ±2.67	
	RF	100.0		100.0	**100.0 ±0.0**†*		100.0	99.63 ±0.63		100.0	99.86 ±0.53	
	K-NN	100.0		100.0	**100.0±0.0**		100.0	**100.0 ±0.0**		100.0	**100.0 ±0.0**	
	NB	100.0		100.0	**100.0 ±0.0**†*		100.0	99.58 ±1.33		100.0	92.08 ±9.94	
	J-48	100.0		100.0	**100.0 ±0.0***		100.0	**100.0 ±0.0**		100.0	90.76 ±9.15	
	DT	100.0		100.0	**99.19 ±1.66***		100.0	99.00 ±1.87		100.0	97.26 ±3.63	

features, the new approach still achieved similar or better classification performance than Method_2 in almost all cases.

4.3 Biomarker Identification

We tested the performance of biomarker identification of the proposed method on the Apple-plus and Apple-minus data sets because only in these two data sets, a set of compounds were spiked-in and predefined as the biomarkers.

Figure 3 shows an example of the approach used to count the number of identified biomarkers. As shown in Figure 3, the intersection between the selected features in the terminal nodes of the tree and the predifined set of biomarkers are used as an evaluation of the biomarker identification task.

Table 4 shows the biomarkers in Apple-plus and Apple-minus data sets (positive and negative modes of the ions).

Table 4: Identified spike-in biomarkers by the proposed GP method and Method$_1$ for the Apple data sets. The biomarkers are identified using their m/z values.

m/z values in Apple-plus data set (12 biomakers)	New Method		Method$_2$	
	Selection Status	% of GP runs	Selection Status	% of GP runs
331.21	✗	0	✓	100.0
471.09	✓	80.00	✓	50.00
107.05, 169.05, 238.05, 275.09, 456.11, 459.13	✓	100.0	✗	0.0
456.62, 475.10	✗	0.0	✗	0.0
449.11	✓	66.67	✓	88.0
229.09	✓	90.00	✗	0.0
m/z values in Apple-minus data set (5 biomakers)	New Method		Method$_2$	
	Selection Status	% of GP runs	Selection Status	% of GP runs
463.0	✓	86.67	✗	0.0
447.09	✓	100.0	✓	86.67
273.03	✓	100.0	✓	93.33
435.13	✓	100.0	✗	0.0
227.07	✓	93.33	✗	0.0

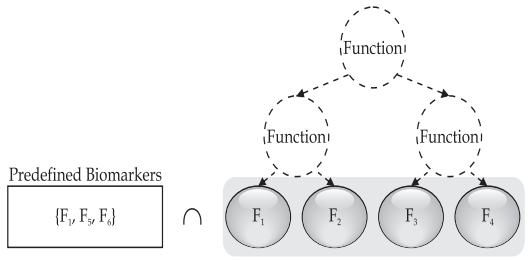

Figure 3: Biomarker Identification approach.

The table also shows the status of identification of the biomarkers by the proposed GP method and Method$_2$. The percentage of runs in which these biomarkers appear are shown in Table 4. As shown in Table 4, GP identified the complete set of biomarkers in Apple-minus data sets. Method$_2$ detected only two bimarkers in 93.33% and 86.67% of the runs, respectively. For Apple-minus data set, the new GP method detected three biomarkers in all its 30 runs and the remaining two in 86.67% and 93.33% of the runs. For the Apple-plus data set, nine out of the twelve biomarkers (75%) are detected by the proposed GP method, where seven biomarkers are identified in 100.0% of runs and the other three are selected in 66.67%, 80% and 90% of the GP runs. However, Method$_2$ identified only three of the twelve biomarkers. This suggests that the new proposed method can be successfully used for the task of biomarker identification as it constructs a new set of features that can achieve better classification accuracy and biomarker detection rate.

5. CONCLUSIONS AND FUTURE WORKS

The goal of this paper was to test the performance of GP in constructing multiple new high-level features and to examine the effect of these new features in terms of dimensionality reduction, classification performance and biomarker identification. The goal was successfully achieved by developing a new GP method, which takes an embedded approach by maximizing the significant discrimination between different classes. The performance of the high-level constructed features are compared to the whole original set of features and the selected set of low-level features from two methods with seven different classifiers. The results show that the new features performed better than the original set of features for all the data sets with most of the classifiers. The results also show that these smaller sets of new features achieved significantly better or similar performance to the selected low-level features on almost all the data sets. Moreover, the constructed features helped in reducing the dimensionality more than the selected features. The biomarker identification results of the proposed method showed that the new GP method can identify 100.0% of the biomarkers in the Apple-minus LC-MS data set and 75% of the predefined biomakers in the Apple-plus data set. Due to its better classification and biomarker identification performance, the new GP can be successfully applied to this task.

As for future directions, it can be tested if using numerical simplification of the evolved trees can reduce the number of constructed features. Furthermore, as there are no publicly available feature construction methods like feature selection, other feature construction methods will be implemented in the future to compare them to the proposed method.

6. REFERENCES

[1] S. Ahmed, M. Zhang, and L. Peng. Genetic Programming for Biomarker Detection in Mass Spectrometry Data. In *Proceeding of the 25 th Australasian Conference on Artificial Intelligence*, pages 266–278, 2012.

[2] S. Ahmed, M. Zhang, and L. Peng. Enhanced feature selection for biomarker discovery in lc-ms data using GP. In *Proceedings of 2013 IEEE Congress on Evolutionary Computation*, pages 584–591, 2013.

[3] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[4] W. Banzhaf, F. D. Francone, R. E. Keller, and P. Nordin. *Genetic Programming: An Introduction: on the Automatic Evolution of Computer Programs and Its Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[5] C. Baumgartner, M. Osl, M. Netzer, and D. Baumgartner. Bioinformatic-driven search for metabolic biomarkers in disease. *Journal of Clinical Bioinformatics*, pages 1–3, 2011.

[6] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[7] G. Chen, T. G. Gharib, C.-C. Huang, D. G. Thomas, K. A. Shedden, J. M. G. Taylor, S. L. R. Kardia, D. E. Misek, T. J. Giordano, M. D. Iannettoni, M. B. Orringer, S. M. Hanash, and D. G. Beer. Proteomic analysis of lung adenocarcinoma: Identification of a highly expressed set of proteins in tumors. *Clinical Cancer Research*, 8(7):2298–2305, 2002.

[8] S. Datta. Feature Selection and Machine Learning with Mass Spectrometry Data. In R. Matthiesen, editor, *Mass Spectrometry Data Analysis in Proteomics*, volume 1007, pages 237–262. Humana Press, 2013.

[9] H. Firpi, E. Goodman, and J. Echauz. On Prediction of Epileptic Seizures by Computing Multiple Genetic Programming Artificial Features. In M. Keijzer, A. Tettamanzi, P. Collet, J. Hemert, and M. Tomassini, editors, *Genetic Programming*, volume 3447 of *Lecture Notes in Computer Science*, pages 321–330. Springer Berlin Heidelberg, 2005.

[10] H. Guo, L. Jack, and A. Nandi. Automated feature extraction using genetic programming for bearing condition monitoring. In *Proceedings of the 2004 14th IEEE Signal Processing Society Workshop*, pages 519–528, 2004.

[11] H. Guo and A. Nandi. Breast cancer diagnosis using genetic programming generated feature. In *2005 IEEE Workshop on Machine Learning for Signal Processing*, pages 215–220, 2005.

[12] H. Guo, Q. Zhang, and A. K. Nandi. Feature extraction and dimensionality reduction by genetic programming based on the Fisher criterion. *Expert Systems*, 25(5):444–459, 2008.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explorer Newsletter*, pages 10–18, 2009.

[14] S. R. Hingorani, E. F. P. III, A. Maitra, V. Rajapakse, C. King, M. A. Jacobetz, S. Ross, T. P. Conrads, T. D. Veenstra, B. A. Hitt, Y. Kawaguchi, D. Johann, L. A. Liotta, H. C. Crawford, M. E. Putt, T. Jacks, C. V. Wright, R. H. Hruban, A. M. Lowy, and D. A. Tuveson. Preinvasive and invasive ductal pancreatic cancer and its early detection in the mouse. *Cancer Cell*, 4(6):437–450, 2003.

[15] R. Kohavi. The power of decision tables. In *Proceedings of the European Conference on Machine Learning*, pages 174–189. Springer Verlag, 1995.

[16] J. R. Koza. Introduction to genetic programming: tutorial. In *Genetic and Evolutionary Computation Conference,GECCO (Companion)*, pages 2299–2338, 2008.

[17] K. Krawiec. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines*, 3(4):329–343, 2002.

[18] Y. Li, Y. Liu, and L. Bai. Genetic algorithm based feature selection for mass spectrometry data. In *BioInformatics and BioEngineering, 2008. BIBE 2008. 8th IEEE International Conference on*, pages 1–6, 2008.

[19] S. Luke. *Essentials of Metaheuristics*. Lulu, second edition, 2013.

[20] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[21] K. Neshatian and M. Zhang. Unsupervised elimination of redundant features using genetic programming. In *Proceeding of the 22nd Australasian Conference on Artificial Intelligence*, pages 432–442, 2009.

[22] K. Neshatian, M. Zhang, and P. Andreae. A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. *IEEE Transactions of Evolutionary Computation*, 16(5):645–661, 2012.

[23] Petricoin, A. M. Ardekani, B. A. Hitt, P. J. Levine, V. A. Fusaro, S. M. Steinberg, G. B. Mills, C. Simone, D. A. Fishman, E. C. Kohn, and L. A. Liotta. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, 359(9306):572–577, 2002.

[24] E. F. Petricoin, D. K. Ornstein, C. P. Paweletz, A. Ardekani, P. S. Hackett, B. A. Hitt, A. Velassco, C. Trucco, L. Wiegand, K. Wood, C. B. Simone, P. J. Levine, W. M. Linehan, M. R. Emmert-Buck, S. M. Steinberg, E. C. Kohn, and L. A. Liotta. Serum Proteomic Patterns for Detection of Prostate Cancer. *Journal of National Cancer Institute*, 94(20):1576–1578, 2002.

[25] E. F. Petricoin, V. Rajapaske, E. H. Herman, A. M. Arekani, S. Ross, D. Johann, A. Knapton, J. Zhang, B. A. Hitt, T. P. Conrads, T. D. Veenstra, L. A. Liotta, and F. D. Sistare. Toxicoproteomics: Serum proteomic pattern diagnostics for early detection of drug induced cardiac toxicities and cardioprotection. *Toxicologic Pathology*, pages 122–130, 2004.

[26] R. Poli, W. B. Langdon, and N. F. McPhee. *A field guide to genetic programming*. Lulu Enterprises, UK Ltd, 2008.

[27] H. Ressom, R. S. Varghese, E. Orvisky, S. Drake, G. Hortin, M. Abdel-Hamid, C. A. Loffredo, and R. Goldman. Ant colony optimization for biomarker identification from maldi-tof mass spectra. In *Proceedings ofthe 28th IEEE Annual International Conference in Engineering in Medicine and Biology Society*, pages 4560–4563, 2006.

[28] H. W. Ressom, R. S. Varghese, L. Goldman, Y. An, C. A. Loffredo, M. Abdel-Hamid, Z. Kyselova, Y. Mechref, M. Novotny, S. K. Drake, and R. Goldman. Analysis of MALDI-TOF mass spectrometry data for discovery of peptide and glycan biomarkers of hepatocellular carcinoma. *Journal of Proteome Research*, 7(2):603–610, 2008.

[29] C. Smith, E. Want, G. O'Maille, R. Abagyan, and G. Siuzdak. XCMS: Processing mass spectrometry data for metabolite profiling using Nonlinear peak alignment, matching, and identification. *Analytical Chemistry*, pages 779–787, 2006.

[30] M. Smith and L. Bull. Feature Construction and Selection Using Genetic Programming and a Genetic Algorithm. In C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, editors, *Genetic Programming*, volume 2610 of *Lecture Notes in Computer Science*, pages 229–237. Springer Berlin Heidelberg, 2003.

[31] R. Wehrens, P. Franceschi, U. Vrhovsek, and F. Mattivi. Stability-based biomarker selection. *Analytica Chimica Acta*, 705:15–23, 2011.

[32] Y. Zhang and M. Zhang. A multiple-output program tree structure in genetic programming. In *Proceedings of The Second Asian-Pacific Workshop on Genetic Programming*, pages 1–12, Cairns, Australia, 2004.