

Genetic Programming for Biomarker Detection in Mass Spectrometry Data

Soha Ahmed¹, Mengjie Zhang¹, and Lifeng Peng²

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}@ecs.vuw.ac.nz, lifeng.peng@vuw.ac.nz

Abstract. Classification of mass spectrometry (MS) data is an essential step for biomarker detection which can help in diagnosis and prognosis of diseases. However, due to the high dimensionality and the small sample size, classification of MS data is very challenging. The process of biomarker detection can be referred to as feature selection and classification in terms of machine learning. Genetic programming (GP) has been widely used for classification and feature selection, but it has not been effectively applied to biomarker detection in the MS data. In this study we develop a GP based approach to feature selection, feature extraction and classification of mass spectrometry data for biomarker detection. In this approach, we firstly use GP to reduce the “redundant” features by selecting a small number of important features and constructing high-level features, then we use GP to classify the data based on selected features and constructed features. This approach is examined and compared with three well known machine learning methods namely decision trees, naive Bayes and support vector machines on two biomarker detection data sets. The results show that the proposed GP method can effectively select a small number of important features from thousands of original features for these problems, the constructed high-level features can further improve the classification performance, and the GP method outperforms the three existing methods, namely naive Bayes, SVMs and J48, on these problems.

1 Introduction

Mass spectrometry (MS) is a method that measures the mass-to-charge ratio (m/z) of charged particles or ions [4]. It is used for the identification of compounds such as peptides and other chemical compounds. The MS main schema works by ionizing chemical compounds to generate charged molecules or molecule fragments and measures their m/z ratios. MS also enables large-scale discovery of candidates for biomarkers which are in terms of machine learning the features selected to discriminate between different categories which indicate biological processes and pharmaceutical or abnormal changes. However, the high dimensionality and small sample size of the data produced by the mass spectrometer make biomarker detection a very challenging task.

Many machine learning techniques have been used for classification of mass spectrometry data. For example, the random forest algorithm [16], principal component analysis and linear discriminant analysis (LDA) [13] are used to classify mass spectrometry data. In [18], k-nearest-neighbour, support vector machines (SVMs) with linear kernel, quadric discriminant analysis are used for classification. Genetic algorithms and t-test in conjunction with SVMs [12] are used to select 10 m/z values as features for classification using SVMs. However, most of these machine learning techniques for classification cannot effectively handle a huge number (i.e. thousands) of features, and feature selection and dimensionality reduction of data must take place in advance.

Genetic programming (GP) is an evolutionary algorithm in which computer programs (or functions) are automatically evolved to solve given problems. GP first starts with randomly chosen programs which are modified subsequently by genetic operators such as sexual recombination (crossover) and mutation based on Darwinian natural selection and gene theory [11]. A main advantage of GP is that it is capable of handling a very large number of features as GP has a built-in capability to select features automatically [15]. Therefore GP can be categorized as an embedded approach to feature selection.

Since very recently, there has been a small number of works only using GP for feature selection and classification on bio-data. For example, GP has been used for classifying $^1\text{H-NMR}$ data [7] and achieved good results. GP has also been recently used for peptide quantification of MS data and measurement of protein quantities within biological samples [19]. However, GP has been rarely used in the area of biomarker detection in MS data.

Goals. The overall goal of the study is to develop a GP based method for biomarker detection and classification in the very high dimensional MS data. This method has three steps: we first use GP with two fitness functions for feature selection, then develop a fitness function for feature construction, and finally the selected and constructed features are used for classification. This new GP method will be examined and compared with naive Bayes, SVMs, J48 decision trees on two data sets, a full scan MS data set and a tandem LC/MS/MS data set. Specifically, we will investigate:

1. what terminals and functions can be used to construct programs;
2. what measures can be used as fitness functions for feature selection, construction and classification;
3. whether the proposed GP method can do a good enough job for biomarker detection in the MS data and;
4. whether the new GP method can outperform naive Bayes, SVMs and J48 for these problems.

Organisation. The rest of the paper is organised as follows. Section 2 describes the data sets and preprocessing. Section 3 describes the new GP approach. Section 4 presents the experimental results with discussions. Section 5 concludes the paper with future work directions.

2 Data Sets and Preprocessing

2.1 Data Sets

The first benchmark data set used here is the premalignant pancreatic cancer MS data, which has 181 samples. 101 of them are control samples (class 1) and the other 80 samples form the PanIN (pancreatic intraepithelial neoplasias) class (class 2). Control samples are *normal* cases while PanIN samples are the *diseased* cases. This data set was obtained from the FDA-NCI Clinical Proteomics Program¹. The MS spectra were generated in SELSI-TOF and QSTAR Plusar i systems, which are composed of m/z values on the x axis and the corresponding intensities on the y axis. The number of attributes/features was originally 350,000 for each sample [8], after binning the data the number of features became 6771 per sample.

The second benchmark data set² is a liquid chromatography tandem MS (LC/MS/MS) generated from the hybrid Q-TOF mass spectrometer [17]. It consists of the serum of 5 healthy individuals (class 1) and the serum of the same 5 healthy individuals mixed with known concentration of spike-in peptides (class 2). The total number of original features in this dataset is 10,411. Note that human experts found 13 known features (biomarkers) for this data set.

2.2 Preprocessing

The MS data usually consists of two columns which are the m/z and the corresponding intensity or relative abundance of this ion in the sample. Due to the high amount of noise which occurs due to the system measurement errors, preprocessing of such data is an essential step for successfully analyzing the data. There are two levels of preprocessing for the MS data [14], either low level preprocessing which involves baseline adjustment, filtering and normalization, or the high level preprocessing. The high level preprocessing includes *peak extraction*, peaks identification or extraction in each of the samples; *filtering of peak*, removing of noisy peaks; *retention time alignment*, matching peaks with similar retention times across multiple scans, and using the groups of the matched peaks for time alignment; *peak list deisotoping*, converting the data to the ground truth data; *baseline adjustment*, removal of low intensity peaks; *gap filling*, when peak identification initially fails to recognize some peaks, filling in missing peaks or filling in data for peaks that are genuinely missing from a sample, by matching the raw data at the suitable retention time.

Not all steps of preprocessing are compensatory, and some of these steps are chosen according to the kind of the application [14]. However, steps like peak extraction and alignment should be made on raw data for successful interpretation of the results, and other steps can be chosen by the analyst. It is very important to perform these first steps diligently since the accuracy and reproducibility of results from analyzing LC/MS/MS data sets depend in part on careful data preprocessing [6].

¹ Available at: <http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp>

² Available at: <http://omics.georgetown.edu/massprep.html>

2.2.1 Preprocessing of Premalignant Pancreatic Cancer Data

This data set is a binned MS data so it requires low level preprocessing. Based on the framework in [8] which consists of baseline adjustment, filtering and normalization are used here. Baseline adjustment is useful for removal of low intensity peaks, where the baseline was estimated by segmenting the whole spectra into windows with a size of 200 m/z ratio intensities. The mean values of these windows were then used as the estimate of baseline value at that intensity. To perform regression, a piecewise cubic interpolation method is used. After this step, filtering of noise was done using Gaussian kernel filter. Normalization was done to remove the systematic differences between replicates, which was performed using area under curve where the maximum value of intensity for each m/z ratio is rescaled to 100. Figure 1 shows the original spectrogram and the preprocessed one.

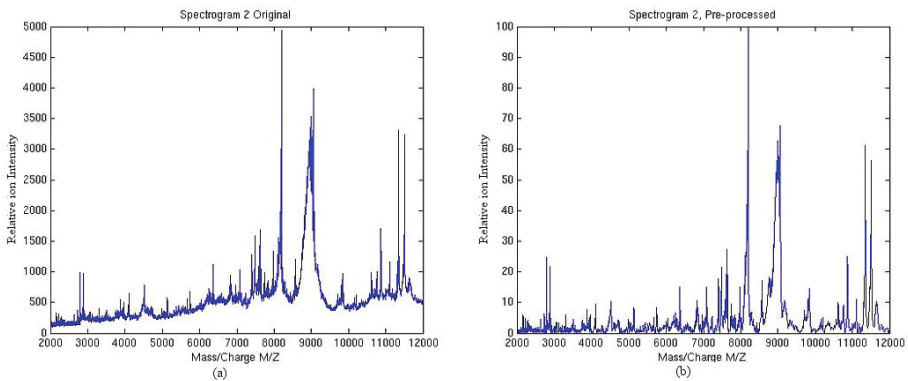


Fig. 1. Premalignant pancreatic cancer data set preprocessing result. Spectrogram 2 is used as an example of the preprocessing. (a) the original spectrogram; (b) the spectrogram after baseline adjustment, noise removal using Gaussian Kernel smoothing and normalization using area under the curve.

2.2.2 Preprocessing of Spike-In Data set

The data set is in its raw form, where the full MS spectra consists of table of values of scan number, LC retention time, ion m/z value and ion intensity. It requires the high level of preprocessing. The first step needed is to extract peaks from the data which is one of the most important steps in MS data analysis. Peptide signals usually appear as local maxima (i.e. peaks) in MS spectra, hence an efficient approach to extracting significant peaks is needed to get meaningful and interpretable results. The peak extraction approach is used based on clustering significant peaks and noisy peaks and removing the noisy peaks. The second step was to filter the peaks to remove noise from each scan, using a percentile of the base peak intensity the filtering is performed, where the base peak is the most intense peak found in each scan. In order to produce the centroid data instead of the raw signal, the peak preserving resampling method is used.

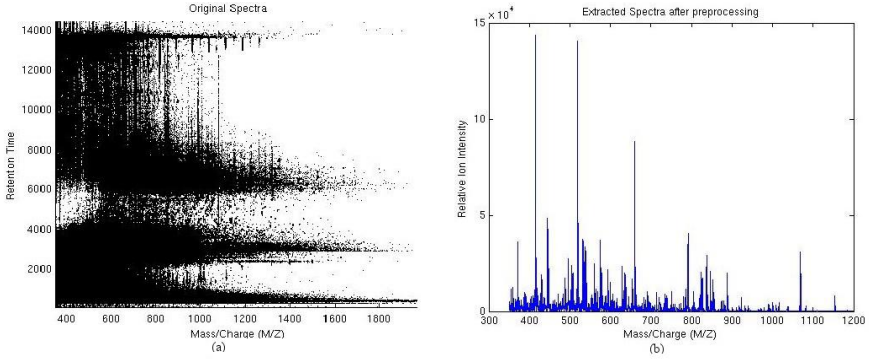


Fig. 2. Spike-in Data set preprocessing. An example of the preprocessing on one of the spectrograms. (a) the original spectrogram; (b) the spectrogram after peak extraction, filtering of peaks, resampling and alignment of peaks.

Finally the alignment of the peaks is used to remove fluctuation of data. The total number of features (scan) was 10,411 and it was reduced to 1252 features per sample after preprocessing. Figure 2 shows an example of the data before and after preprocessing.

3 The GP Method

Although the preprocessing stage has already significantly reduced the number of features, the number of features in MS data sets is still typically very large (too large for most machine learning algorithms to easily cope [8]). So further selection of the important features and construction or generation of new high-level features are still necessary for classification.

In this paper, we use the common tree-based GP [2] for this task. The proposed method has three steps: (1) we use GP's built-in capability for feature selection with the development of two fitness functions; (2) we develop a fitness function for new feature generation/construction; and (3) the selected and constructed features are used in GP for classification. In the rest of this section, we will describe the terminal set, function set, fitness functions, genetic operators and parameter settings for the proposed method.

3.1 Terminal and Function Sets

For n samples, the MS data is typically represented by a $k \times (n + 1)$ matrix: $(m/z, Int) = (m/z, Int_1, \dots, Int_n)$, where k is the number of m/z ratios (features) observed, m/z is a vector of the measured m/z ratios, and the Int_i are the corresponding intensities for the i th sample. There is also a vector $Y = (y_1, \dots, y_n)$ to denote the class label. The main goal is to predict y_i based on the intensity profile $Int_i = (Int_{1i}, Int_{2i}, \dots, Int_{ki})$ [21]. For the two data sets in this study, there are two classes, and the class labels y_i can be defined as class1 or class2, respectively.

We used tree based genetic programs [3], each of which is constructed using a terminal set and a function set. The terminal set consists of the features of a data set, plus a randomly generated constant number. The values of the feature terminals are the intensity values of the corresponding features.

The function set consists of the four mathematical operations, the min, max functions, and a conditional operator *ifte*:

$$FuncSet = \{+, -, \times, \%, \min, \max, ifte\}$$

where $+$, $-$ and $\times$ have the usual meanings while $\%$ is the protected division which is the same division operation except that the result of division by zero returns zero. $\max$ and $\min$ return the greater and the smaller of the two arguments, respectively. The *ifte* takes three arguments, and returns the value of the second argument if the first is negative; otherwise it returns the value of third argument. The use of *ifte* function aims to evolve complex and non-linear functions for feature selection and classification.

3.2 Fitness Functions

To deal with feature selection, construction, and classification, we design three fitness functions as described below.

1. Standard Classification Accuracy: The first fitness function uses the standard classification accuracy on the training set. The classification accuracy is the percentage ratio of the number of examples correctly classified by an evolved genetic program to the total number of examples in the training set.

$$Fit_1 = \frac{\# \text{Correctly Classified Examples}}{\text{Total Number of Examples}} \quad (1)$$

For a particular sample (example), an evolved genetic program will return a single floating point value. If the value is less than or equal to zero (single threshold), then the example is classified as class1; otherwise it is classified as class2. This fitness function will be used for feature selection in the first step and also classification in the third step.

2. Area Under ROC Curve: The first fitness function is generally effective for simple tasks of binary classification. However, it often cannot cope with difficult cases. As the biomarker detection and classification on MS data is a difficult task, we hypothesise that the simple fitness function might not work well. So we use the *area under ROC curve* (AUC) as the second fitness function. Since accurately calculating the AUC of a program is difficult and time consuming, we approximate this value following the approach proposed in [2]. Here multiple threshold values are used, where the program outputs are sorted in ascending order and used as threshold values for classification. Each time the TP (true positive), TN (true negative), FP (false positive), and FN (false negative) values are calculated. The maximum of TP and TN for all thresholds are selected. The fitness function is calculated by the sum of the trapezoids fitted under the ROC

points, where the each point of ROC represents the classification accuracy of the two classes at each threshold value [2]. This function will also be used for feature selection and classification.

$$Fit_2 = \sum_{i=1}^{N-1} 0.5(TPR_{i+1} + TPR_i)(FPR_{i+1} - FPR_i) \quad (2)$$

Where the TPR and the FPR are the true positive and false positive rates respectively.

$$TPR = \frac{TP}{TP + FN} \text{ and } FPR = \frac{FP}{FP + TN} \quad (3)$$

3. Linear Discriminant Analysis: Linear discriminant analysis (LDA) and the related Fisher's linear discriminant are methods to find a linear combination of features which separates two or more classes of objects [22]. This measure is used as the third fitness function of GP, which aims to further reduce the number of features by combining the original low-level features into a single high-level feature, as shown below:

$$Fit_3 = \frac{|\mu_i - \mu_j|}{\sqrt{Var_i + Var_j}} \quad (4)$$

The high-level feature is actually the entire genetic program. The means and standard deviations of the program output values with all training examples for a particular class will be calculated. The "middle point" of the means for the two classes normalised by the standard deviations will be used as the threshold value to determine whether an example should be classified as class1 or class2. If the program output value for a given example is less than or equal to this threshold value, this example will be classified as class1; otherwise class2.

This fitness function is used for construction/generation of brand new, high-level features and provides a solution to dimensionality reduction. We expect this fitness function can further improve classification performance for some machine learning algorithms.

3.3 Genetic Operators and Parameters

For generating the programs in the initial population and for the mutation operator, the ramped half and half method [1] is used. The standard subtree crossover and mutation [3] are applied to this method. Elitism is also used here to make sure the best program in the next generation will not be worse than that in the current generation. For feature selection and feature generation/construction, the population size used is 1024, crossover, mutation and elitism rates are 80%, 19% and 1% respectively. The tournament selection size used is 7. The maximum number of generations is set to 50 or it is terminated if ideal fitness is reached. This number is chosen to avoid overfitting as it can occur if the number of generations is increased. The maximum depth of the tree is set to 8.

For evaluation of the selected and the generated features using GP, some of the parameters selected are changed as the search space is less than the original one. The number of generations is set to 30, the tournament selection size is set to 4, and the maximum depth of the tree is set to 7. All other parameters are kept the same.

4 Results

4.1 Results on Premalignant Pancreatic Cancer Data Set

All the preprocessing framework is done using bioinformatics toolbox in Matlab [5]. ECJ package [20] is used for genetic programming, Weka library [10] is used for classification and evaluation.

4.1.1 Feature Selection

The data set consists of 181 samples with 101 belonging to class Control and 80 samples to class Case. Each sample contains 6771 features. The data set is divided into 50% as a training set and 50% as a test set. Each experiment is repeated 30 independent runs, and the mean and standard deviation of the results, the best result and the number of features selected are shown in Table 1. The number of features are selected from the best evolved program.

Table 1. Test set classification accuracy and the number of selected features by GP the first two fitness functions (STD and AUC). SD: standard deviation.

Fitness Function	Best (%)	Mean±SD (%)	# Features Selected
STD-GP	67.78	53.37± 5.69	367
AUC-GP	71.04	57.52± 6.79	100

As shown in Table 1, GP can at best achieve around 70% of accuracy on the test set using both fitness functions since this problem is difficult. Although the GP results are not as good as expected, the common machine learning algorithms such as decision trees (J48), support vector machines (SVM) and naive Bayes from Weka [10] cannot directly run the experiments since the number of features is too large. Due to the implicitly feature selection capability, the proposed GP method successfully runs the experiments and automatically select a small number from the original 6771 features. While 367 features are selected using the first fitness function (standard accuracy), only 100 features are selected using AUC as the fitness function. Of course, if we increase the search space (population size, number of generations, etc.), we might obtain better results, but the main goal here is to select a small number of features from the huge number (6771) to be used for classification. So we will investigate whether the selected features can be used by GP and other machine learning methods for classification.

4.1.2 Feature Generation/Construction

In the above feature selection stage, GP uses its built-in feature selection ability to significantly reduce the number of features/dimensions over an order of magnitude. However, the number of features is generally over 100, which might still be difficult for common machine learning algorithms. Using the third fitness function, the number of features is further reduced to only a single feature, which is the program output with the largest fitness values among all individuals. The individual with the highest fitness value indicates that the difference between the two classes is large since the magnitude of the Fisher criterion value determines the degree of separation of two classes [9]. The newly generated features for different examples will be tested using GP and compared with naive Bayes, SVMs, J48 algorithms for final classification, which is reported below.

4.1.3 Evaluation and Discussions

Table 2 shows the classification results of GP on the test set using the selected features (by the first two fitness functions) and new constructed features (the third fitness function) with a comparison of Naive Bayes, SVMs and J48. The first line of the table shows the accuracy of different methods using the 367 features selected by GP with the first fitness function (standard accuracy), the second line uses the 100 features selected by GP with the second fitness function (AUC), and the third line uses the constructed feature by GP with the third fitness function (LDA).

Table 2. Test set accuracy of GP and SVM, naive Bayes, J48 tree algorithms using selected and constructed features

Fitness	Accuracy (%)			Best(%)		Mean $\pm$ SD (%)	
	Naive Bayes	SVM	J48	AUC-GP	STD-GP	AUC-GP	STD-GP
STD-GP	51.38	56.91	55.80	78.19	72.22	70.05 $\pm$ 4.89	62.59 $\pm$ 5.05
AUC-GP	54.70	55.80	55.80	69.44	63.33	58.71 $\pm$ 6.03	55.59 $\pm$ 4.20
LDA-GP	62.64	62.64	64.84	68.89	65.93	66.82 $\pm$ 1.51	62.20 $\pm$ 2.43

According to Table 2, the features selected and constructed by GP with different fitness functions can be used by the common machine learning algorithms NB, SVM and J48 and lead to comparable or even better results than the original accuracy GP achieved when it was used for feature selection (Table 1). These results suggest that GP can effectively select a small number of features from a huge number of features in MS data for classification. For the three common machine learning methods, the 367 features selected by GP with the first fitness function and the 100 features selected by GP with the second fitness function achieved very similar results. However, the single feature constructed by GP with the third fitness function by the three learning methods achieved considerably better (7-9%) results. This suggests that the single high-level feature (genetic program) captured the regularities for this problem and can be effectively used by all the three learning methods.

For the single constructed feature as input to the GP classifier (line 3), GP with the first fitness function (STD-GP) *on average* achieved very similar

performance(62.20%) to the NB and SVM methods, but 2% worse than J48 (64.84%). GP with the second fitness function (AUC-GP) *on average* still achieved better results than all the three commonly learning methods. The best GP run with both STD and AUC as the fitness functions achieved better results than the three commonly used machine learning methods. One noticeable characteristic for the GP method with both fitness functions is that using the 367 or 100 selected features achieved better results than using the single constructed features. This suggests that GP has very strong ability for constructing good classifiers using different combinations of low-level or medium level features, but cannot do much if the feature level is very high and the number of such features is too small (in this case only one).

4.2 Results on Spike-In Data set

This data set has 10 examples each with 1252 features. The small sample size and the huge number of features makes this problem extremely challenging. As the number of examples is so small, we use the one-leave-out cross validation method in our experiments.

4.2.1 Biomaker Detection

On this data set, human experts identified 13 features (biomarkers) that can successfully describe the classification problem. In this subsection, we would like to investigate how many of these 13 features can be detected by GP.

As the AUC fitness function consistently achieved better performance than the standard accuracy fitness function in GP for all the machine learning methods for the first data set, we use the AUC fitness function for feature selection (biomarker detection) and run 30 independent times. The results in Table 3 shows that GP successfully detected 5 of the 13 biomarkers. Using only the 5 detected biomarkers and other features selected by AUC-GP, the proposed GP method achieved **100%** accuracy. This suggests that the 13 biomarkers identified by human experts are not the only best features in the data set, and there exist other combinations of features that should also be considered as biomarkers as they can achieve perfect results. Note that using standard classification accuracy as the fitness function, GP can only detect one of the 13 biomarkers, which suggests that AUC is better than the standard accuracy as the fitness function for feature selection and biomarker detection on this data set.

4.2.2 Classification Results

Table 4 shows the classification results of GP and the three common existing learning algorithms (NB, SVM and J48) using the features selected and constructed by GP with the three fitness functions. Again, the features selected and constructed by GP can be used by these learning algorithms, and all the four methods achieved better results than the first data set, suggesting that the classification task in this data set is easier than that in the first set. In particular, GP with AUC as the fitness function has the potential to achieve 100% accuracy

Table 3. Biomarkers detected by AUC-GP. The highlighted features are the biomarkers successfully detected by GP.

Feature No.	m/z
1	501.25
2	450.23
3a	530.78
3b	354.19
4	523.77
5a	648.84
5b	432.89
6	586.98
7a	624.99
7b	630.35
8	943.43
9a	712.43
9b	570

(the best run), which will open the door of GP to be applied to biomarker detection and classification in MS data. Although GP with the single constructed features achieved a higher average accuracy, the result is not significantly better (by a T-test) than using the selected features.

Table 4. Test set accuracy of GP and SVM, naive Bayes, J48 tree algorithms using selected and constructed features on second data set

Fitness	Accuracy (%)			Best (%)		Mean $\pm$ SD (%)	
	Naive Bayes	SVM	J48	AUC-GP	STD-GP	AUC-GP	STD-GP
STD-GP	60.00	70.00	70.00	100.00	100.00	72.66 $\pm$ 13.46	63.88 $\pm$ 17.87
AUC-GP	50.00	60.00	70.00	100.00	100.00	80.66 $\pm$ 3.65	58.33 $\pm$ 14.34
LDA-GP	66.67	66.67	100.00	100.00	100.00	86.67 $\pm$ 20.57	74 $\pm$ 26.34

5 Conclusions

Detection of biomarkers in MS data is a challenge due to the high dimensionality of the data and the small sample size. Selecting the right preprocessing steps of MS data is essential for successful classification and detection of biomarkers. In this study we used GP on two kinds of MS data; the first is a full scan MS data and the second is the tandem LC/MS/MS data. Since GP has a built-in feature selection capability, we used GP for selecting features using two different fitness functions. The first is the standard classification accuracy and the second is the AUC as the fitness function. The results show that GP can be successfully used for feature selection in classification of MS data, and construction of high-level features from low-level existing features can further improve the classification performance. The results also suggest that GP as a classifier outperformed the three common machine learning algorithms on the two data sets.

The results on the second data set show that GP can successfully detect 5 out of 13 biomarkers identified by human experts, and the 5 detected features (with other features selected) can lead to perfect classification accuracy (100%), suggesting that there are far more groups of features that can classify the data. This potential of GP will be further investigated in the future.

Although very preliminary, this paper represents a very first work of GP in biomarker detection, feature selection/construction and classification in MS data. Further investigation in this direction will be carried out in the future.

References

1. Al-Sahaf, H., Neshatian, K., Zhang, M.: Automatic feature extraction and image classification using genetic programming. In: ICARA, pp. 157–162 (2011)
2. Bhowan, U., Johnston, M., Zhang, M.: Developing New Fitness Functions in Genetic Programming for Classification With Unbalanced Data, pp. 406–421 (2012)
3. Bhowan, U., Zhang, M., Johnston, M.: Genetic Programming for Classification with Unbalanced Data. In: Esparcia-Alcázar, A.I., Ekárt, A., Silva, S., Dignum, S., Uyar, A.Ş. (eds.) EuroGP 2010. LNCS, vol. 6021, pp. 1–13. Springer, Heidelberg (2010)
4. Boggess, B.: Mass Spectrometry Desk Reference (Sparkman, O. David). *Journal of Chemical Education* 78(2), 168 (2001)
5. Cai, J., Smith, D., Xia, X., Yuen, K.-y.: MBEToolbox: a Matlab toolbox for sequence data analysis in molecular biology and evolution. *BMC Bioinformatics* 6(1), 64 (2005)
6. Cruz-Marcelo, A., Guerra, R., Vannucci, M., Li, Y., Lau, C.C., Man, T.-K.: Comparison of algorithms for pre-processing of SELDI-TOF mass spectrometry data, pp. 2129–2136 (2008)
7. Davis, R.A., Charlton, A.J., Oehlschlager, S., Wilson, J.C.: Novel feature selection method for genetic programming using metabolomic ¹H NMR data. *Chemometrics and Intelligent Laboratory Systems* 81(1), 50–59 (2006)
8. Ge, G., Wong, G.W.: Classification of premalignant pancreatic cancer mass-spectrometry data using decision tree ensembles. *BMC Bioinformatics* 9(1), 275 (2008)
9. Guo, H., Zhang, Q., Nandi, A.K.: Feature extraction and dimensionality reduction by genetic programming based on the Fisher criterion. *Expert Systems* 25(5), 444–459 (2008)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. *SIGKDD Explorations* 11(1), 10–18 (2009)
11. Langdon, W.B., Poli, R., McPhee, N.F., Koza, J.R.: Genetic programming: An introduction and tutorial, with a survey of techniques and applications. In: *Computational Intelligence: A Compendium*, pp. 927–1028 (2008)
12. Li, L., Tang, H., Wu, Z., Gong, J., Gruidl, M., Zou, J., Tockman, M., Clark, R.A.: Data mining techniques for cancer detection using serum proteomic profiling. *Artificial Intelligence in Medicine* 32(2), 71–83 (2004)
13. Lin, Q., Peng, Q., Yao, F., Pan, X.-F., Xiong, L.-W., Wang, Y., Geng, J.-F., Feng, J.-X., Han, B.-H., Bao, G.-L., Yang, Y., Wang, X., Jin, L., Guo, W., Wang, J.-C.: A classification method based on principal components of seldi spectra to diagnose of lung adenocarcinoma. *PLoS ONE* 7(3), e34457 (2012)

14. Listgarten, J., Emili, A.: Statistical and computational methods for comparative proteomic profiling using liquid chromatography-tandem mass spectrometry. *Mol. Cell. Proteomics* 4, 419–434 (2005)
15. Neshatian, K., Zhang, M., Andreae, P.: Genetic Programming for Feature Ranking in Classification Problems. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 544–554. Springer, Heidelberg (2008)
16. Satten, G.A., Datta, S., Moura, H., Woolfitt, A.R., de Carvalho, M.G., Carlone, G.M., De, B.K., Pavlopoulos, A., Barr, J.R.: Standardization and denoising algorithms for mass spectra to classify whole-organism bacterial specimens. *Bioinformatics* 20(17), 3128–3136 (2004)
17. Tuli, L., Tsai, T.-H., Varghese, R., Xiao, J.F., Cheema, A., Resson, H.: Using a spike-in experiment to evaluate analysis of LC-MS data. *Proteome Science* 13+ (February 2012)
18. Wagner, M., Naik, D., Pothen, A.: Protocols for disease classification from mass spectrometry data. *Proteomics* 3(9), 1692–1698 (2003)
19. Wedge, D.C., Gaskell, S.J., Hubbard, S.J., Kell, D.B., Lau, K.W., Eyers, C.: Peptide detectability following esi mass spectrometry: prediction using genetic programming. In: Lipson, H. (ed.) GECCO, pp. 2219–2225. ACM (2007)
20. White, D.R.: Software review: the ECJ toolkit, pp. 65–67 (2012)
21. Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., Zhao, H.: Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics* 19(13), 1636–1643 (2003)
22. Zhu, L., Han, B., Li, L., Xu, S., Mou, H.: Null Space LDA Based Feature Extraction of Mass Spectrometry Data for Cancer Classification. In: BMEI, pp. 1–4 (2009)