Enhanced Feature Selection for Biomarker Discovery in LC-MS Data using GP

Soha Ahmed  
School of Engineering and Computer Science  
Victoria University of Wellington  
New Zealand, 6011  
Email: soha.ahmed@ecs.vuw.ac.nz

Mengjie Zhang  
School of Engineering and Computer Science  
Victoria University of Wellington  
New Zealand, 6011  
Email: mengjie.zhang@ecs.vuw.ac.nz

Lifeng Peng  
School of Biological Sciences  
Victoria University of Wellington  
New Zealand, 6011  
Email: lifeng.peng@vuw.ac.nz

Abstract—Biomarker detection in LC-MS data depends mainly on the feature selection algorithm as the number of features is extremely high while the number of samples is very small. This makes the classification of these data sets extremely challenging. In this paper we propose the use of genetic programming (GP) for subset feature selection in LC-MS data which works by maximizing the signal to noise of the selected features by GP. The proposed method was applied to eight LC-MS data sets with different sample sizes and different levels of concentration of the spiked biomarkers. We evaluated the accuracy of selection from the list of biomarkers and also using the classification accuracy of the selected features via the support vector machines (SVMs) and Naïve Bayes (NB) classifiers. Features selected by the proposed GP method managed to achieve perfect classification accuracy for most of the data sets. The results show that the proposed method strikes a reasonable compromise between the detection rate of the biomarkers and the classification accuracy for all data sets. The method was also compared to linear Support Vector Machine-Recursive Features Elimination (SVM-RFE) and t-test for feature selection and the results show that the biomarker detection rate of the proposed approach is higher.

I. INTRODUCTION

Biomarker detection has an important role in early diagnosis of disease and prognosis of treatment results [1]. Biomarkers can be proteins, peptides (chunks of proteins) or metabolites whose level is changing in case of diseases and they can be further used for determining the stage of a disease in a patient [2]. The success of the biomarker detection process depends on many factors including reproducible and accurate phenotyping of the biological samples, the suitability of the analytical methods which generate the data, the comprehensiveness of preprocessing techniques which extract the information and refine the raw analytical data. Another important factor is the quality of the computational method which detects the limited number of compounds which have the potential to discriminate between classes of samples [1].

Mass spectrometry (MS) and Liquid chromatography-mass spectrometry (LC-MS) are now used increasingly for high throughput analysis of biological data and biomarker detection research. This analysis usually produces thousands of features (features related to compounds and background signal) in a sample but only a small number of samples is available. Features are characterized by their mass to charge ratio (m/z) and retention time values which in ideal case are directly linked to the identity of the proteins or the metabolites. For each pair of m/z and retention time, a corresponding intensity (abundance) is produced as a feature for that compound. LC-MS data preprocessing is usually considered after the LC-MS analysis in order to reduce the data complexity and extract the information relevant to the compounds and decrease features related to noise. However, features produced after preprocessing can be related to preprocessing artifacts as well. Therefore, the performance of preprocessing work-flow will directly affect the biomarker detection process. Preprocessed LC-MS data sets contain a huge number of features compared to the sample size. Furthermore, the LC-MS analysis is rather time consuming, which makes it impossible to increase the number of samples to a level which balances the huge number of features. This makes the process of biomarker detection an extremely challenging problem for any machine learning algorithm.

Biomarker detection usually starts with feature selection to provide the set of features/spikes/compounds which can provide the best classification of the samples. Feature selection is either a filter based method which assesses the quality of features based on an evaluation criteria or built-in methods (wrappers and embedded approaches) which depend on a learning algorithm to select the best list of features [3].

Different statistical and machine learning methods had been used to select features and classify MS data [4]. For example, SVM-RFE was used to select features in [5], and the main algorithm worked by assigning a weight to each feature through training a linear SVM and removing the lowest weight features. SVMs classifiers were then used to classify the protoemics MS data sets. In [6] and [7], three algorithms (principle component analysis and linear discriminant analysis and random forest algorithm) had been used for reducing the dimensionality, extracting features, and classifying MS data. Support vector machines (SVMs), k-nearest-neighbour and quadratic discriminant analysis [8] had been used and compared for MS data classification. There have been very few trials on the classification of LC-MS datasets [9]–[11]. After preprocessing and aligning the data, Idborg et al. [11] used the PCA, unfolded-PCA, partial least squares to classify the normal and the drug injected mice. Radulovic et al. [9] performed the classification by visualization of the data, which was only limited to two samples in training set for each class and one sample in the test set. The main difference of classification...
of the chromatographic versus non chromatographic MS are the preprocessing methods. After the suitable preprocessing, the classification algorithms need not to distinguish between LC-MS and MALDI/SELDI data.

Most of these methods were focusing on finding the feature sets with the highest classification performance based on ranking individual features independently. This totally ignored the relation between features, which can directly affect the selection process [12]. Moreover, these methods were not directly assessing the detection rate of the biomarkers, meaning that there is no validation that the detected features are the real biomarkers.

A. Goals

The main goal of this paper is to develop a GP method for feature selection in LC-MS data sets that contain an extremely large number of features with a relatively small number of samples. To achieve feature selection we used the implicit capability of GP to select subsets of features. The size of the feature subset is determined probabilistically by assigning higher probabilities to smaller sizes. To achieve the ranking, each feature is ranked by its signal to noise ratio (SNR) where the feature with higher SNR is given a higher rank. Specifically we investigate the following questions:

1) How can the search space be reduced by selecting subsets of features that can provide better classification performance?
2) What is the effect of ranking the selected features by GP on the classification performance?
3) What is the detection rate of the biomarkers in the top ranked features selected by GP?

B. Organisation

The rest of the paper is organised as follows. Section II briefly gives the background of the research. Section III presents the GP method for feature subset selection and ranking. Section IV describes the experimental set up of the proposed GP method. In section V, we evaluate the proposed method by applying it to the data sets and investigate how the provided method improves the classification performance. We also measure the performance of the proposed system in the biomarker detection task. Section VI concludes the paper and points to the future work directions.

II. BACKGROUND

A. Genetic Programming for Biomarker Detection

GP is an evolutionary computation algorithm which solves problems by generating computer programs for solutions [13]. It usually starts with a random initial population of individuals to search for a solution to a given task and over a number of generations it modifies the initial solutions through a set of genetic operators guided by the fitness function. The program in the tree based GP is represented by a mathematical function tree where the tree nodes are the terminals or functions. The GP algorithm consists of the following steps [13]–[15]:

1) Initialize a population of individuals;
2) Compute the fitness of each individual according to the fitness function;
3) While the stopping conditions are not met, do the following:
   • Use the selection method to select some of the individuals;
   • Apply changes to the selected individuals using the genetic operators;
   • Put the new individuals to the next generation;
   • Compute the fitness of the individuals in the new generation;
4) Return the highest fitness individual as the best solution.

GP has been used to solve complex learning problems and optimization where the searching space is extremely huge [15]. Unlike other machine learning algorithms, GP can combine several advantages in the same system. This includes the ability to dynamically build models for classification through a set of mathematical rules and expressions and the ability to select the most relevant features for classification.

A small number of studies used GP for establishing the classification of cancer and tumors by the use of the biomarkers (proteomics, metabolic or genomic signatures). These studies used GP to mathematically combine features for the production of good classifiers with maximum information and minimum number of features [16]. Yu et al. [17], for example, used GP classifiers on prostate cancer genomic data sets. In this study it was demonstrated that although GP used fewer biomarkers than other classifiers, it produced a comparable classification accuracy. This suggests that GP can be used for biomarker detection as fewer biomarkers are required in order to reduce the clinical cost [16]. An early study for the application of GP on MS data analysis was done in [18]. GP was used to detect and quantify the percent of sucrose in orange juice. The method performed a regression operation and compared the expected values of the sucrose and the estimated values by GP. In [19] MALDI MS data was analyzed for protein biomarker detection and classification of 106 breast cancer patients. In this study GP was used to select the best protein clusters that can detect the biomarkers to correctly classify breast cancer. In [20], PCA was used for dimensionality reduction followed GP as a classifier. GP was used in [21] to select the discriminating m/z features from a training set composed of 4 samples and one samples was used as a test. In this study, the evaluation was done by visualizing the data and it was not clear how the feature selection process was performed and how many runs the GP was performed. The analysis of the metabolic data using GP was investigated in [22]. The study aimed to detect changes in the levels of biochemicals between thousands of biochemical features. In [23] GP was used to explore the metabolic compounds differences (biomarkers) in specific plants, measuring the concentration of these compounds and finding the rules that discriminates these plants. LC-MS data was used in this study where the experiments was aiming at investigating the function, relation of salicylic acid in the plant defense to block accumulation.

B. LC-MS Analysis and Preprocessing

MS is used for measuring the molecular masses of proteins and smaller compounds. These masses can contain information
for the identification of the species of proteins and metabolites [24]. Inside the MS instrument, the samples are ionized and they are moved to the mass analyzer to measure their mass to charge ratios (m/z) and in the detector the corresponding intensity of each m/z value is measured. In order to make the spectrum simpler and to increase the detection coverage, the MS machine is coupled with a separation technique such as liquid chromatography (LC), in the front to separate the different molecules before the MS detection. The data produced with LC-MS set up is called LC MS/MS spectrum. This spectrum contains the features of retention time, scan numbers, m/z ratios and the corresponding intensities of the parent and the fragment ions [24], [25].

Usually, the MS data is accompanied with a high amount of noise which occurs due to the measurement errors of the system. Therefore, preprocessing of LC-MS data represents an essential step for successfully analyzing the data [25].

Preprocessing framework of LC-MS data is composed of several steps which includes:

1) **Peak extraction**, peaks identification or extraction from background noise, where this step is used to separate the peaks belonging to real compounds from peaks which are produced due to noise;
2) **Peak filtering**, which removes the noisy peaks;
3) **Retention time alignment**, which eliminates the fluctuation of data. It is done by matching peaks with similar retention times across multiple scans.
4) **Baseline adjustment**, removal of low intensity peaks, which are usually due to the machine artifacts and considered as noise; and
5) **Gap filling**, some of the peaks are missing due to failure of peak identification step initially to recognize some peaks. Filling these missing peaks is done by matching the raw data at the suitable retention time.

### III. THE GP METHOD

We use GP to reduce the search space and select subset of features that can achieve the best classification performance with the minimum number of features. An overview of the system is shown in Figure 1. The proposed GP method helps in discovering the hidden relationship between features and the class labels. N parallel runs were carried out on each data set. GP constructs a decision stump for each of the two classes in the data set. This decision stump is the output of GP program where its performance is determined by the fitness value. The performance of the decision stump is measured according to its ability to classify and separate the class labels and at the same time use the minimum number of features. We set a threshold value of zero to classify the instances. Thus, for a specific instance in the training set, if the program output (decision stump for classifying the instances) is \( \leq 0 \), the instance is classified as class 1; otherwise as class 2.

![Figure 1: An overview of the system.](image)

The feature used in all the GP runs are passed to the feature ranker where features used in the programs are ranked according to their signal to noise ratio (SNR). The feature with higher SNR is assigned a higher rank. Features selected by GP are then saved according to their ranks. The projected data sets with top features are passed to SVMs and NB classifiers to measure the classification performance of the ranked features.

Algorithm 1 shows the steps of the proposed GP algorithm. The algorithm starts by taking the data set \( D \) as the input. This data set consists of \( N \) instances and \( c \) class labels. First GP creates \( P \) initial population and initializes the \( f_{\text{max}} \) to zero. The algorithm keeps track of the best program by updating the variable \( f_{\text{max}} \). The main GP search loop is at the while-loop which will terminate either when the maximum number of generations is reached or when the max possible fitness is reached. After that, the best individual with the maximum fitness is saved. For each instance, the program uses the feature values from \( X \) and produces a single floating point value. This value is used as a decision stump to compute \( TP, TN, FP, FN \). The next step is to perform the selection and the breeding and then move to the next generation \( Curr + 1 \). The feature vector \( X \) of \( \text{BestProgram} \) is passed to the SNR feature ranker to return the \( (X,r) \), the vector pair of features selected and their corresponding ranks.

### Algorithm 1 GP algorithm for subset feature selection and ranking

Input \( D \), a dataset of the form \( D=(N,c) \) where \( N \) is a set of instances of size \( n \) and \( m \) original features and \( c \) is the vector containing the class label of the instances.

Output \( (X,r) \), a vector pair with the features selected and their ranks

\[
\begin{align*}
\text{Output} & \quad (X,r), \text{a vector pair with the features selected and their ranks} \\
\text{Input} & \quad D, \text{a dataset of the form } D=(N,c) \text{ where } N \text{ is a set of instances of size } n \text{ and } m \text{ original features and } c \text{ is the vector containing the class label of the instances.} \\
\text{Algorithm} & \quad \text{GP algorithm for subset feature selection and ranking} \\
\text{while} & \quad \text{while } \text{Curr}<M \text{ and } f<f_{\text{max}} \text{ do} \\
\end{align*}
\]

A. **Fitness Function**

We aim at selecting subsets of features that can maximize the classification and at the same time minimize the number...
of features. Therefore our fitness function is multi-objective inspired by [26]. Often the classification accuracy or the error rate of classification can be used as a fitness measure, however these methods in some cases may not get the optimal fitness for classification due to the low variability between classes and the high variability within-class. We use the correlation between the prediction of classification and the observed reality as a measure of raw fitness [27] by restricting this to select the minimum number of features. The correlation is given by:

\[ C_1 = \frac{TP \times TN - FP \times FN}{\sqrt{(TN + FN)(TN + FP)(TP + FN)(TP + FP)}} \]  

where \( TP, TN, FP, FN \) are the number of true positives, true negatives, false positives and false negatives, respectively. This value is normalized so that its value ranges between 0 and +1.0, thus the normalized value will be

\[ C = \frac{1 + C_1}{2} \]  

We use the following fitness function:

\[ f = C \times (1 + \beta e^{-x/n}) \]  

In (3), \( C \) is the measure of normalized correlation obtained in (2), \( x \) is the cardinality of the subset of features used in the program, \( n \) is the original number of features in the data set and \( \beta \) is a parameter which determines the relative importance between the correlation measure and the number of features. \( \beta \) is changing with generations, so initially the use of fewer feature is preferred but with increasing the generations the more importance is given to the classification correlation. \( \beta \) is give by:

\[ \beta = 2\alpha(1 - \frac{Curr}{M}) \]  

where \( \alpha \) is a constant of value of 0.1, \( Curr \) is the current generation number and \( M \) is the total number of generations.

When \( x \) increases the factor \( e^{-x/n} \) decreases exponentially and so the fitness measure. Therefore, if two programs on the same training points gives the same classification correlation, a higher fitness will be given to the one which uses a smaller number of features.

### B. Ranking of Features

Although GP has the ability to select the subset of features that can perform well during the evolution to satisfy the fitness function, GP can not directly perform the ranking of features [14]. Thus, we trace features used by the generated programs in all the GP runs and rank them according to the SNR of each feature in order to use the top ranked features which can reduce the computational and the clinical costs required for validation of the biomarkers. The SNR is the discriminative power of a feature towards a class. Thus for a two-class problem, the SNR captures the separation between the two classes by identifying the mean difference between classes and the variance within each class.

The SNR for \( feature_i \) is calculated as follows [26]:

\[ SNR_i = \frac{\mu_{c1} - \mu_{c2}}{\sigma_{c1} + \sigma_{c2}} \]  

where \( \mu_{c1} \) and \( \mu_{c2} \) are the means of the feature in class 1 and class 2 respectively, \( \sigma_{c1} \) and \( \sigma_{c2} \) are the standard deviations of the feature in class 1 and class 2 respectively.

The higher rank is given to the feature with higher SNR value.

### IV. EXPERIMENTAL DESIGN

#### A. LC-MS Data Sets

The data sets used in our experiments are obtained from Netherlands Bioinformatics Center\(^1\). These datasets are spiked with peptide biomarkers and therefore the biomarker detection rate can be tested clearly by measuring the number of correct features (biomarkers) selected.

The samples collection, preparation, storage, LC-MS analysis and preprocessing is described in [1] and [28].

1) Spiked porcine CSF data set: This data set is composed of two classes of samples: non spiked (class 1) and high-spiked (class 2). Each class contains 5 samples with high between-class variability and low within-class variability. The number of features in each sample is 9889 with 38 added spiked features. This data set is denoted as \( DS_0 \).

2) Spiked human urine data sets: Fifty urine samples were obtained from 15 healthy females and 35 healthy males and spiked with different peptides. Seven data sets were obtained from these samples which belong to two classes either low (class 1) or high spiking levels (class 2). The number of features in each sample in all the data sets is 29529 with 151 added spiked peptides defined by the data providers. These sample classes were obtained by combining different spiking levels and sample sizes, which resulted in data sets of different between-class and within-class variability of spiked peptides. We use the following notation for each of the seven data sets: \( DS_{12-24-30} \), \( DS_{10}, DS_{3}, DS_{4}, DS_{5}, DS_{6}, DS_{7} \) throughout the paper to denote the the seven human urine data sets. Table I describes the characteristics of all the data sets.

<table>
<thead>
<tr>
<th>TABLE I: Data Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>( DS_0 ) porcine CSF</td>
</tr>
<tr>
<td>( DS_0 ) human urine</td>
</tr>
<tr>
<td>( DS_{12-24-30} ) human urine</td>
</tr>
<tr>
<td>( DS_{10}, DS_{3}, DS_{4}, DS_{5}, DS_{6}, DS_{7} ) human urine</td>
</tr>
</tbody>
</table>

\(^1\) Available at:https://trac.nbic.nl/BiomarkerFeatureSelection
TABLE II: GP run parameter values

<table>
<thead>
<tr>
<th>GP Parameter</th>
<th>Parameters Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Population</td>
<td>Ramped Half-and-Half</td>
</tr>
<tr>
<td>Tree Depth</td>
<td>5-8</td>
</tr>
<tr>
<td>Generations</td>
<td>30</td>
</tr>
<tr>
<td>Mutation Rate</td>
<td>0.19</td>
</tr>
<tr>
<td>Crossover Rate</td>
<td>0.8</td>
</tr>
<tr>
<td>Elitisim</td>
<td>0.01</td>
</tr>
<tr>
<td>Population Size</td>
<td>500</td>
</tr>
<tr>
<td>Selection Method</td>
<td>Tournament Method</td>
</tr>
<tr>
<td>Tournament Size</td>
<td>5</td>
</tr>
</tbody>
</table>

B. Experiment Setup

We use tree-based genetic programming where the output of the program is a single floating point. For the GP system, the programs in the initial population are generated using the ramped half-and-half method [13]. The individual program tree depth is set to 5 and it can be increased to 8 during the evolution. The population size is 500. The tournament selection is used with a tournament size of 5. The standard subtree crossover and mutation [13] are used. An elitist policy is taken to make sure the best individual in each generation is not lost. The evolution will be terminated when either the fitness reaches 1 or the maximum number of generations (30 generations) is reached. All the performance measures are evaluated using the 10-fold cross-validation. The experiments on each data set are repeated for 30 independent runs with different random seeds. Therefore a total of 300 runs were adopted (30 \times 10) for each data set. We used the EClIP package [29] for GP and the Weka [30] library for the classification and evaluation processes. Table II summarizes the GP run parameters used in our method.

C. Terminal and Function Set

The LC-MS data is represented by \((RT, m/z, Int) = (RT, m/z, Int_1, ..., Int_n)\), where RT is a vector of the measured retention time, \(m/z\) is a vector of the measured m/z ratios, and the \(Int_i\) are the corresponding intensities for the \(i^{th}\) sample. The main goal is to predict class label based on the intensity profile [31]. Thus our terminal set consists of the \(Int_i\) values which are the features in addition to a randomly generated constant.

TABLE III: Function Set

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Double, Double</td>
<td>Performs mathematical addition</td>
</tr>
<tr>
<td>−</td>
<td>Double, Double</td>
<td>Performs mathematical subtraction</td>
</tr>
<tr>
<td>(\times)</td>
<td>Double, Double</td>
<td>Performs mathematical multiplication</td>
</tr>
<tr>
<td>(%)</td>
<td>Double, Double</td>
<td>Performs protected mathematical division</td>
</tr>
<tr>
<td>(ifte)</td>
<td>Double, Double, Double</td>
<td>Returns the value of the second argument if the first is negative; otherwise it returns the value of third argument</td>
</tr>
</tbody>
</table>

The function set consists of the basic mathematical operations \(+, -, \times\) and \(\%\), and a conditional operator \(ifte\). The use of \(ifte\) function aims to evolve complex and non-linear functions for feature selection and classification. The \(ifte\) returns the second argument if the first argument is less than zero or returns the third argument otherwise. The \(\%\) is a protected division which performs the same division operation except that the result of division by zero returns zero. Table III describes the function set, the number of parameters taken by each function and the type of each parameter.

D. Evaluation

To evaluate the effectiveness of our proposed GP-based feature selection method, we adopted a number of experiments. First we evaluated the selected features by GP using SVMs and NB to investigate the effectiveness of the selected features. After ranking the features selected by GP, we passed the highly ranked features to SVMs, NB classifiers in order to measure the effectiveness of the top ranked features. Usually in biomarker detection process, the detected biomarkers are passed to laboratory biological validation. Thus, fewer biomarkers are required in order to reduce the clinical cost. Our objective is to investigate if we can achieve a high classification performance by using a few highly ranked features. We expect an increase in the classification performance by using highly ranked features and a decrease in the performance when using low ranked features.

V. RESULTS AND DISCUSSION

A. Results of the GP Selected Features

GP selected a much smaller number of features than the original number of features and it managed to improve the classification performance of all the data sets. The numbers of features selected by GP for each of the data sets are 597, 975, 1209, 451, 940, 987, 1289, 578, respectively.

Table IV shows the classification accuracy of the SVMs and NB using 10-folds cross validation on the eight LC-MS data sets. First we used all the original features (Orig.) then we used features selected by the proposed GP method for comparison.

As shown in table IV, in most cases a classifier (SVMs, NB) using features selected by GP achieved much better classification performance than using all the original features. Except for \(DS_0\) and \(DS_1\) the results are the same as using the original features. We can observe that in \(DS_0\) and \(DS_1\) NB achieve 100% accuracy using GP features while SVMs achieve 100% for \(DS_0\) by using both the original and the GP selected features. NB performed better than SVMs for \(DS_1\), \(DS_2\), \(DS_3\) and \(DS_5\). However, in \(DS_4\) and \(DS_7\) SVMs achieve better performance and both of them has the same performance for \(DS_0\) and \(DS_6\).

B. Results of Feature Ranking

Although the selected features made a big improvement than using the original features, the number of features was still big in terms of clinical validation. Thus we hypothesized that ranking of features might be useful in two ways, firstly increasing the performance of the classification and secondly decreasing the computational and clinical cost. In Figure
2 we show the effect of the ranking of features in each data set. It can be observed that the top 10 ranked features managed in achieving perfect classification accuracy (100%) in $DS_0$, $DS_1$, $DS_2$, $DS_5$ when used with both SVMs and NB. The performance of SVMs with $DS_0$ and NB with $DS_1$ remained unaffected by decreasing the number of features which may be due to the high between-class variability and the low within-class variability, which makes the selection of discriminating features simpler. In $DS_2$, NB achieved 100% regardless of the number of features. SVMs achieves 100% with the top 10 features in $DS_3$ while NB reaches 100% with the top 10 features in $DS_4$. The only exception is on data sets $DS_6$ and $DS_7$, where there is low between-class and high within-class variability which makes that task more challenging. In this case the accuracy did not reach 100% but it improves the accuracy from using all the feature sets selected from GP by 9% to 30% in case of SVMs and NB respectively. We can also observe that the performance decreases rapidly as we use more low ranked features. In $DS_0$, $DS_4$, $DS_6$ and $DS_7$ the performance begin to increase again when using the complete set of selected features by GP. The proposed GP method succeeded in achieving perfect classification accuracy for six of the eight data sets which suggests that the proposed GP method can be applied successfully for this task. In the rest of the section we asses the biomarker detection rate of the proposed method with comparison to SVM-REF and t-test feature selection.
TABLE IV: Experimental Results of the GP Selected Features

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Features</th>
<th>SVMs</th>
<th>NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS₀</td>
<td>Org. 9889</td>
<td>100.00</td>
<td>80.00</td>
</tr>
<tr>
<td></td>
<td>GP 597</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>DS₁</td>
<td>Org. 29529</td>
<td>50.00</td>
<td>90.00</td>
</tr>
<tr>
<td></td>
<td>GP 975</td>
<td>80.00</td>
<td>100.00</td>
</tr>
<tr>
<td>DS₂</td>
<td>Org. 29529</td>
<td>58.33</td>
<td>66.67</td>
</tr>
<tr>
<td></td>
<td>GP 1209</td>
<td>83.33</td>
<td>100.00</td>
</tr>
<tr>
<td>DS₃</td>
<td>Org. 29529</td>
<td>83.33</td>
<td>75.00</td>
</tr>
<tr>
<td></td>
<td>GP 451</td>
<td>91.67</td>
<td>100.00</td>
</tr>
<tr>
<td>DS₄</td>
<td>Org. 29529</td>
<td>83.33</td>
<td>73.33</td>
</tr>
<tr>
<td></td>
<td>GP 940</td>
<td>93.33</td>
<td>90.00</td>
</tr>
<tr>
<td>DS₅</td>
<td>Org. 29529</td>
<td>58.33</td>
<td>50.00</td>
</tr>
<tr>
<td></td>
<td>GP 987</td>
<td>66.67</td>
<td>100.00</td>
</tr>
<tr>
<td>DS₆</td>
<td>Org. 29529</td>
<td>75.00</td>
<td>83.33</td>
</tr>
<tr>
<td></td>
<td>GP 1289</td>
<td>91.67</td>
<td>91.67</td>
</tr>
<tr>
<td>DS₇</td>
<td>Org. 29529</td>
<td>76.67</td>
<td>70.00</td>
</tr>
<tr>
<td></td>
<td>GP 578</td>
<td>90.00</td>
<td>70.00</td>
</tr>
</tbody>
</table>

C. Biomarker Detection

We perform an assessment of the proposed GP method in terms of the number of biomarkers detected, since in the data sets used the biomarkers are identified by the data sets providers. GP is compared with SVM-RFE and the parametric univariate t-test for feature ranking. We start with the results obtained from data sets DS₀ and DS₁. These two data sets are composed of a sample size of 5 per class and they are characterized by having high between-class variability. Moreover, these data sets are characterized by having low within-class variability of spiked peptides. Therefore, in these data sets, identification of biomarkers represented here as spiked-in compounds is considered relatively simple. As shown in Figure 3, GP detected 34 out of the 38 biomarkers in DS₀ and 70 from 151 in DS₁. SVM-REF did not detect any of them and t-test detected only 15 in DS₀. In DS₁, SVM-REF and t-test detected 45 and 50 biomarkers respectively. This suggests that even with the simple task SVM-REF and t-test cannot perform well in this task. The data sets DS₂, DS₃, DS₄, DS₅, DS₆ and DS₇ are more challenging, they are composed of human urine samples of different sample sizes per class (6, 12 and 15) and their within-class variability is high. The larger the within-class variance of spiked peptides makes the expected behavior of biomarkers more difficult to detect, since potential biomarkers may be present in sample class with different concentrations. We can observe that in DS₂, DS₃ and DS₄ the performance of GP in detecting biomarkers remains stable regardless of the small sample size. GP outperforms SVM-REF in these three data sets. In case of t-test the sample size affected rapidly the performance of biomarker detection where the greater the sample size the more biomarkers to detect. In DS₅, DS₆ and DS₇ the task is even more challenging where the between-class variability is low, GP outperforms both SVM-REF and t-test by 4-21%. It is noticed that GP detected some of the defined biomarkers in addition to other features, and this set of features managed to achieve perfect accuracy. This can have two possible reasons one is that there are some discriminating features produced due to preprocessing artifacts and in this case none of the machine learning methods can differentiate them. The other strong possibility is that some of the biomarkers was not be defined by the experts due to the complexity of the samples and in this case GP can be used to guide the biomarker detection process. The performance of GP remains stable regardless of the sample size, within-class or between-class variability, which suggests that GP can be applied successfully for this task.

VI. CONCLUSIONS

In this paper we proposed the use of GP for subset feature selection in LC-MS data sets for biomarker detection. Eight LC-MS data sets with spiked-in compounds with different sample size, between-class and within-class variability were used to assess the performance of the proposed method. Features selected by GP are ranked using their signal to noise ratio and the top ranked features are passed to SVMs and NB classifiers. GP managed to select features which improves the classification performance and in many cases provided perfect classification performance. After ranking of features, the top 10 features achieves 100% in most of the data sets, which means GP has the ability to select minimum features with maximum performance. Moreover, the use of more low-ranked features degrades the performance of classification in all cases. This suggests that the proposed ranking of features is
useful not only for improving the classification also for saving computational and clinical cost required for validation afterwards. The number of biomarkers detected by GP outperforms SVM-REF and t-test for feature selection. GP performance was considerably stable in detecting biomarkers and it stroked a considerable compromise between classification accuracy and the biomarker detection rate regardless of the sample size and between-class or within-class variability of spiked peptides. However, SVM-REF and t-test in case of low between-class variability and the smallest sample size did not detect any of the biomarkers. The classification performance of features selected by GP after ranking in many cases is 100% and at the same time GP detected some of the defined of biomarkers in addition to some other features. This can have two possible reasons one is that these discriminating features are generated due preprocessing artifacts. The other possible reason is that GP detected biomarkers that the experts cannot recognize and therefore GP can automatically detect biomarkers, and this hypothesize will be investigated in the future. Since biomarker discovery is usually intended to support clinical diagnosis, it is advantageous to obtain a discriminating feature set with a minimum number of features and maximum classification accuracy. Based on this criteria, the proposed GP method can be a good choice due to the minimum number of features it can select with better classification performance.

REFERENCES


