Genetic Programming for Biomarker Detection in Classification of Mass Spectrometry Data

by

Soha Ahmed

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Doctor of Philosophy
in Computer Science.

Victoria University of Wellington
2015

Abstract

Mass spectrometry (MS) is currently the most commonly used technology in biochemical research for proteomic analysis. The primary goal of proteomic profiling using mass spectrometry is the classification of samples from different experimental states. To classify the MS samples, the identification of protein or peptides (biomarker detection) that are expressed differently between the classes, is required.

However, due to the high dimensionality of the data and the small number of samples, classification of MS data is extremely challenging. Another important aspect of biomarker detection is the verification of the detected biomarker that acts as an intermediate step before passing these biomarkers to the experimental validation stage.

Biomarker detection aims at altering the input space of the learning algorithm for improving classification of proteomic or metabolomic data. This task is performed through feature manipulation.

Feature manipulation consists of three aspects: feature ranking, feature selection, and feature construction. Genetic programming (GP) is an evolutionary computation algorithm that has the intrinsic capability for the three aspects of feature manipulation. The ability of GP for feature manipulation in proteomic biomarker discovery has not been fully investigated. This thesis, therefore, proposes an embedded methodology for these three aspects of feature manipulation in high dimensional MS data using GP. The thesis also presents a method for biomarker verification, using GP. The thesis investigates the use of GP for both single-objective and multi-objective feature selection and construction.

In feature ranking, the thesis proposes a GP-based method for ranking subsets of features by using GP as an ensemble approach. The proposed

algorithm uses GP capability to combine the advantages of different feature ranking metrics and evolve a new ranking scheme for the subset of the features selected from the top ranked features. The capability of GP as a classifier is also investigated by this method. The results show that GP can select a smaller number of features and provide a better ranking of the selected features, which can improve the classification performance of five classifiers.

In feature construction, this thesis proposes a novel multiple feature construction method, which uses a single GP tree to generate a new set of high-level features from the original set of selected features. The results show that the proposed new algorithm outperforms two feature selection algorithms.

In feature selection, the thesis introduces the first GP multi-objective method for biomarker detection, which simultaneously increase the classification accuracy and reduce the number of detected features. The proposed multi-objective method can obtain better subsets of features than the single-objective algorithm and two traditional multi-objective approaches for feature selection. This thesis also develops the first multi-objective multiple feature construction algorithm for MS data. The proposed method aims at both maximising the classification performance and minimizing the cardinality of the constructed new high-level features. The results show that GP can dis- cover the complex relationships between the features and can significantly improve classification performance and reduce the cardinality.

For biomarker verification, the thesis proposes the first GP biomarker verification method through measuring the peptide detectability. The method solves the imbalance problem in the data and shows improvement over the benchmark algorithms. Also, the algorithm outperforms a well-known peptide detection method. The thesis also introduces a new GP method for alignment of MS data as a preprocessing stage, which will further help in improving the biomarker detection process.

Dedication

To my source of inspiration, the greatest and most important love in my life, Hamza, my beloved son.

Acknowledgment

I would like to acknowledge and express my gratitude to everyone who gave me assistance and support during my PhD study.

First of all, my deep thanks and gratitude to my supervisors, Prof.Mengjie Zhang and Dr. Lifeng Peng. Prof. Mengjie Zhang has dedicated his efforts and time to improve my research skills and writing, providing me with his precious feedback on every detail in my work. Dr. Lifeng Peng provided a lot of support and helped improve my biological background through the interesting discussions we had in the Mass Spectrometry lab. I am also grateful to her for helping me improve my research writing skills. I must also thank and acknowledge my colleague and dear friend Dr. Bing Xue, who helped me a lot by providing feedback on my research papers and thesis.

I als o want to acknowledge the Victoria Doctoral Scholarship, the Marsden Fund of New Zealand (VUW0806), as well as the School of Engineering and Computer Science and Faculty of Engineering at Victoria University of Wellington for their financial support over the past three years. I wish to thank my friends in the Evolutionary Computation Research Group (ECRG), who made a creative and cooperative research environment. Last but not least, I wish to thank my family especially, my husband, my parents and sister for their love, encouragement and support.

A List of Publications

1. Soha Ahmed, Mengjie Zhang, Lifeng Peng: Improving feature ranking for biomarker discovery in proteomics mass spectrometry data using genetic programming. Connect. Sci. 26(3): 215-243, 2014

2. Soha Ahmed, Mengjie Zhang, Lifeng Peng: Measuring Peptide Detectability in LC-MS Data using Genetic Programming. Submitted to Natural Computing. Under review.

3. Soha Ahmed, Mengjie Zhang, Lifeng Peng, Bing Xue: Multiple feature construction for effective biomarker identification and classification using genetic programming. Proceedings of the 23rd Genetic and Evolutionary Computation Conference (GECCO), Vancouver, BC, Canada, 2014: 249-256.

4. Soha Ahmed, Mengjie Zhang, Lifeng Peng: Prediction of detectable peptides in MS data using genetic programming. Proceedings of the 23rd Genetic and Evolutionary Computation Conference (GECCO Companion) , Vancouver, BC, Canada, 2014: 37-38.

5. Soha Ahmed, Mengjie Zhang, Lifeng Peng: A New GP-based Wrapper Feature Construction Approach to Classification and Biomarker Identification. Proceedings of IEEE Congress on Evolutionary Computation 2014, Beijin, China, 2756-2763

6. Soha Ahmed, Mengjie Zhang, Lifeng Peng: A Genetic Programming Based Approach to Multiple Alignment of Liquid Chromatography-

Mass Spectrometry.Proceedings of the 12th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Computational Biology, EvoApplications (EvoBIO), Lecture Notes in Computer Science. Granada, Spain, 2014: 915-927

7. Soha Ahmed, Mengjie Zhang, Lifeng Peng, Bing Xue: Genetic Programming for Measuring Peptide Detectability. SEAL 2014: 593-604

8. Soha Ahmed, Mengjie Zhang, Lifeng Peng: Enhanced feature selection for biomarker discovery in LC-MS data using GP. Proceeding of the IEEE Congress on Evolutionary Computation, Cancun, Maxico, 2013: 584-591.

9. Soha Ahmed, Mengjie Zhang, Lifeng Peng: Feature Selection and Classification of High Dimensional Mass Spectrometry Data: A Genetic Programming Approach. Proceedings of the 11th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Computational Biology (EvoBIO2013), Vienna, Austria, Lecture Notes in Computer Science: 43-55.

10. Soha Ahmed, Mengjie Zhang, Lifeng Peng: Genetic Programming for Biomarker Detection in Mass Spectrometry Data. Proceedings of the 25th Australasian Conference on Artificial Intelligence, Sydney, Australia 2012: 266-278.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter introduces the thesis and includes the problem statement, the motivations, the research goals, the major contributions, and, finally, the organisation of the thesis.

1.1 Problem Statement

Nowadays, mass spectrometry (MS) has become the most dominant technique for high-throughput analysis of proteomes and metabolomes [169]. The main task of MS data analysis is the classification of samples from different classes.

Classification of MS data from control and treated biological samples can lead to the identification of features (biomarkers), which can predict a specific experimental status. The set of detected features provide information about the effect of therapy or lead to describe new molecular targets for therapeutics [47]. To detect the biomarkers, the obtained MS profile must be analysed through several computational methods.

The mass spectrometer produces spectra datasets, where each spectrum consists of tens of thousands of features, which create a large feature space for classification [173]. Another issue is the small number of samples generated by the mass spectrometer due to cost and time constraints.

These factors make biomarker detection in MS data (identification of features for classification) extremely challenging [182].

The spectrum is also interrupted by different kinds of noises during the stages of analysis in the mass spectrometer, which results in reducing the quality of the data. Due to the noise introduced to the MS data spectra, a preprocessing framework must take place. The selection of the preprocessing steps and parameters can also affect the biomarker detection process [142]. In addition to the previously mentioned challenges in MS data, the biomarkers sometimes appear at a low abundance that introduces another difficulty in the selection of the correct proteins that can be defined as the biomarkers.

After the biomarker detection, the biomarkers pass through two other stages which are shown in Figure 1.1. The detection stage produces the possible list of biomarkers, which then pass to verification through measuring their detection probability in the mass spectrometer. Finally, the verified biomarkers pass to the experimental validation that involves testing these biomarkers laboratory. The first two stages of biomarker identification (feature manipulation and biomarker verification) are tackled in this thesis. However, the experimental validation is beyond of the scope of this research.

Underlying computational problems: The compuational problems here are firstly the large number of features. This introduces the problem of curse of dimensionality which represents a major obstacle for classification. Generally, in feature manipulation, the search space increases exponentially according to the number of features(2^n subsets of features if the number of features is n) [110]. In case of thousands of features, it is impossible to exhaustively search all subsets of features which form the candidate solutions. Secondly, the small number of training examples which makes the search for the relevant features more difficult [178]. Thirdly, for verification, the class imbalance problem of the data which makes the

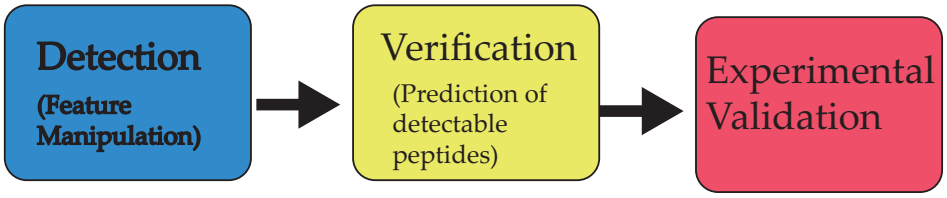


Figure 1.1: Stages of the biomarker identification process

performance more the bias to the majority class, and hence, the features selected or constructed will also be biased to the majority class [70].

Meanwhile, most of biomarker detection methods produce a long list of biomarkers, and it is impractical to send all of them for experimental validation. Intermediate verification of biomarkers can reduce the number of biomarkers for experimental validation, and thus reduce the cost. Biomarker verification of proteomics MS data can be performed through the prediction of their detectability in the mass spectrometer, which can be done through linking the peptide detectability to their physiochemical properties. However, peptide detection is not an easy task either because of the imbalance problem of the data (i.e. the number of detected peptides in the observed class is very small compared to the number of peptides in the non-observed class). The imbalance problem makes building a predictive model for peptide detection a challenging problem.

Feature manipulation can help to solve the biomarker detection problem [8]. It provides a means to transform the representation of the input to a classification algorithm to improve its performance [106]. Feature manipulation consists of *feature selection* and *feature construction*. *Feature manipulation* is particularly useful in solving the problem of *the curse of dimensionality*, which causes the degradation of the performance of the classification algorithm. *Feature ranking*, which is a branch of feature selection, involves weighing individual features according to their relevance to the classification task [18]. *Feature selection* directly eliminates non-relevant and redundant features of the problem [117]. Neither feature ranking nor selection creates new features, while *feature construction* creates new high-

level features as functions of the original features to enhance the quality of representation [63].

Feature manipulation can be performed as filter, wrapper or embedded approaches [117]. In filter approaches, information based evaluation is typically used to measure the goodness of the features [106]. In wrapper approaches, a classification algorithm is wrapped to the system to evaluate the features. In the embedded approaches, the classification algorithm is itself the feature selection method, i.e. the feature manipulation and the classifier training process are combined into a single process [123].

Feature manipulation of the high dimensional MS data is a hard task due to the large search space and the small number of samples. Many machine learning approaches have been proposed to solve feature selection and ranking in MS data [148, 156]. However, most of these approaches depend on the univariant feature selection measure, such as formal statistical tests (t-test) or independent feature ranking. The individual ranking or selection of features obviously neglects the relationships between the features [31, 104, 162, 188].

Also, very few works on biomarker detection consider feature construction to improve the performance of classification. Furthermore, verification of the detected biomarkers, using their peptide detectability, has not been considered to date.

Genetic programming (GP) is an evolutionary algorithm [92] which can build programs and functions dynamically. The programs built by GP range from mathematical expressions to classification models, and, GP can therefore be flexible in searching the complex solution space. GP has the potential to deal effectively with the challenges in MS data. However, there has been little work using GP for MS biomarker detection.

1.2 Motivations

The MS technology effectively performs the analysis and characterization of alterations in proteins and metabolites. It offers the possibility of discovering novel biomarkers through the use of machine learning approaches to analyse the data [120].

Biomarker detection is a difficult problem, especially when the number of features is large, thus increasing the search space exponentially [25]. Most of the benchmark biomarker detection paradigms depend on the individual feature evaluation (single feature ranking). The individual evaluation introduces a set of redundant features which, when they are working together, often degrade the classification performance.

The filter approach for feature selection might introduce redundancy to the features. Despite the better performance of the wrapper approaches, they require high computational cost and can lack generalisability to classifiers other than the wrapped classifier used for evaluation. The embedded approach has the potential to avoid the disadvantages of a filter and wrapper approaches as it is not computationally intensive like the wrapper approach, and also can provide better performance than the filter approach [106].

1.2.1 Why GP

GP is an effective evolutionary global search algorithm [85]. The most common form of GP is a tree-based representation that offers great potential for biomarker detection through feature manipulation. The major reasons are presented as follows.

- GP has an automatic and intrinsic feature manipulation capability as a part of the evolutionary process for building a classification model. Therefore, it acts as an embedded approach [178].

  - For feature selection, the individuals of GP consist of functions, variable terminals and constant terminals. The variable terminals correspond to the features of the dataset. These individuals vary in size and not all the features appear in the individuals, which make the selection of features intrinsically in GP [92].
  - For feature ranking, in the variable-length individuals, some features appear more than once, which gives a weighing factor to the more frequent features.
  - For feature construction, the features can be combined through the mathematical operators by means of individual codification, which constructs a high-level feature automatically [124].

- GP as an evolutionary technique has the potential to search huge spaces for optimal or near optimal solutions, unlike some existing techniques that are often trapped in local optima (e.g. Hill climbing) [92].

1.2.2 Limitations of Existing Approaches

The limitations of the existing approaches for MS biomarker detection are summarised in Table 1.1.

1.3 Goals

The overall goal of this thesis is to develop a new embedded GP approach to biomarker detection and verification in the classification of MS data. To achieve this goal, we investigate the following research objectives.

1. Develop a new GP ensemble approach to feature ranking. The existing approaches depend mostly on an invariant measure to evaluate each feature independently which can result in redundant or irrelevant features. The proposed algorithm is expected to combine ad-

Table 1.1: limitations of the Existing Approaches

using individual feature ranking, which ignores relationships between features [188],	The potential of GP for feature selection or ranking has seldom been investigated for biomarker detection in MS data.
using wrapper approaches which are computationally expensive, especially with the high-dimensionality of MS data [133],	Using GP in classification problems, as either a filter or wrapper approach for feature ranking, selection and construction, has shown promising performance. However, the use of the embedded capability of GP, which can combine both filter and wrapper approaches' advantages, has not been seriously investigated.
using principle component analysis or wavelet transformation to transform the data for classification, which require certain assumptions and constraints and are limited to specific kinds of tasks [144].	The potential of GP in biomarker detection, particularly in feature construction has not been investigated before.
There is gap between biomarker detection and biomarker verification	the use of GP for biomarker verification has not been explored before, especially to solve the problem of imbalance in the datasets.
Most previously approaches are single objective	Biomarker detection is a multi-objective problem. The objectives are maximising classification performance and minimising the number of features used for classification, considering that the detected features must pass an experimental validation stage. Experimental validation is costly and, careful selection is needed to decide which biomarker will pass through it. However, GP based multi-objective optimisation has never been used for MS biomarker detection.

vantages of several feature ranking metrics, select a smaller number of top ranked features, and evolve a better ranking of the features. This objective will be discussed in Chapter 3.

2. Develop a new GP approach to multiple feature construction. Most of the previous GP feature construction approaches are taking a wrapper approach to constructing a single feature which is insufficient to improve the classification or take a filter approach and use multiple GP trees to develop multiple features equal to the number of classes and this will increase the computational cost. Our approach will construct new high-level features by taking an embedded approach, and will use a single GP tree to construct multiple features automatically through the use of the output of different branches of the tree. The proposed approach is expected to discover complex relationships between original features, reduce the number of selected features, and enhance the classification performance. This objective will be discussed in Chapter 4.

3. Develop a new GP embedded approach to feature selection in MS data using multi-objective optimisation. The existing approaches did not consider multiobjective optimisation for MS biomarker detection and hence, it is worth investigation. The proposed approach is expected to select subsets of the original features that enhance the classification performance using the intrinsic capability of GP to select features automatically while building the classification model. Meanwhile, the multi-objective method is expected to keep the trade-off between the classification accuracy and the number of features without the prior assumption of the importance of each objective. The set of non-dominated feature subsets is expected to be better than or similar to the SPEA2 [192] and NSGAII [27] approaches for feature selection and the single objective approach. This objective will be discussed in Chapter 5.

The single objective method feature construction approach proposed in Chapter 4 will be extended to a multi-objective method in that Chapter. The new GP multi-objective algorithm is expected to keep the trade-off between the classification performance and the number of new features. The proposed GP method is evolving a set of Pareto-front non-dominated constructed features which can improve the performance of the classification algorithm.

4. Develop a GP biomarker verification method that will be trained on both yeasts and human serum datasets using the peptides physico-chemical properties as features. This method is intended to bridge the gap between the biomarker detection and the experimental validation stages which has not been considered before. The capability of GP to perform unbalanced classification will be used to build the prediction model of the peptide detection problem. The detected biomarkers can be provided as examples in the test set to the system to verify their detectability in the mass spectrometer. This objective will be discussed in Chapter 6.

5. Further investigation on the MS data preprocessing will be performed by developing a new GP approach for multiple alignment of the LC-MS data. The proposed method will help in correcting the distortion in the data and thereby, improve the biomarker detection process. This method is presented in Chapter 7.

1.4 Major Contributions of the Thesis

The thesis makes the following contributions that are summarised in Figure 1.2.

1. The thesis proposes a new embedded-based ensemble feature ranking approach by combining two feature ranking metrics. The new

approach considers both the classification performance and the number of features. The proposed method successfully improves the performance over using each metric independently by eliminating the redundant features and selecting the most relevant ones. The system also acts as a classifier and proves the capability of GP to outperform state-of-the-art classifiers. Testing the proposed approach on MS datasets with predefined biomarkers shows that the top ranked features contain most of the predefined biomarkers.

Parts of this contribution have been published in:

Soha Ahmed, Mengjie Zhang, Lifeng Peng: Improving feature ranking for biomarker discovery in proteomics mass spectrometry data using genetic programming. Connect. Sci. 26(3): 215-243, 2014.

Soha Ahmed, Mengjie Zhang, Lifeng Peng: Feature Selection and Classification of High Dimensional Mass Spectrometry Data: A Genetic Programming Approach. Proceedings of the 11th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Computational Biology (EvoBIO2013), Vienna, Austria, Lecture Notes in Computer Science: 43-55, 2013.

Soha Ahmed, Mengjie Zhang, Lifeng Peng: Enhanced feature selection for biomarker discovery in LC-MS data using GP. Proceedings of the IEEE Congress on Evolutionary Computation (CEC2013), Cancun, Mexico, 584-591 2013.

Soha Ahmed, Mengjie Zhang, Lifeng Peng: Genetic Programming for Biomarker Detection in Mass Spectrometry Data. Proceedings of the 25th Australasian Conference on Artificial Intelligence (AAI2012), Sydney, Australia 2012: 266-278.

2. The thesis proposes a novel multiple feature construction GP system. The existing feature construction methods can construct either a single feature or multiple features from multiple independent runs/programs depending on the number of classes. However, the

proposed algorithm automatically combines original features using the mathematical operations and constructs new multiple high-level features from a single evolved program. The results on several benchmark datasets show that the proposed algorithm has the potential to create new features, which improve the performance of common classifiers. More investigation of the features selected from the construction stage shows that most of the selected features in the GP individuals are the biomarkers predefined by the domain experts.

Parts of this contribution are published in:

Soha Ahmed, Mengjie Zhang, Lifeng Peng, Bing Xue: Multiple feature construction for effective biomarker identification and classification using genetic programming. Proceedings of the 23rd Genetic and Evolutionary Computation Conference (GECCO), Vancouver, BC, Canada, 2014: 249-256.

Soha Ahmed, Mengjie Zhang, Lifeng Peng: A New GP-based Wrapper Feature Construction Approach to Classification and Biomarker Identification. Proceedings of IEEE Congress on Evolutionary Computation 2014, Beijing, China, 2756-2763, 2014.

3. The thesis proposes a new GP-based multi-objective embedded feature selection method for biomarker detection. The system uses GP to find complex relationships between features and classes, and then selects subsets of features based on their discrimination power between different classes. The results show that GP finds more and better relationships between the features more than the conventional methods. The GP multi-objective algorithm uses the ideas of Non-dominated Sorting Genetic Algorithm (NSGA II) [27] and Strength Pareto Evolutionary Algorithm (SPEA2) [192] to maximise the classification accuracy and minimise the cardinality of features. The results show that the embedded approach outperforms the traditional multi-objective methods and the single objective GP approach. This

thesis also proposes the first multi-objective GP feature construction approach for biomarker detection. The system aims at maximising the classification performance and minimising the number of the constructed features simultaneously. The proposed algorithms produce lower number of high-level features that improve the classification performance on the used benchmark MS problems.

4. This thesis uses GP as a peptide detection method to verify the detected biomarkers. The proposed method solves the class imbalance problem in the data and avoids the bias to the majority class by giving equal weights to both classes when building the classification model. The GP method is trained on three different datasets using the peptides physicochemical properties as feature vectors. The proposed algorithm improves the classification accuracy of the minority class which contains the biomarkers peptides thus helping to verify of these biomarkers.

   Parts of this contribution have been published in:

   Soha Ahmed, Mengjie Zhang, Lifeng Peng: Genetic Programming for Measuring Peptide Detectability. Proceedings of the 10th Simulated Evolution And Learning (SEAL2014), Dunedin, New Zealand, 2014, 593-604.

   Soha Ahmed, Mengjie Zhang, Lifeng Peng: Prediction of detectable peptides in MS data using genetic programming. Proceedings of the 23rd Genetic and Evolutionary Computation Conference (GECCO Companion2014), Vancouver, BC, Canada, 2014: 37-38.

5. The thesis presents preliminary results of a new GP method for multiple alignment of LC-MC. The proposed method aims at correcting the distortion that will assist in improving the detecting of the biomarker candidates.

   Parts of this contribution has been published in:

Soha Ahmed, Mengjie Zhang, Lifeng Peng: A Genetic Programming Based Approach to Multiple Alignment of Liquid Chromatography-Mass Spectrometry. Proceedings of the 12th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Computational Biology, EvoApplications (EvoBIO), Lecture Notes in Computer Science. Granada, Spain, 2014: 915-927.

1.5 Organisation of the Thesis

The rest of the thesis is organised as follows.

Chapter 2 presents a review of the literature on biomarker detection of MS data, feature manipulation using GP, and other Evolution Computational(EC) algorithms. The major contributions of the thesis are presented in Chapters 3-6. Each chapter addresses one of the research objectives of the thesis required to fulfill the overall goal. The contributions chapters are shown in Figure 1.2. Chapter 7 presents some investigations on the use of GP for alignment of LC-MS data. Chapter 8 concludes the thesis.

Chapter 2 explains the background for MS data generation, details the nature of the data and the preprocessing steps. The basic background for machine learning, classification, feature manipulation, evolutionary computation, particularly GP, and multi-objective optimisation are presented in this chapter. It reviews related work on MS data biomarker detection, feature manipulation using EC techniques, particularly using GP. It also discusses the open questions and challenges that form the thesis motivations.

Chapter 3 proposes a novel GP-based method for ensemble feature ranking in MS data. It includes two feature ranking metrics, selects the best features from both metrics and measures the goodness of the selected features, depending on their frequency of occurrence in the evolved programs. The scores are then used for further ranking of features. A variety

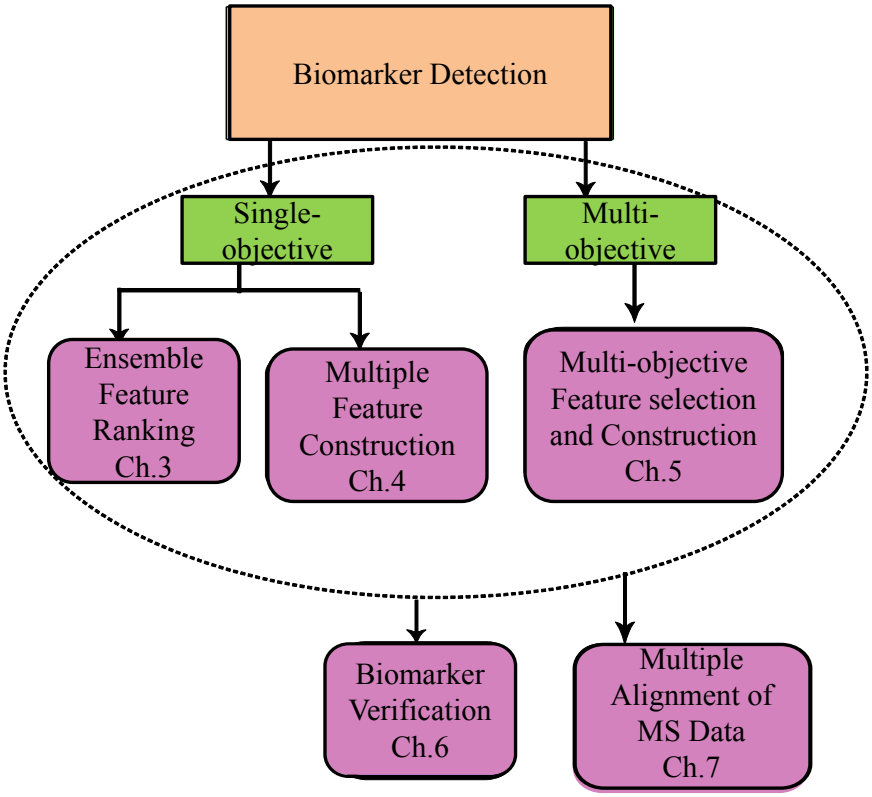


Figure 1.2: Overview of the major contributions

of classification algorithms, including GP, are used to test the top ranked features from several benchmark MS problems.

Chapter 4 develops a new multiple feature construction GP approach. The proposed method automatically constructs multiple new high-level features from a single evolved model. The single objective method uses a new fitness function that maximises linear discriminant analysis and minimises the p-value of the constructed features. The proposed approach with the constructed features is examined and compared with the methods using the original selected features.

Chapter 5 proposes multi-objective GP-subset feature selection methods using embedded fitness evaluation. The multi-objective method aims to maximise the classification accuracy and minimise the number of biomarkers. The method is tested and compared with the single objective method and two benchmark multi-objective methods. The empirical results on several benchmark problems are presented and discussed. The chapter also discusses some advanced topics on feature construction by using multi-objective GP for maximising the classification performance and minimising the number of high-level features. The method is mainly based on GP with ideas from NSGAII and SPEA2 for measuring the Pareto fitness. These algorithms are all embedded approaches that use the capability of GP for feature manipulation during the process of evolving a classification model.

Chapter 6 proposes the use of GP for measuring the peptide detectability through building a classifier that uses the peptides' physiochemical properties to predict whether the peptides will be detected in the mass spectrometer. The detection of peptides indicates whether they can be verified as biomarkers or not. Since the detectable peptides are very small compared to the non-detectable, the problem is tackled as an unbalanced classification problem.

Chapter 7 discusses and examines a GP alignment method that is developed in our work.

Chapter 8 summarises the thesis work and highlights the research goals and overall conclusions. It also indicates some possible future work and research directions.

1.6 Benchmark Datasets for Evaluation

Some benchmark MS datasets with varying difficulty and properties are used to test the performance of the proposed GP approaches. Evaluating the methods also involved testing them on a range of classifiers to test

their generalisability in regards to many classifiers and they are not biased to a specific classier.

MS datasets are typically continuous data. The datasets are obtained from several sources (available online or obtained from various MS labs, including VUW school of Biological Sciences proteomics lab). The datasets are carefully chosen to have a different number of features (365– 45200), as well as different number of instances (10– 253). We also used some datasets with predefined biomarkers to test the biomarker detection rate of the proposed approaches.

Mostly, MS datasets are binary classification problems (case/ control). In some cases, datasets involve three classes (case/ control/ treated) or four classes (case stage 1/ case stage 2/ treated/ control).

Table 1.1 summarises the datasets that are used as representative samples of the MS classification problems that the proposed approaches can solve with the explanation of the thesis chapters that used the dataset. In the table, the number of samples is shown in the third column, and the number of examples in each class is given between brackets. The last eleven datasets contain some predefined biomarkers that are spiked into the datasets during the preparation process. More details about the datasets acquisition are explained in the following chapters.

Table 1.2: Benchmark MS Datasets

Name of the Dataset	# Features	# Samples	Chapter
Ovarian cancer high-resolution (OVA1)	15000	216 (121+95)	chapters 3,4 and 5
Ovarian cancer low-resolution (OVA2)	15154	253 (162+91)	chapters 3,4 and 5
Premalignant pancreatic cancer (PAN)	6771	181 (80+101)	chapters 3,4 and 5
Arcene (ARC)	10,000	200 (100+100)	chapters 3,4 and 5
Detection of drug-induced toxicity (TOX)	45200	62(28+34)	chapters 3,4 and 5
Hepatocellular carcinoma (HCC)	36802	150 (78+72)	chapters 3,4 and 5
Detection of glycan biomarkers (DGB)	16075	128(78+25+25)	chapters 3,4 and 5
Prostate cancer (Pros)	15,000 (63+190+26+43)	322	chapters 4,5
Apple-plus	773	40 (10+10+10+10)	chapter 4
Apple-minus	365	40 (10+10+10+10)	chapter 4
DS_a porcine CSF	9889	10 (5+5)	chapter 3
DS_b human urine	29529	10 (5+5)	chapter 3
DS_c human urine	29529	12 (6+6)	chapter 3
DS_d human urine	29529	24 (12+12)	chapter 3
DS_e human urine	29529	30 (15+15)	chapter 3
DS_f human urine	29529	12 (6+6)	chapter 3
DS_g human urine	29529	24 (12+12)	chapter 3
DS_h human urine	29529	30 (15+15)	chapter 3
DS_i human urine	29529	30 (15+15)	chapter 3

Chapter 2

Literature Review

This chapter is divided into five main parts. The first part explains the basic background of mass spectrometry technology, while the second part describes the background of machine learning, feature manipulation and classification algorithms used throughout the thesis. The third part gives a description of evolutionary computation (EC) algorithms including genetic programming and other common EC algorithms. The fourth part of the chapter includes a description of multi-objective optimisation and its most common algorithms. The fifth part of the chapter discusses the previous work on biomarker detection of mass spectrometry data and points out the main limitations of previous approaches. The fifth part also explains the related work of feature selection and construction done using different algorithms specifically genetic programming. The chapter ends with a summary.

2.1 Mass Spectrometry

Mass spectrometry (MS) offers high throughput analysis of the biological samples by determining the elemental compositions of these samples [169, 177]. The mass spectrometer measures the molecular masses of proteins or peptides, and these masses can be used for identification of the

compounds [103]. It is composed of three parts, which are the ionisation source, the mass analyser and the detector. Firstly, the proteins or peptides are ionised in the ionisation source. Secondly, these ionised molecules are analysed by the mass analyser, which measures their mass to charge ratios (m/z). The third part is the detector, which counts the ions for each m/z value and produces the spectrum.

There are two working modes for the mass spectrometer. The first is the full mode, which measures the m/z values of the parent ions. The product spectrum is called MS spectrum. This spectrum is composed of the m/z ratios of the ions and their corresponding relative intensities. An example of the MS spectrum is shown in Figure 2.1. The second is the tandem mode, which fragments the parent ions and measures the m/z ratios of the fragment ions. The produced spectrum from the tandem mode is called MS/MS spectrum [103].

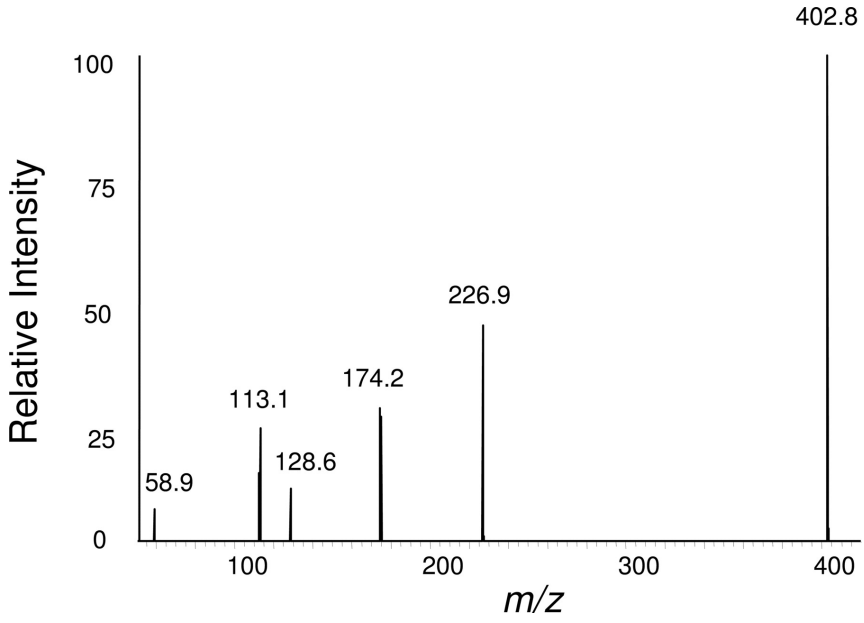


Figure 2.1: Mass Spectrum example [95]

To make the spectrum less complex, a separation procedure is often performed on the samples prior to the mass spectrometer. This separating method can be a liquid or gas chromatography (LC or GC). The result-

ing spectrum is called the LC-MS or GC-MS spectrum that contains the retention times, the m/z ratios and their corresponding intensities [169].

The MS technique is the dominant technique used nowadays for the identification and quantification of the molecules in biological samples, leading to the elucidation of their chemical structures and the discovery of biomarkers in biomedical research due to its high sensitivity, high accuracy and high-throughput capability [20].

MS can be applied to proteomic or metabolomic research areas. In proteomics, the first objective is to quantify the proteomes that offers better understanding of cellular and the structural [11, 167] mechanisms. The second objective is to identify the quantified protein. The third objective is the biomarker discovery, where biomarkers are the molecules that indicate specific biological states linked to pathogenic processes, or pharmacological responses to a therapeutic intervention. In the metabolic research area, the role of MS can be extended to the biochemical reactions and biomarker discovery characterizing physiologically important metabolites.

However, running MS is a time-consuming process, and it is not practical to produce a large number of samples. Also, each sample typically has a huge number of features (as many millions of features) [20].

2.1.1 Proteins and Peptide

A protein is a molecular compound that consists of a chain of amino acids. When two amino acids of the proteins are linked together through the peptide bond, they form a dipeptide. More amino acids linked together are called polypeptide. Very large polypeptide chains form the proteins [125].

Proteins are the governors or the controllers of the cells which have a variety of cellular functions. Some proteins (e.g. enzymes) are responsible for determining which reactions take place. Others have their role in signaling and transport. Structural proteins form some elements that make

the cells maintain its shape and size [149]. Also, motor proteins generate mechanically forces. Furthermore, the presence or absence of proteins are not only important to the state of a cell, but also changes in the abundance of these proteins can make the discrimination between healthy and diseased cells.

2.1.2 Metabolome and Metabolite

A metabolite is any substance involved in metabolism or the metabolic process, i.e. the process including a set of chemical reactions that changes a molecule into another either for storage or for use in another reaction or as a by-product [179]. The metabolome encompasses a large variety of components including lipids, amino acids, organic acids, nucleotides, steroids, vitamins, sugars, etc. Metabolome is the complete set of Small-Molecule metabolites. Figure 2.2 shows the relationships between genome, proteome and metabolome.

2.1.3 Mass Spectrometry-Based Proteomics

Proteomics is the process of exploring the whole proteome that is the entirety of the proteins and its modifications [167]. Unlike the genome of the organism, the proteome constantly changes over time. For example, the proteomes of healthy and patient people are different. Proteomics aims at an analysis of the changes of the different states of the proteome. This analysis will help biologists gain more knowledge about the functions of the protein and will also help medical scientists develop agents and methods to cure diseases.

MS is a key technology for high-throughput protein analysis. MS has achieved great progress toward the identification, quantification, and characterization of the proteins that constitute a proteome [20].

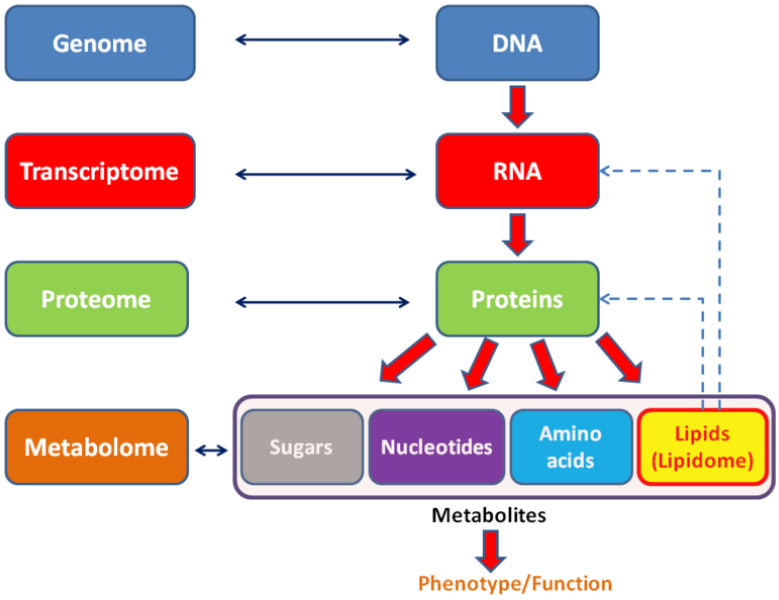


Figure 2.2: General schema showing the relationships starting from Genome to Metabolome. Image published under free document license in the Wikipedia (http://www.wikipedia.com).

2.1.4 Mass Spectrometry-Based Metabolomics

Metabolomics is the set of chemical reactions that occur in cells that are organised in the metabolic pathways where the chemical is transformed into another chemical through a series of steps according to this pathway or map [37]. Mass spectrometry provides a powerful tool for metabolomics analysis which can guide researchers to the metabolic states and provide vital information in the decision-making step for target identification and validation in drug research.

2.1.5 MS Research Directions

MS research has four main directions:

- Proteins and peptide identification: the first method of identification is the peptide mass fingerprinting which uses the masses of

the peptides to search a database of predicted theoretical masses that would have been produced from digestion of known proteins [191]. The second method is the De-novo (peptide) sequencing that is performed without prior knowledge of the amino acid sequence [69,105,166]. This process works by assigning amino acids from peptide fragment masses of a protein [163].

- Proteins and peptide quantification: determining the quantity of a specific protein in the sample is another area arising for MS. Protein quantification through MS can be done in two methods chemical labeling or label-free methods [185].

  Using the labeling methods [19], a mixture of proteins are labeled with an isotope that is used to compare the peaks directly from different samples. The labeling methods include SILAC (Stable Isotope Labeling with Amino acids in Cell culture) [127] and ICAT (Isotope Coded Affinity Tags) [65]. These methods provide accurate protein quantification but are considered expensive as they require additional processing steps and high-cost labeling agents. The label-free methods depend on the signal intensities (detected peaks for each peptide) to measure the abundance of the peptide. Label-free quantification has two possibilities, either extracting the ions from the spectra or the spectral count, i.e. the number of MS spectra in which peptides of an analysed protein can be found.

- Biomarker detection and classification algorithms: biomarker detection and classification algorithms are related to the process of finding the peaks or the feature patterns (biomarkers) that can be used to classify samples and discriminate different classes, e.g., from different cell states or from healthy and control classes. Biomarker discovery usually starts with feature selection to overcome the high dimensionality problem. The selected features are passed to a classifier to

assess the classification accuracy of the selected features. The set of features that provide better classification are the biomarkers.

- Peptide detection for biomarker verification and protein quantification: after biomarker detection, the detected biomarkers pass through two other stages, verification and experimental validation. Peptide detection, which is the probability that a the certain peptide can be observed in a mass spectrometer, can be used as a verification method of candidate biomarkers. The process of peptide detection can also be used for absolute quantification of peptides and proteins. The task of peptide detection can be performed using machine learning techniques by training a model using a peptide sequence's properties. The peptides of the observed class are classified as the verified biomarkers. Peptide detection is the middle stage between biomarker detection and experimental validation.

2.1.6 Biomarkers

The term "biomarker" can be defined as a biological marker or a medical sign that acts as an indicator of a specific medical state. Biomarker refers to the indications of a specific illness or non-illness of a patient. In the literature, there are more precise definition of biomarker for example, the National Institutes of Health Biomarkers [59] Definitions Working Group defined a biomarker as "a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic responses to a therapeutic intervention." The World Health Organization (WHO), and in coordination with the United Nations and the International Labour Organization, has defined a biomarker as "any substance, structure, or process that can be measured in the body or its products and influence or predict the incidence of outcome or disease" [158].

Biomarker can also indicate the effect of therapies and drugs that can be extended to discoveries of new drugs and prognosis of diseases. A biomarker is a protein, peptide or a metabolite which occur in a biological sample.

From a machine learning point of view, a biomarker is an important feature that has a higher capability to discriminate between the different classes. Hence, the set of features that provide better classification are the biomarkers. In MS spectrum, each instance is composed of a number of features where each feature refers to a certain peptide or metabolite. The feature is composed of the feature identity which is the m/z value, and a feature value which is the intensity value. The intensity values are used in the feature manipulation algorithm to determine the biomarker candidates. This process is referred to as the biomarker detection process. The m/z value of each biomarker is used afterwards to identify what protein or matabolite this biomarker is.

2.1.7 Mass Spectrometer

The general setup of a simple mass spectrometer is composed of three parts: The first part is *ion source* in which the mixture of molecules is ionised to facilitate the separation process of these ions according to their mass to charge (m/z) ratio measured using the second part which is *mass analyser*. The third part is the *detector* which counts the resulting ions for each mass [167]. The structure of the mass spectrometer in shown in Figure 2.2. There are different techniques of ionisation and mass analysis.

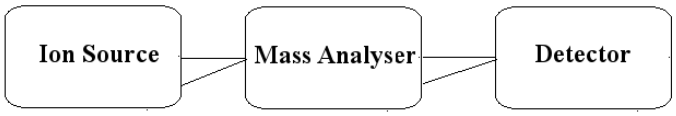


Figure 2.3: Mass spectrometer structure [167].

ionisation techniques

ionisation techniques include matrix-assisted laser desorption ionisation (MALDI), surface-enhanced laser desorption/ionisation (SELDI) and electrospray ionisation (ESI) [165].

For MALDI, the analyte has to be mixed with a matrix substance. The mixture is dried out and crystallized then it is hit with a laser in the vacuum. At a specific time, the matrix absorbs the energy and blows up producing the ionised molecules in the process [12, 168]. Most of the MALDI ions are only singly charged, which makes MALDI spectra easy to interpret and analyse. SELDI is a variation of MALDI that uses a target modified to achieve biochemical affinity with the analyte compound [50].

The second method of ionisation is the electrospray ionisation (ESI) [112], where the analyte is mixed with a liquid. ESI is usually coupled with LC to separate the mixture and make the spectra even simpler before the MS analysis. The analyte solution passes through an electrospray needle to release the ions. Figure 2.4 shows ESI schematics.

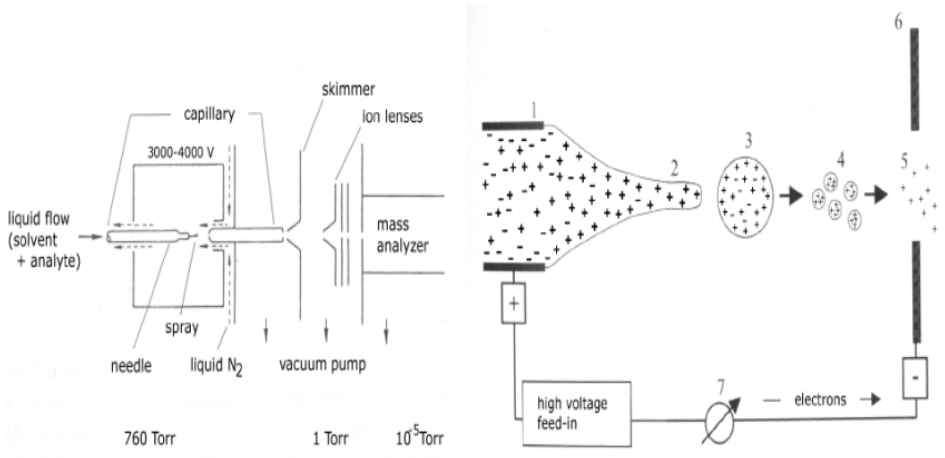


Figure 2.4: Electrospray ionisation (ESI) schematics [58].

Mass analysers

The mass analyser helps in determining the mass range, sensitivity and accuracy of the instrument [167]. There are different types of mass analysers, the most common including [186]:

- **Sector field mass analyser:** uses electric or magnetic fields for deflections, where the lighter, more charged, faster ions are more deflected [54].
- **Time-of-flight (TOF):** TOF accelerate the ion produced by the static electric field and then measure the time they need to reach the detector [167]. Lighter ions reach the detector first.
- **Linear quadrupole ion trap:** This mass analyser uses electrical fields to stabilise or destabilise ions passing through a radio frequency quadrupole field. Using this approach, certain ions are trapped in a two-dimensional electrical field and can be selectively discarded from the trap by their m/z value [186].
- **Fourier Transform Ion Cyclotron Resonance (FT-ICR) mass analyser:** This analyser makes ions move circularly with a homogeneous magnetic field. The frequency of rotation depends on the m/z of the ion. To measure ions with different masses the alternating field is changed, and the signals for different masses retrieved via Fourier transform (FT) [186].
- **Orbitrap:** This is perhaps the most recent mass analyser [186]. Orbitrap takes up a circular motion (orbit) through electrostatic attraction. At the same time, it oscillates along the axis of the central electrode. This oscillation generates signals in the detectors that can be mapped to mass-to-charge ratios by FT. In contrast to the FT-ICR, an orbitrap uses an electrostatic rather than a magnetic field. Therefore, no cooling is necessary. The resolution is nearly as good as that of an FT-ICR [54].

Detectors

The mass analyser destroys the ions during analysing, so the detector detects a particle and then multiplies the effect of this particle as the number of detected ions is often very small [36]. Possible detectors are a photomultiplier, secondary ion multiplier, ion-to-photon detector, Faraday cup, channel electron multiplier, or Daly detector [36].

2.1.8 MS Data Analysis

The pipeline of the MS data analysis consists of the following steps [97]:

- peak signal filtering and baseline subtraction: remove noise and baseline artifacts that have resulted from instrumental, chemical or biological errors.
- peak extraction: detects and extracts the accurate positions, heights and total ion counts of all the peaks.
- identification of the compounds (qualitative MS): identify the compounds in a sample given peak information. For protein identification, the MS takes the peptide mix as an input and produces a list of masses which are used as indexes of the corresponding proteins in a database of known proteins. The list of the masses is called the peptide mass fingerprint (PMF). During the search the matching proteins, the modified or unmodified peptides and their masses together with scores per peptide and protein, and the sequence coverage are given as results. This approach can work with a certain complexity, but if there are many proteins in a sample, this approach is no longer appropriate. Therefore mechanisms for fragmentation of peptide are applied (tandem MS/MS) that allow prediction [191] and de-novo sequencing [69, 105, 166] from fragmentation spectra, making the identification of proteins in very complex samples possible.

- quantification (quantitative MS): determines the quantity of each compound in the samples. It is insufficient to know which proteins in the cell, it is also important to know how abundant these proteins are to be able to differentiate between normal and abnormal cells. MS enables high-throughput ability to identify and quantify proteins from the same sample. However, it is not easy to make it work quantitatively because the measuring sensitivity differs between different types of molecules [97, 167].

- intensity normalisation: normalise the ion counts of the compounds.

- multiple map alignment: correct the distortion and the fluctuation of the retention time and m/z dimension of multiple raw or feature maps in case of tandem MS. In case of MS, correct the fluctuation in m/z values.

- classification and biomarker discovery: find the feature patterns (differentially expressed peaks) that can be used to classify samples [97, 167]. A biomarker is the indication of a change in expression or state of a protein or a metabolite that correlates with a risk of disease progression. Biomarker discovery will lead to drug discovery and development, indicate drug efficiency, reduce cost and duration of experimental trials and finally, can help in the cure of diseases.

  The use of MS technique for the identification and quantification of the molecules in biological samples will lead to the elucidation of their chemical structures and the discovery of biomarkers in biomedical research due to its high sensitivity, high accuracy and high-throughput capability [97].

  Depending on the underlying type of mass spectrometer, a raw MS spectrum or LC-MS/MS map, can vary from several hundreds of megabytes up to several gigabytes, whereas only a small fraction of data contains the signal of interest. This accentuates the need for fast

and effective machine learning algorithms for each of the analysis steps mentioned above. This will allow for high throughput, fast proteomics and metabolic approaches.

2.2 Machine Learning

Machine learning is the research field dealing with algorithms for automation of knowledge from data [8, 18].

Two main approaches to machine learning are supervised and unsupervised learning [8].

In supervised learning, the learning algorithm learns to produce the correct outputs from given inputs. A typical example application of supervised learning is classification in which a classifier takes, as input, the description and properties of several examples of an object and produces, as an output, a class label for that object [119]. The classification process consists of two phases, training and testing where, in the training phase, the classifier has been trained by or learned from examples from the problem called instances accompanied by the class labels, where the whole set of instances is called the training set [119]. In the testing phase, the instances that are unseen by the classifier are used to test the performance of the classifier. The collection of instances used during the testing phase is called the test set [119]. Another example of the supervised learning is regression, i.e. the estimation of a function or modelling the problem solution with equations [8].

In unsupervised learning the examples are unlabelled, which means that during the training process the learning algorithm cannot use class labels. An example of unsupervised learning is clustering. A main challenging problem of clustering is the evaluation of the clustering algorithm [8].

2.2.1 Classification algorithms

Tree-based Classification algorithms

The tree-based classifiers use a tree or a series of trees as a learning method to differentiate between classes and to build the predictive model [8]. Examples of tree-based classifiers include decision trees (DT), Random Forest (RF) and Naive Bayes Tree (NB-Tree). DT classifiers follow the decision tree learning method (C4.5) [8]. DT's leaves represent class labels and its branches represent features that lead to these class labels [8].

RF constructs a multitude of decision trees for training [21] while NB-tree uses Naive Bayes classifiers at the leave nodes of a decision tree [89].

Non-tree-based Classification algorithms

The Non-tree based classifiers include Bayes classifiers, function classifiers, Nearest Neighbour classifiers and rule-based classifiers. Bayes classifiers make a probabilistic approach to classification, which assumes that the input-output relationships can be represented as probability distributions [8, 71]. NB is a probabilistic classifier based on Bayes theorem. NB makes an assumption that all the input features are conditionally independent [71].

Function Classifiers are the classification algorithms which depend on a certain function for building the predictive model. Examples are Support Vector Machines (SVM), Neural Networks (NN) and Voted Perceptron (VP).

SVMs construct hyperplanes in a high dimensional space and classify examples based on the side of the hyperplanes they fall on [71]. SVMs tend to maximize the distance between these hyperplanes where the points are fixed support vectors. The machine that uses that the hyperplane is called the support vector machine [71,172]. NN classifiers work by transforming the information through layers of a network. The network acts as a function that maps the instances or observations to the target class labels [24].

VP is based on the perceptron algorithm and uses kernel functions to build hyperplanes as decision boundaries [43].

In Nearest Neighbour classifiers, the output class is the class of the nearest training example. Finally, rule-based classification algorithms depend on the IF-then rule for building the classification model. Examples of rule-based classifiers are Decision table (DT), where a 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 [88]. Another example is Conjunctive Rule (CR), which builds a single conjunctive rule to predict the class labels. It uses the "AND" logical operator to determine a correlation between features and classes [180]. The OneR classifier performs classification like a 1-level decision tree [75]. The CART classifier generates partial decision trees several times to infer rules [41].

2.2.2 Feature Manipulation

Feature manipulation consists of mainly feature selection and feature construction. Feature ranking goes under the umbrella of feature selection while feature extraction goes under the umbrella of feature construction [106].

Feature selection is the process of selecting a subset of the original, relevant features and neglecting the redundant or irrelevant features [106]. However, feature construction constructs a new set of high-level features.

There are three approaches for feature manipulation: the wrapper approach, the filter approach and the embedded approach [8].

Wrapper Approach

The wrapper approach uses a learning algorithm in the search process of feature selection or construction. It depends on the learning algorithm

(classifier) to evaluate the selected or constructed features. Its evaluation is done without the knowledge of the structure of the regression or the classification function. It can therefore be wrapped to any learning machine [117]. The evaluation depends on the classification performance of the candidate solutions to find the best subset of features, which performs best for classification. Usually, the wrapper approach is computationally more expensive when the number of features is large as it involves training a classifier for evaluation and accordingly requires a large search space [106].

Filter Approach

The filter approach is based on performance evaluation calculated directly from the data. It does not depend on any learning algorithm [117], which is the main difference between the filter approach and the wrapper approach. The filter approach typically uses a relevance measure, which is often the measure of correlation between the features and the class labels, or generally between the features and target values (outputs) [117]. For feature ranking, this relevance measure can be calculated for each individual feature, providing a rank for each feature. The features of the lowest rank are removed [63]. However, the ranking of individual features is only useful if the features are independent of each other. If the features are correlated, some low-ranked but important features (when contaminating with other features) might not be taken into consideration and, therefore will not be selected [123]. This makes filter approaches less effective as it does not consider the relationship between the features.

Embedded Approach

In an embedded approach, the learning and the feature selection or construction mechanisms interact with each other, which is the main difference between embedded methods and other feature manipulation approaches.

Meanwhile, unlike the wrapper approach, the embedded approach does not separate the learning from the feature selection or construction processes. Instead, it determines the features and the classifier simultaneously during the training process [63]. Examples of the embedded methods include decision trees, where the tree is built by partitioning the data according to the importance of the features to the classification accuracy. Genetic programming is also classified as an embedded approach as it has an intrinsic capability of selecting or constructing features, which can improve the classification accuracy [132].

2.3 Evolutionary Computation

Evolutionary Computation (EC) is a branch of artificial intelligence, which consists of evolutionary algorithms, swarm intelligence and others. Evolutionary algorithms (EAs) are heuristic optimisation algorithms that are inspired by the natural evolution and Darwinian principles, e.g. reproduction, selection, crossover and mutation. EAs are characterised by having a population of solutions where, each candidate solution is an individual. The evaluation of the individuals is done by means of a fitness function. In the evolution stage, selection of individuals is performed, and different genetic operators are applied to the selected individuals. Examples of EAs algorithms are genetic algorithms (GAs) [74] and genetic programming (GP) [92].

Another important branch of EC is swarm intelligence including, particle swarm optimisation (PSO) [84] and ant colony optimisation (ACO) [32]. A brief description of GAs, PSO and ACO is provided below. GP will be described in detail in the next subsection since it is directly used in the thesis.

2.3.1 Genetic Algorithms

Genetic algorithms (GAs) [74] are one of the first evolutionary algorithms which has adopted the process of natural genetic evolution [55]. The population in GAs is encoded as chromosomes, where each chromosome is represented as a series of fixed length of bits (0s and 1s) [118]. Similar to other EC techniques, GAs use the genetic operators to evolve the populations during the search for the solution. The main difference in the use of genetic operators in GAs is that the mutation is applied to one chromosome, while the crossover is applied to two chromosomes. GAs have been successfully applied to a variety of applications such as pattern recognition [159], image processing [131] and bioinformatics [128].

2.3.2 Particle Swarm Optimisation

Particle swarm optimisation (PSO) [84] resembles the evolutionary computation algorithms in many features. It uses a population of individuals encoding the solutions of the problem. These individuals are manipulated according to the survival of the fittest rule. However, PSO and other swarm intelligence methods are inspired by social behaviours [83]. PSO resembles the social behaviour of flying birds. It does not use genetic operators like in GAs. However, individuals evolve by cooperation and competition between the individuals. Each individual in the swarm is called a particle. The particles search for the optimal solution by flying in the search space according to its flying experience and its neighbour particles [151].

2.3.3 Ant Colony Optimisation

Ant colony optimisation (ACO) [32] is inspired by the social behaviour of ants which aims to find the optimal and shortest path to the food and the colony [33]. Individuals of the population are represented by ants. The

ants of the population make a path for other members of the colony to follow by depositing a pheromone on the ground. The path that has the most pheromones is produced as the best-designed solution.

2.3.4 Genetic Programming (GP)

The GP algorithm is inspired by the natural biological evolution, where the phenotype of GP is the genetic program. The genotype of a GP program is the representation of this program, which can vary according to the type of GP. In the tree-based GP (which has also been used in this chapter), each individual is represented by a tree consisting of a set of nodes of terminals and functions, and only the fittest individuals pass directly to next generations. Groups of individuals form the population, where the evolution of the population is driven by selection and mating [92, 123].

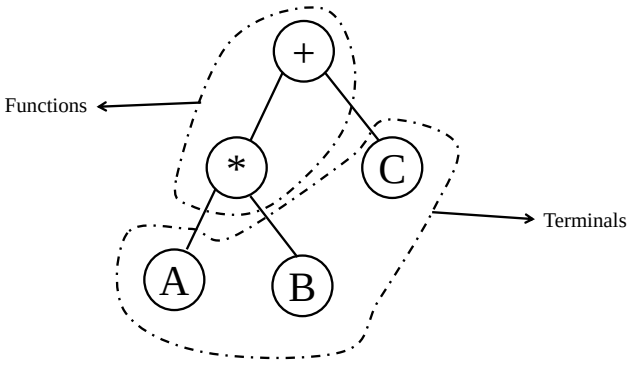


Figure 2.5: A GP Example that represents the program (A*B)+C.

Program Representation

A tree-based GP program is represented by a tree where the tree nodes are functions and terminals. A function node performs an operation and its child represents the arguments of this function, while a terminal node has no children. The terminal nodes are either variables, representing the

input values to the programs, or constant values which are randomly generated [136]. Figure 2.5 shows an example of a tree based genetic program representing the program (A*B)+C, where the + and * are the functions and the A, B, C are the terminals.

The main framework of GP consists of the following steps [123, 136]:

- An initial population is generated by a ”random” generation of individuals using the predefined function and terminal sets.
- Each individual of the population is evaluated and assigned a fitness value.
- Generate iteratively new populations according to the following steps until the stopping condition is met:
  1. Select some of the individuals using a specific selection procedure;
  2. Apply genetic operators to the selected individuals to produce new individuals;
  3. Put the new individuals in the next generation;
  4. Evaluate the new individuals, using the fitness function;
- When the stopping criterion is met, the best program is used as the solution to the problem.

Initialisation of the Population

Like all other evolutionary computational algorithms, GP starts the search by a randomly generated initial population. The methods of generating the initial population are either "grow method", "full method." or the "ramped half-and-half method" [92, 123].

In the "grow method," all the nodes can be selected randomly from the function and the terminal sets as long as the maximum depth is not

reached. When the maximum depth is reached, the node must be selected from the terminal set. This not the case in the ”full method”, where the nodes less than the maximum depth are all functions, not terminals. The "ramped half-and-half method" combines these two approaches, where half of initial the population is generated using the grow method, and the other half is generated using the full method. The main purpose of the ramped half-and-half method is to deliver a diverse population [92, 123].

Evaluation of Individuals

The evaluation determines how good an individual or a program can solve the problem. It is measured through the fitness function. The fitness function can be designed according to the task or the problem. For example, it can be the classification accuracy (the number of correctly classified examples/total number of examples) in the classification task or the error rate or the amount of time it takes to solve the problem (if the problem is related to time). The fitness of an individual determines the probability of its selection for the genetic operators [92].

Selection of Individuals

Selection provides the method for selecting the individuals who will pass to the mating pool. There are different types of selection methods that can be used in the GP algorithm, e.g. the roulette wheel selection (” Fitness Proportional Selection”) where the individuals are randomly selected based on their fitness in which the fitter individual has more chance to be selected. Another popular selection method is called tournament selection in which a specific number of individuals are picked up randomly from the population according to the size of the tournament, and the fittest individual is selected. A big tournament size gives less probability to select bad individuals. If the size is equal to one, the process will have a higher randomness percentage, as the fitness will not be considered [136].

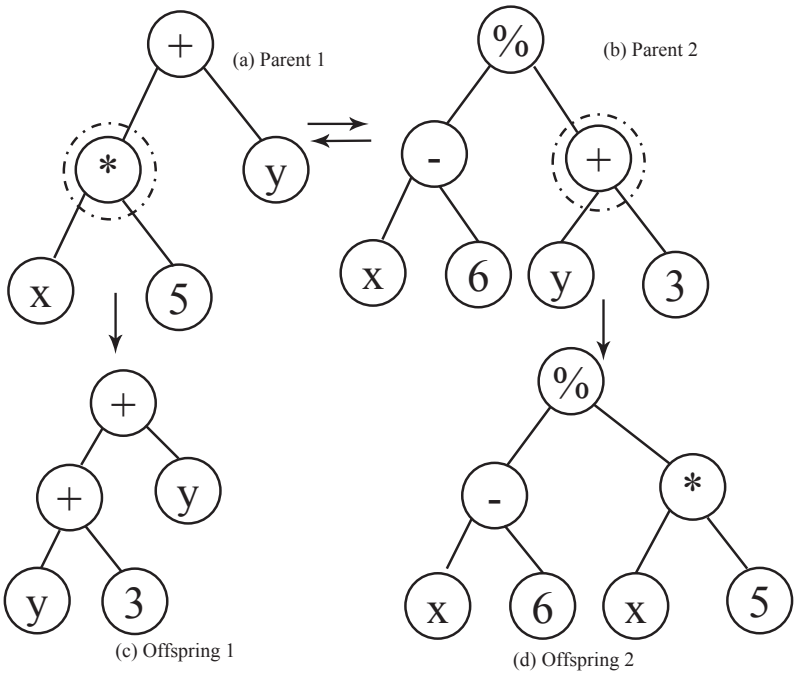


Figure 2.6: Subtree Crossover. (a) and (b) are the parents where the highlighted node is the randomly selected node and the subtree rooted from it is switched with the other subtree to form the two offspring in (c) and (d).

Genetic Operators

Genetic operators are needed to perform changes to the populations' individuals so that new children passed to the next generation are not the same

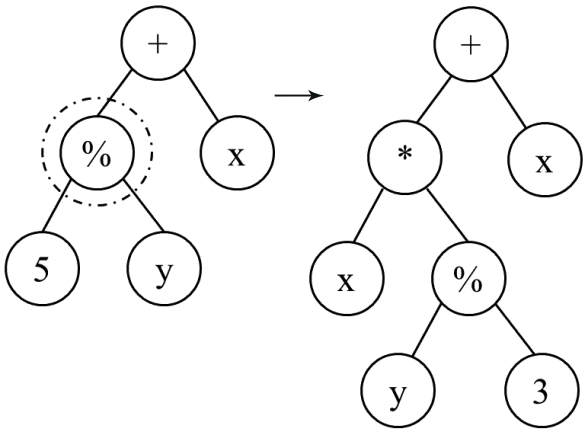


Figure 2.7: Subtree Mutation.

as those in the current generation. The basic genetic operators include reproduction, crossover and mutation. The application of each of these operators is determined by the rate or the probability assigned to each of these operators. The *reproduction* operator simply copies the selected individuals by the selection method of the next generation. The *crossover* is performed on two parents that are selected by the selection method used. A node is selected randomly from each parent as a crossover point and then swapping a subtree in each of the parents at the crossover point takes place [92]. This concept is represented in Figure 2.6 by an example where (a) and (b) represents the two parents (individuals) selected by the selection mechanism. The crossover point is selected and the subtrees under the crossover are exchanged to form the new children (offspring at Figure 2.6 (c) (d)). The *mutation* operator is performed on one individual at a time where a random subtree is selected and it is exchanged by a randomly generated subtree. Another type of mutation makes a condition that only a terminal replaces a terminal and a function replaces a function. An example of the mutation is shown in Figure 2.7. Besides these genetic operators, the *elitism* operator is usually used in GP to select the top or the elite individuals of the population and copies it to the next generation to make sure that the best individuals are not lost during the random selection mechanisms [92, 123].

2.4 Multi-objective Optimisation

When two or more conflicting objectives occur, and an optimal decision needs to be taken. This results in a multi-objective problem [126]. Multi-objective optimisation solutions are evaluated in terms of the trade-off between the conflicting objectives which can minimised or maximised [56].

A single objective optimisation problem can be represented mathematically as (if the problem is minimisation) [56],

$$minf(x), x \in C \tag{2.4.0.1}$$

where C is a set of constraints.

Multi-objective optimisation is formulated as

$$min(f_1(x), f_2(x), ..., f_n(x)), x \in C \tag{2.4.0.2}$$

where x is a feasible solution, n is the number of objectives ($n > 0$) and C is the set of constraints. There is no feasible solution that can minimise all objectives simultaneously. This introduces the need for Pareto optimal solutions [126].

If a solution is not worse in all objectives and it is better than the another in at least one of the objectives, it will dominate this solution [192].

Pareto optimal contains the set of non-dominated solutions where, a specific solution cannot improve any of the objectives without degrading at least one of the other conflicting objectives [126].

The non-dominated solution forms the Pareto front in which no solution can be judged to be better than others. Figure 2.8 shows an example of the Pareto front when two objectives conflicts with each other and the first objective tends to be minimised while the second objective tends to be maximised.

2.4.1 Common Multi-objective Optimisation Techniques

Evolutionary algorithms (EAs)have been widely used to produce the Pareto set. This is because they can produce multiple Pareto-optimal solutions in a single run. EAs may make use of similarities of solutions by recombination of these solutions.

The most common evolutionary multi-objective optimisation techniques include the following:

Non-dominated Sorted Genetic Algorithm (NSGA) [27] is an extension of genetic algorithms for multiple objectives optimisation algorithms. The

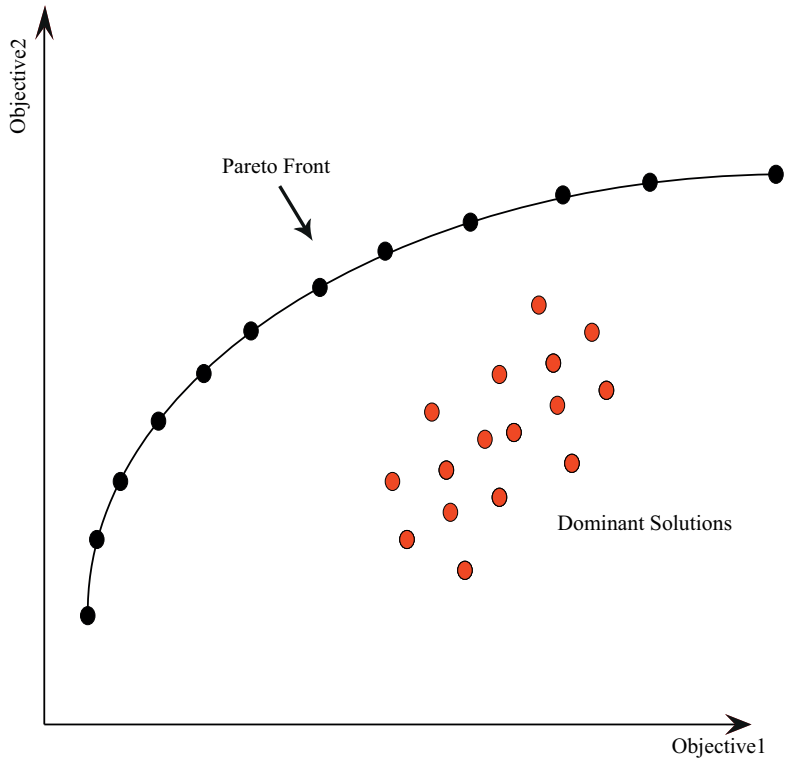


Figure 2.8: Pareto dominance example

algorithms use the evolutionary process to improve the fit candidate solutions in the Pareto front. The population is sorted according to the Pareto dominance [27].

NSGAII [27] extends NSGA, which was proposed to overcome the high computational cost that occurs due to the non-dominated sorting in each generation. NSGAII also introduces elitism to the algorithm that helps to speed up the technique and prevents the loss of good solutions. The dominance rank of a solution S_i is used for evaluating the fitness, i.e. the number of other solutions in the population that dominate S_i. NSGAII tends to minimise the fitness, i.e. a solution in the Pareto front will have the best fitness of zero [27].

Strength Pareto Evolutionary Algorithm (SPEA) uses the population and an external archive to store the non-dominated solutions [193]. The

archive has a specific size limit and a clustering algorithm is used to remove the further non-dominated members, and keeps the solutions which preserve the characteristics of the front. The fitness value is assigned to the archive afterwards and the population solutions and then comes the mating selection phase.

The problem that face SPEA is the fitness assignment, where all the individuals that are dominated by the archive solutions have the same fitness. As a result, if the archive has one solution then all the population individuals will have the same rank. The selection of this individual in the archive is therefore decreased, making SPEA, in this case, act as a random search algorithm. Another limitation of SPEA is the density estimation that is used only during the clustering algorithm. This means that it is used in regards of the archive. Finally, the clustering algorithm can lose the outer solutions which are necessary to keep a good spread of the non-dominated solutions [193].

SPEA2 [192] has solved some limitations of SPEA. SPEA2 uses a fine-grained fitness assignment, truncated density estimation, fixed archive size and it does not use a clustering algorithm. In SPEA2, the individuals of the archive are the only individuals used in the mating selection phase [192].

To evaluate the Pareto dominance, SPEA2 uses both dominance rank and dominance count. Dominance rank of a solution is the number of the solutions that dominate this solution, while the dominance count is the number of solutions that a given solution dominates. A solution with a smaller number of solutions that dominate it (lower rank) and a higher count is a better solution.

2.5 Related Work

The MS data analysis for the biomarker detection task can be divided into the following goals [103]:

- Low-level biomarker discovery: discovering the MS peaks that are responsible for differences between classes of diseased and normal persons. This process is typically performed through feature manipulation.
- High-level biomarker discovery: identifying the proteins, peptides or the metabolites which correspond to the peaks.
- Verification of the candidate biomarkers: measuring the detectability of the peptides in the Mass spectrometer.
- Validation of the verified biomarkers: passing the set of verified biomarkers to the experimental validation.

2.5.1 Statistics and Machine Learning for Biomarker Detection on MS data

Statisticians take a different view of the low-level biomarker discovery task or finding the discriminating features. Typically a t-test, Wilcoxon rank test or any suitable statistical test is done on each feature independently to determine the score by which each feature discriminates between different classes [103]. After this, an assumption is made that the features are not discriminative, and the distribution of these test scores is modelled. This distribution is commonly simulated by using a permutation test. The permutation test is usually performed on one feature at a time, which will result in thousands of significance tests (measuring the p-value). This approach was used in [104, 162, 188] after preprocessing. The limitation of selecting features depending on the p-value was analysed and explained in [110] with details. Using only the statistical approaches can result in redundant features, and hence these biomarkers might be a result of noise and not real biomarkers.

Machine learning approaches for biomarker detection have been used for data classification and frequently a feature selection step is used to

improve the classification performance [91]. [133] performed a predictive study on ovarian cancer data where they used genetic algorithms to find a subset of features (m/z values), which could improve the classification accuracy. The selected features were used to classify the new samples. The evaluation was done according to the ability of the features to form two clusters with the correct membership in the class [103]. A similar approach was used by [10]. The optimal final set of features in these two studies was formed from the original input space (features) and the final classifier was a linear one. The approaches were only tested on one dataset which might not be generalised to other datasets. Several studies have been done afterwards that obtained near perfect classification accuracy on some different MS datasets [148,156]. However, the main problem was the reproducibility of the results which remains limited.This means that the biomarkers of these studies on the same datasets are different, and hence these biomarkers might be untrustable. [100] used T-test and genetic algorithms for feature selection in conjunction with SVMs as a classifier on three SELDI datasets. In [100], the features selected by genetic algorithm (top 10 features) performed better than T-test, which suggests that a high-order interaction between features can provide more powerful discrimination. The main limitation of this algorithm is the high computational cost of using the wrapper approach. In [144], the author used principle component analysis (PCA), unfolded-PCA and partial least squares for the classification of the normal and the drug injected mice analysed using LC-MS. This was done after preprocessing and alignment of the data. [144] used an LC-MS dataset, which is composed of two samples in each class in the training set and one sample in the test set to perform the classification and biomarker detection through visualisation of the data. In addition to the disadvantage PCA of using the assumption that the dimensionality of data can be reduced by linear transformation, using visualisation of the data to detect the biomarkers might not be accurate.

2.5.2 GP for Biomarker Detection using Feature Selection

GP has been used for biomarker detection using feature selection in MS or LC-MS data few times. Only a small number of studies [35,53,76,109,187] have used GP to combine features for the production of good classifiers. These features have been indicated as the biomarkers. These studies have been only applied to gene expression data and it have not been investigated on MS data. GP was used in [187] to classify a prostate cancer MS dataset, which showed that GP used a smaller number of biomarkers than other classifiers with comparable classification accuracy. This is beneficial, as usually a smaller number of features to be detected is preferable in order to reduce the experimental cost required for validation of the detected biomarkers [181]. However, the approach has not been extented to more datasets and it has not been tested on datasets with predefined biomarkers. Hence, it is not proved that the biomarkers detected are the true biomarkers. GP was used in an early study [51] for the analysis and the quantification of the amount of sucrose in orange juice. In this study, the detection and quantification has been done by performing a regression process and the values estimated by GP were compared with the expected values of sucrose. This approach has not considered the power of GP for feature selection, as the algorithm was using GP for quantification of the missing values of the data and not for automatically selecting the biomarkers. In [3] the features selected by GP with different fitness functions achieved a certain level of success on MS and LC-MS datasets. Moreover, in [154] different feature selection metrics were used with GP to further select a smaller number of features. The features selected by GP were used and those features managed to improve the classification accuracy.

One hundred and six breast cancer patient samples were analysed using MALDI MS by [29]. These samples were analysed using GP for biomarker detection by selecting protein clusters that can correctly classify breast cancer best. GP was used by [52] to analyse the metabolic biomarkers aimed

at detecting changes in the levels of biochemical compounds by searching among thousands of biochemical compounds. In [82], GP was used to explore the metabolites (biomarkers) in specific plants by measuring the concentration of these compounds and finding the rules that discriminate these plants. LC-MS data was used in [82]where GP was used to investigate the function, and relation of a specific compound (salicylic acid) to the resistance of diseases in the plant immune system.

Although these studies has considered the use of GP for selecting the biomarkers, it has not fully used the adavantages of GP for discovering the relevant features.

2.5.3 GP for Feature Construction

GP has been used successfully for feature construction in two trends: 1) feature construction using attribute values for classification problems and 2) feature construction using raster graphics for object and edge detection problems. In the former trend, the constructed features are the results of scalar functions of original features [39]. In the latter trend that acts on images, the constructed features are the filters which operate on the raw images' raw pixel values [44, 45, 94, 153]. GP has also been widely used for feature construction [62,93,107,122] 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 [39] GP was used as an embedded approach to construct features, while in [61,62], the authors used a wrapper approach to construct important features. In addition, a filter approach was used in [107] to construct a single feature per class depending on the entropy measure.

These feature construction approaches (used for other applications other than biomarker detection) has either constructed a single feature, which might not be able to improve the classification performance of MS data due to the high percentage of noise and small training examples, or con-

struct features from multiple GP individuals (a feature for each class) which will increase the computational cost in case of MS data.

Feature construction has not been used before for biomarker detection despite the success of feature construction for improving the classification.

2.5.4 Other Evolutionary Algorithms for Biomarker Detection

Other EC techniques, such as GAs [137], PSO and ACO have been applied in biomarker detection using feature selection [115]. Typical EC methods for biomarker detection are briefly reviewed in this section.

Feature selection in GAs is performed by representing all the features in each individual. Each chromosome is represented by n binary bits if the number of features is n, where a 1 indicates that the feature has been selected. The best solution has the subset of features selected throughout the search process [78].

GAs have been applied to biomarker detection mostly as a wrapper approach. For example in [116], SVMs is wrapped to GAs to discover the biomarkers in gene expression data with the accuracy of SVMs as the fitness function. The algorithm is divided into two phases where the first phase the algorithm runs for a specific number of generations to select features and each feature is assigned a score. For the rest of the generations, each feature is assigned an average fitness score by dividing the total fitness score by the number of times that the feature was chosen in an individual. The algorithm is robust.However, the main issue here is the computational time of the wrapper approach. Another example is in [137] where the wrapped classifier is 1-NN, and the algorithm employs a procedure similar to bootstrapping to enhance the robustness of selected gene signatures (biomarkers). None of these algorithms were applied to biomarker detection in MS data where the amount of noise in this data significantly affects the feature selection process.

PSO has been widely used for feature selection with a high success rate [183, 184]. Only a limited number of studies, such as [115, 121, 141] used PSO for biomarker detection. In [115], a hybrid ACO/PSO algorithm is used to identify the set of biomarkers in SELDI MS data, while in [141] a simple PSO method wrapped to SVM is used for MS biomarker detection. A multi-objective PSO approach wrapped to artificial neural network classifier was applied to gene expression data to maximise both specificity and sensitivity. PSO can be a good choice for biomarker detection. However, it does not have the embedded capability of GP, which gives GP the power of combining the advantages of wrappers and filters.

2.5.5 Multi-objective Optimisation for Biomarker Detection

Multi-objective optimisation offers solutions to the optimisation of different conflicting objectives [126].

Biomarker detection must consider the trade-off between the classification performance and the number of features without the prior specification of the relative importance of each objective. The number of features should be as small as possible to be able to pass them to experimental validation. Therefore, for evaluation of biomarker selection, two objectives should be considered, maximise the classification performance and at the same time minimise the number of features.

There have been a limited number of studies that use multi-objective optimisation for biomarker selection in microarray gene expression data [42,56]. PSO was used in [121] for multi-objective optimisation in gene expression data as a wrapper approach, using an artificial neural network for evaluation. In [56], a Pareto Optimal approach (PO) with Analytical Hierarchy Process (AHP) was used to select subsets of features in microarray data for biomarker detection. In [42], a multi-objective genetic algorithm was used on metabolomics MS data to maximise the classification accu-

racy and minimise the number of features. None of the existing methods used a multi-objective embedded approach for biomarker detection.

2.5.6 Peptide Detection for Biomarker Verification

Computational approaches to prediction of peptides' observability in mass spectrometer were adopted to address the complexity of the laboratory methods. Decision trees [48] and artificial neural networks (ANN) [164] have been used to relate the physiochemical properties of proteins to their MS detectability. Evolutionary algorithms were also used in a small number of studies to solve the peptide detection prediction problem in MS data. For example, genetic algorithms (GA) [170] have been used to solve this problem where the aim was to reach the optimum experimental conditions for protein detection in MS. GP was used only in two studies [38,175].

The verification of biomarkers is a hard problem due to the high dynamic range of proteins [79], the complexity of the data and the lack of a universal bridging method. Early studies [48, 164, 170] have reported success for the use of machine learning techniques for measuring the peptide detectability. However, a complete understanding of the important properties necessary for peptide detection is lacking and a powerful method which provides a model that can be interpretable is needed. Furthermore, the peptide datasets are usually highly unbalanced, which means the number of peptides in the non-observable (majority class) is usually much higher than the number of observable peptides (minority class). This makes the classifier more biased to the non-observable class and mostly the high responding peptides are not going to be correctly classified.

Classification of unbalanced data can be characterized as either external or internal approaches [15]. External approaches create a balanced class distribution for training by transforming the original unbalanced data while keeping the learning algorithm unchanged [14].

The internal approaches utilise the unbalanced ratio in the training process and adapts the learning algorithm to deal with the uneven distribution of the classes [160]. This is done through cost adjustment in the learning algorithm. Other methods combine internal and external approaches [34, 129].

GP has been used effectively for classification of unbalanced data [14–16]. GP can be classified as an internal approach in which the cost adjustment is performed through using a fitness function that balances the fitness between the two classes [155, 178]. Despite the capability of GP to solve the various of problems involved with peptide detection (feature selection and classification of unbalanced data), the use of GP is very seldom investigated to solve this problem.

2.6 Chapter Summary

The MS provides means for biomarker detection that will be beneficial in many applications, including early detection of diseases and discovery of new drugs. The analysis of MS data for a biomarker detection is challenging due to the high-feature-to-sample ratio and the presence of noise in the data. Biomarker detection is performed through feature manipulation.

Due to the flexibility and capability of GP to automatically select and construct features, GP can be a promising choice for biomarker detection.

There has been a rapid grow in the MS biomarker detection research. However, there are still major open issues that remain to be investigated:

Feature Ranking in Biomarker Discovery: Most of the previous methods used a single ranking method to rank features and used the top ranked features for classification. However, different feature ranking metrics provide different ranks for the same features according to their evaluation criteria. None of these methods tested the use of more than one ranking metrics together. The collection of these ranking metrics can provide subsets of top ranked features and low-ranked features. This collection has

the potential to provide a better collection of features, leading to better classification results. In [147], GP was used to combine four metrics and it was applied successfully on datasets with a small number of features, but it was not tested on datasets with a large number of features as in the MS datasets. Also, using the single ranking schemes ignores the interactions and relationships between the features.

Feature Construction: Most of the GP based feature construction approaches were based on constructing a single feature, 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 [60, 107]. The second approach is therefore inappropriate for high dimensional data like MS data, where the number of features exceeds thousands. Moreover, none of these methods investigates the effect of constructing multiple features from a single tree during the evolutionary process of GP. Also, feature construction in MS data has not been considered before.

Multi-objective GP for Biomarker Detection: Multi-objective GP optimisation for MS biomarker detection using both feature selection and construction has not been considered before and, investigation of this direction needs to be carried out.

Biomarker Verification using Peptide Detection: The computational approaches for peptide detection are effective, but the limitation of not considering the imbalance problem of the peptides datasets lowers their sensitivity performance. Moreover, the previous approaches mostly considered selecting the features using ranking methods, which ignores the dependence and interactions between the features. Hence, considering GP for performing these multiple tasks is a worthing trial and can potentially lead to improving the process of detectability. Furthermore, verifying the

detected biomarkers using the peptide detection method needs more investigation.

The next four chapters propose new GP algorithms that can address the above issues.

Chapter 3

Ensemble Feature Ranking

3.1 Introduction

Feature selection is an important technique for biomarker discovery in MS data because many of the classification techniques cannot easily handle such a huge number of features. Feature ranking is a type of feature selection, where each feature is given a rank according to its relevance to the classification task [147]. Different feature ranking approaches usually give different ranks to the same features. Clearly, some of the top features may be highly relevant or powerful while other features may be weakly relevant or redundant [147]. Further selection and ranking of features based on sets of features produced by different feature ranking methods have the potential to provide a new and smaller set of features with less redundancy and more relevance to classification.

3.1.1 Chapter Goals

The overall goal of this chapter is to investigate the capability of GP for improving feature ranking performance. To achieve this goal a new ensemble-based feature ranking GP algorithm has been developed. The algorithm combines two well-known feature ranking metrics, namely information

gain (IG) and relief-f (RF), to select a new and smaller set of features. A new rank that can effectively improve the classification performance of the selected features is given to each of the selected features. Meanwhile, GP is used as a classifier as well, and the proposed algorithm takes an embedded approach. Specifically, we will investigate the following objectives:

- what ranking scheme is suitable for selecting good features;
- whether a small number top ranked features obtained by the proposed GP method can achieve better classification performance than using all the original features;
- whether the smaller top ranked features can outperform a relatively large number of the top ranked features obtained by IG and RF, respectively;
- whether different classifiers using the 20 top ranked features can achieve better performance than the 100 top ranked features obtained by IG and RF, respectively.

Chapter Organisation: The rest of the chapter is organised as follows. The second section describes the new ensemble GP method for feature ranking in MS data. Experimental design are presented in the third section. The fourth section descries the datasets and preprocessing. The fifth section presents the results of discussions. Some further discussions are presented on the sixth section. The seventh section gives a summary of the chapter.

3.2 Ensemble Feature Ranking GP Algorithm

Different criteria are used in different feature selection metrics [147] these can result in different sets of features, which may contain highly relevant features and also weakly relevant features. Thus, we hypothesise

that combining the features selected by multiple feature selection methods may yield better features that can improve the classification performance. Our objective is to use GP as an ensemble method to guide the feature selection process by combining top ranked features obtained by two well-known feature ranking metrics, IG and RF. GP is expected to produce a new and smaller set of features that can effectively improve the classification performance of the selected features. The two metrics are chosen due to their wide applications in the literature, their effectiveness in high dimensional data [150] and also their distinct characteristics. The two metrics were previously used in several bio-data mining applications, and achieved good results [57,77].

Figure 3.1 shows an overview of the proposed GP method. The proposed method has four steps: (1) we use the two feature ranking techniques (IG and RF) to rank the original features; (2) the top 100 ranked features by each of the two metrics are used as terminals of the GP method. According to the experiments, increasing the number of features from each metric more than 100 has resulted in increasing the search space, and therefore in decreasing the performance. The intrinsic capability of GP is used to search for good combinations of those features to form a (hopefully) better set of features; (3) the features selected by the best GP evolved program of each run are ranked according to their frequency of occurrence, and a new score is given to the features if it appears in more runs; (4)the 20 top ranked features are used for evaluation (classification with GP classifier and other classifiers). This parameter is selected according to the literature [10,133].

3.2.1 Overall Process

Since the datasets do not have a separate test set and also to overcome the limitation of the small number of training examples, ten-fold cross-validation is used where the data is divided into ten folds.

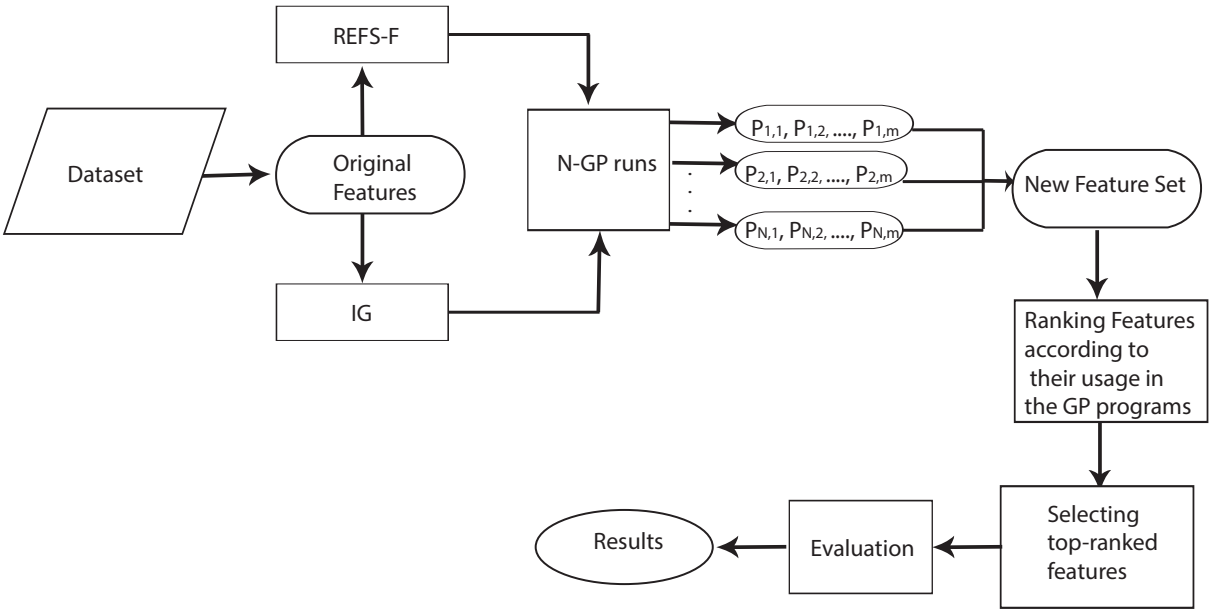


Figure 3.1: Overview of the GP-based approach.

At the beginning of each of the 10-folds cross-validation process, the random seed of GP is initialised, and the following steps are performed. Since the GP process is initialised at every fold, the bias of GP to the feature selection process is avoided. The process of 10-folds cross validation is explained as follows. For each fold, 30 independent GP runs are used, and hence, the total number of independent runs are 300.

1. Shuffle the data randomly;
2. Divide the data into ten-folds;
3. For every fold, do the following:
   (a) Use the current fold as a test and the rest of the folds as the training set
   (b) Run the GP Algorithm
   (c) Use the selected features to transform training and test sets.

(d) Calculate the accuracy of the test set

4. Calculate the average accuracy of 10 folds.

The rationale is as follows. Firstly, one set of the high-ranked features by one method and another set of high-ranked features by another method are used as input. Using these two sets of features can provide a mixture of some high-ranked and low-ranked features together that can potentially perform better than using all the individual features.

Secondly, the two metrics IG and RF use different criteria to rank individual features, and, we thus expect the combinations of the two groups of features using GP could potentially lead to better performance.

Thirdly, using GP to select features from the high-ranked features from IG and RF instead of all features can reduce the search space and also computational cost.

Fourthly, GP has an implicit feature selection capability, and we expect GP to select some individual features automatically from those chosen by the two metrics. The selected features combine together via the operators in the function set to form a small feature set, which can result in better classification performance.

Finally, we hypothesize that the more frequently occurring features must have the better impact on the classification. We therefore ranked each feature in the new feature set according to their frequency of occurrence in the GP programs, and used the top 20 features for evaluation (classification).

The aim of ranking the selected features is to examine the effect of the top ranked features selected by GP. This is done by comparing them with the top-ranked features by the individual feature metrics (IG and RF). For feature selection and ranking, the terminal set is composed of the top 100 features from each of the two metrics (IG and RF).

The terminal set has therefore 200 feature terminals, besides to randomly generated constant terminals. For the classification phase, the ter-

minal set is composed of the top 20 features selected by GP. In the rest of this section, we will describe the feature selection metrics, the terminal set, the function set, the fitness function, and parameter settings for the proposed method.

3.2.2 Feature Selection Metrics

The two feature selection metrics, Information Gain (IG) [150] and Relief-F (RF) [161] , are used to rank the importance of the individual features. We briefly describe them as follows.

IG determines the amount of information gained about a class when a certain feature exists or not [147]. It is defined as follows:

$$IG(\grave{x}, c_i) = \sum_{c \in \{c_1, c_2\}} \sum_{\grave{x} \in \{f, \bar{x}\}} P(\grave{x}, c) \frac{\log P(\grave{x}, c)}{P(\grave{x}) P(c)} \tag{3.2.2.1}$$

where x and $\bar{x}$ denotes the presence, and the absence of a feature. The two classes (healthy and diseased) are denoted by c_1 and c_2. The probability of the occurrence features $\grave{x}$ in a specific class is represented by $P(\grave{x}, C)$. $P(\grave{x})$ and $P(c)$ represent the probability of the selection of a feature and class, respectively.

RF searches for two nearest neighbours for a given example, one of the same class (hit) and the other of a different class (miss) [161] , and calculates the importance of the feature, which is given by:

$$\begin{aligned} I(X) \;=\; & \text{P(f} \mid \text{nearest instance} \in \text{different class)} \\ & -\text{P(f} \mid \text{nearest instance} \in \text{same class)} \end{aligned}$$

where P refers to probability, f denotes a value of a feature. A good feature should differentiate between instances belonging to different classes and should have similar values for the examples from the same class.

3.2.3 Fitness Measure

We aim at generating a subset of features that can improve the classification accuracy of the proposed GP approach, and at the same time reduces the number of features. We define the fitness function as the classification accuracy by restricting the fitness to select the minimum number of features. The classification accuracy is given by the following:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{3.2.3.1}$$

where TP is the number of true positives or the correctly classified disease examples. TN is the number of true negatives or the correctly classified healthy examples. FP and FN are the false positives and false negatives, respectively. The evaluation of the fitness measure is performed after filtering the selected subset of features by the evolved GP program.

We use the following fitness function:

$$fitness = A \times (1 + \sigma e^{-m/M}) \tag{3.2.3.2}$$

In Equation 3.2.3.2, A is the measure of accuracy obtained in Equation 3.2.3.1 and m is the number of features used (selected) in the GP program. M is the original number of features and σ is a parameter that is used to determine the relative importance between the classification accuracy and the number of features. σ is given by

$$\sigma = 2a(1 - \frac{CurrGen}{MaxGen}) \tag{3.2.3.3}$$

σ, in the early generations is more biased to decrease the number of features but with increasing generations, the classification accuracy is given more importance. a is a constant value of 0.1, $CurrGen$ is the current generation number and $MaxGen$ is the total number of generations.

When the value of m increases, the exponential factor $e^{-m/M}$ decreases and so does the fitness value. Therefore, if two programs (at same training point) have the same accuracy value, a higher fitness will be given to the program that has a smaller number of features.

A threshold value of 0 is used to classify the instances. For a specific instance of the training set, if the program output is ≤ 0, the instance is classified as c_1; otherwise as c_2.

3.2.4 Description of the Algorithm

Details about the algorithm are shown in Algorithm 1. The dataset is divided using 10-folds cross validation as explained earlier. For each training set, the algorithm starts by creating the initial population of individuals. The main loop of search in GP will end by either reaching the maximum number of generations ($MaxGen$) or when achieving the maximum fitness (f_{max}), 100%, which indicates that the problem is solved. The best program is determined by updating the best fitness variable (f_{max}). The fitness function aims at maximising the classification accuracy and minimising the number of features. The fitness of each program is determined by using the program output as a decision stump to determine the TP, TN, FP, FN at line 8. The constant factor a is given a value of 0.1 on line 10. At line 11, the parameter σ is calculated to control the importance of the classification accuracy and the number of features across the generations. At line 12, the fitness is obtained by measuring the classification accuracy of the program and multiplying it by a factor, which decreases the number of features. At lines 13-16, the $BestProgram$ with the best fitness in this generation is obtained, then at line 17, the selection and breeding of individuals are performed. The features used in the $BestProgram$ are sorted according to their frequency in the program, and the feature vector along with their ranks are returned as the result of this algorithm.

Algorithm 1 Selection and ranking of features through evolving classifiers

/* The algorithm return the best GP program with smaller number of features that can perform better in terms of classification accuracy */

Input D, a dataset of the form **D=(N,c)** where N is a set of instances of size N with 100 top features from IG and 100 top features from RF. c is the vector containing the class label of the instances.

Output (X,r, ACC), a vector with the features selected in the best GP individual $BestProgram$ and their rank according to their frequency of usage. The algorithm also returns the accuracy of GP as a classifier.

1. **for** k=1 to 10
2. Divide D in to 10 folds

   Take the current fold as a test set and the rest folds as a training set

   $P \Leftarrow$ create the initial population of individuals;

   $f_{max} \Leftarrow 0$; // Initialization of the maximum fitness

   while $CurrGen < MaxGen$ or $f_{max} < 1$ **do**
7. **foreach** $individual \in P$ **do**
8. $TP, TN, FP, FN \Leftarrow 0$;
9. Compute TN, TP, FP, FN of the training set; // According to the threshold value (zero)
10. $a \Leftarrow 0.1$
11. Calculate $\sigma = 2a(1 - \frac{CurrGen}{MaxGen})$
12. Evaluate fitness $f = \frac{TP+TN}{TP+TN+FP+FN} \times (1 + \sigma e^{-m/M})$;
13. **if** $f > f_{max}$ **then**
14. $f_{max} \Leftarrow f$;
15. $BestProgram \Leftarrow individual$;
16. **end if**
17. Perform selection and breeding; // Perform selection and genetic operators
18. $CurrGen \Leftarrow CurrGen$+1;

   end while

   Use $BestProgram$ to evaluate the test set

   Calculate the accuracy of the test set

   Compute the frequency of occurrence $\forall$ X $\in BestProgram$; // Calculate the number of usage of each feature in $BestProgram$

   $r[i] \Leftarrow$ indexed descending sorted;

   end for
25. Compute the average of all the folds as a result of GP classification accuracy ACC
26. return (X,r,ACC);

3.3 Experiments Design and Setup

3.3.1 GP Settings

The MS data is represented by $(m/z, I) = (m/z, I_1, ..., I_n)$, where m/z is a vector of the measured m/z ratios and I_i is the corresponding intensity of the i^{th} sample. The LC-MS data is represented by $(time, m/z, I) = (time, m/z, I_1, ..., I_n)$, where time is the retention time vector of the production time of a protein or a peptide. The m/z and time values do not change across all samples as they correspond to the identity of the samples, thus we can consider the m/z and time values as the features' identities. The feature values are the intensity profile. The objective here is to predict the class label based on this intensity profile [182]. The thirteen datasets used have two classes and the class labels can be defined as class 1 or class 2, respectively.

Terminal Set. The goal of GP, as stated earlier, is to further select a smaller number of features from the feature pool selected by IG and RF, and to rank these features to improve the overall classification performance.

Function Set. The four common mathematical operators $+, -, \times$ and $\%$ were used in addition to the square root $\sqrt{}$ and $\max$ functions. The division operator ($\%$) is protected where it returns 0 for the division by 0. The $\sqrt{}$ is also a protected operator, in which, if the argument is a negative value, the absolute of this value is taken. The goal of using $\sqrt{}$ and $\max$ functions is to evolve non-linear and complex functions that can perform well for classification and feature selection.

GP Parameters. For the GP system, the tree-based GP [96] is used where each program produces a single floating-point number at its root as a result of its evaluation (output). The standard subtree crossover and mutation [145] are used with a probability of 80% and 15%, respectively. The initial population is generated using the ramped half-and-half method [145]. The individual program tree depth is minimum 5 and can be increased to 8 during the evolution. The population size is 1024. The selec-

Table 3.1: GP settings

Function set	$+, -, \times, \%, \sqrt{}$ and $\max$
Variable terminals	200 features (top ranked 100 features from IG and RF metrics)
Constant terminals	Randomly generated constants
Initialization method	Ramped Half-and-Half
Initial tree Depth	5
Maximum tree depth	8
Generations	30
Mutation probability	15%
Crossover Rate	80%
Elitism	5%
Population Size	1024
Selection type	Tournament
Tournament Size	10

tion method used is the tournament selection with a tournament size of 10. Elitism is taken here with 5% probability to make sure that the best individual in the next generation is not worse than that in the current generation. The process of evolution will be terminated when the maximum number of generations (30 generations) is reached. Table 3.1 summarises the GP settings of our method. These parameters are estimated according to the literature [92].

3.4 Datasets and Preprocessing

3.4.1 Datasets

Four MS datasets and nine LC-MS datasets were used in our experiments. In the MS datasets, the samples include patients with cancer and healthy individuals. The MS datasets include the following:

- OVA1 dataset [133]: this dataset is composed of 121 cancerous and 95 healthy samples. This dataset was generated from a hybrid quadrupole

time-of-flight (TOF) mass spectrometer (ABI Qstar) fitted with a ProteinChip1 array interface. The samples were analysed by surface-enhanced laser deisotopic SELDI, and the m/z values range from 500 to 2000 Da.

- OVA2 dataset [133]: consists of spectra from 162 patients with ovarian cancer and 91 healthy individuals. To produce this dataset serum samples were analysed on two SELDI-TOF mass spectrometers and the molecular masses range from 0 to 20,000 Da. Mass resolution is routinely achieved below 400 Da.

- PAN dataset [49]: consists of a diseased group of 80 individuals and a healthy group of 101 individuals. The samples were subjected to SELDI-TOF MS on a Protein Biology System 2c. The m/z values range from 800.00 to 11,992.91 Da.

- ARC dataset [64]: is composed of 100 cancerous samples (ovarian or prostate cancer) and 100 healthy samples. The dataset results from merging datasets from three different sources (ovarian cancer samples of two different types and prostate cancer samples). The data was obtained with the SELDI technique and the m/z values range from 200 to 10,000 Da.

The LC-MS datasets are generated from the serum samples with known concentration of spike-in peptides which are the human defined biomarkers (class 1) or without spike-in peptides (class 2). The characteristics of the LC-MS datasets are described as follows.

- Spike-in serum dataset: This dataset is obtained from Georgetown University[1]. It contains thirteen peptide biomarkers spiked in the first class samples while the second class includes only serum samples. Each class contains five samples.

[1]Available at: `http://omics.georgetown.edu/massprep.html`

These two groups of data were generated from five serum samples obtained from five healthy individuals, and the two groups were acquired by the same LC-MS method. The spiked-in MassPrep peptide mixture is a selection of nine peptides with a wide range of polarities, and isoelectric points. Chromatographic separation was performed on Waters NanoACQUITY system using BEH C18 column (Tuli et al., 2012). The MS scan cycles of seven seconds included m/z values ranging from 350 to 2000 Da and five MS/MS scans, where the m/z values that range from 50 to 2000 Da. More details about the sample collection, preparation, and storage can be found in [169]. This dataset is denoted as DS_a.

- Spiked porcine CSF dataset: The number of samples in this dataset is ten, five samples in each class, where class 1 samples are non-spiked, and class 2 samples are high spiked. This dataset is characterised by high between-class variability and low within-class variability. The number of features in each sample is 9889 with 38 added spiked features defined as the biomarkers. More details regarding the preparation and acquisition of this dataset are described in [73]. The samples with spiked peptides were prepared by mixing 20 μ L CSF digest with 20 μ L of a tryptic digest of horse heart cytochrome C. Spiked and non-spiked samples were injected five times in a random order into an Agilent QTOF 6. This dataset is denoted as DS_b.

- Spiked human urine datasets: Seven datasets were obtained from 40 chromatograms, each with two classes that have high or low levels of spiking levels of peptides. The number of biomarkers is 151, and the total number of features in each sample is 29529. The datasets have different between-class and within-class variabilities and different samples sizes. Fifty urine samples were obtained and 200 μ L were taken from each sample and spiked with a tryptic digest (Promega, Madison, WI, V5111) of bovine carbonic anhydrase (Sigma, Stein-

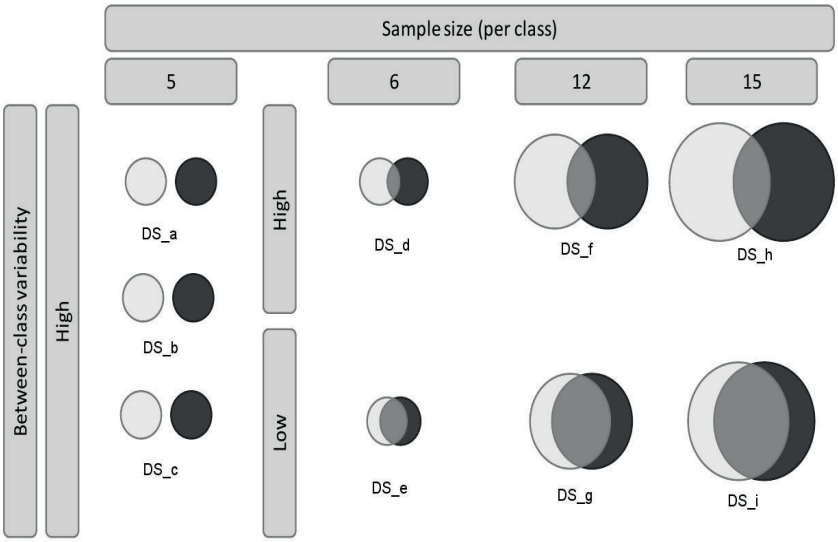


Figure 3.2: Properties of the LC-MS datasets used in the experiments.

heim, Germany, C3934, Uniprot entry: P00921) as well as with seven synthetic peptides. The sample was analysed five times using an Agilent G2445A LC/MSD-Trap-SL ion trap mass spectrometer. We use the following notation for each of the seven datasets: DS_c, DS_d, DS_e, DS_f, DS_g, DS_h, DS_i throughout the chapter to denote the seven human urine datasets. Figure 3.2 depicts the properties of the LC-MS datasets used in the experiments. The datasets DS_b, DS_c, DS_d, DS_e, DS_f, DS_g, DS_h, DS_i were all obtained from Netherlands Bioinformatics Center[2].

3.4.2 Data Preprocessing

Preprocessing of the MS data involves a sequence of operations. These operations represent essential steps for analysing the data successfully. These steps include:

[2]Available at: `https://trac.nbic.nl/BiomarkerFeatureSelection`

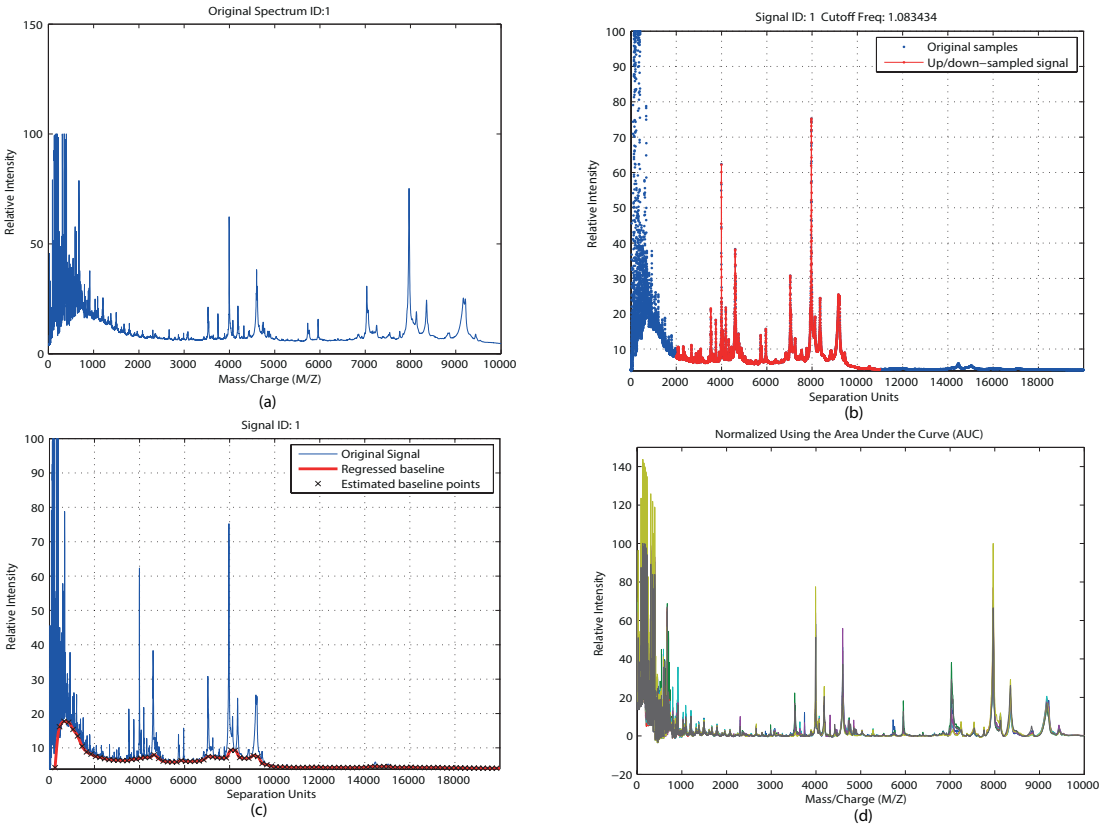


Figure 3.3: Preprocessing steps of the low-resolution ovarian cancer dataset. (a) the original spectrum. (b) the baseline adjustment of the first signal. (c) resampling of this signal. (d) normalisation of the samples using AUC.

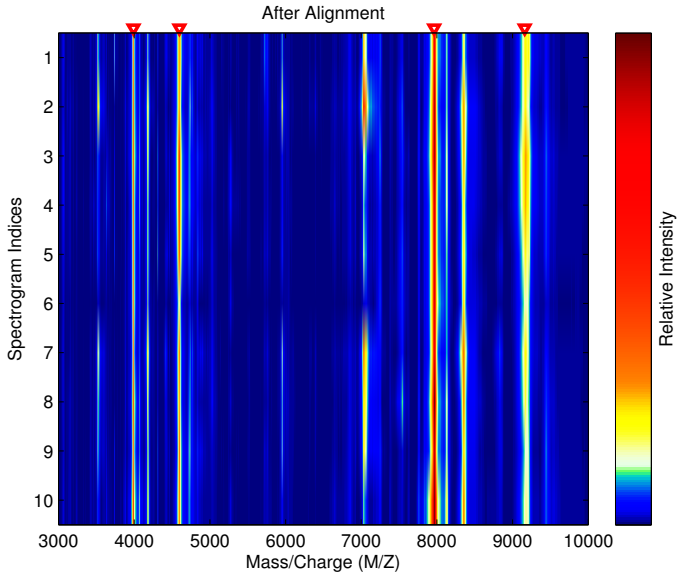


Figure 3.4: An example of the alignment of 10 spectra of the low-resolution ovarian cancer dataset.

1. Baseline correction and signal filtering: remove noise and baseline artifacts.

2. Peak picking and extraction: find and extract the real peaks corresponding to molecules and remove the peaks that result from instrumental errors.

3. Multiple map alignment: correct the distortion of the retention time and m/z dimension of multiple raw or feature maps.

4. Intensity normalisation: normalise the spectral counts to remove the fluctuation in the intensity values across the different spectra.

The LC-MS data is a time series of the MS spectra. The preprocessing of the non-chromatographic MS data can share some common steps of preprocessing with the LC-MS datasets although the preprocessing framework of the non-chromatographic MS is not exactly the same as the LC-MS

data. For example, the steps of the baseline adjustment, filtering and normalisation are the same for both MS and LC-MS datasets. However, the alignments of MS and LC-MS data are different. The alignment of MS data is performed on the m/z values while the alignment of LC-MS data is done on both the retention time and m/z values.

OVA1 and OVA2 datasets: During the MS analysis, the number of features produced in all the samples may not be the same. Therefore, the first step is to make the number of features equal for all samples to obtain the same m/z point at all MS spectra [133]. This is done by using the resampling algorithm in the toolbox. The background and chemical noise are removed by the baseline adjustment step. The noise is usually higher at the low-intensity peaks. To estimate the baseline, a window of size 50 m/z for the high-resolution data is passed across the spectra and the minimum values of the m/z ratios are calculated. For the low-resolution data, the window size is set to 500 m/z points. Afterwards, the baseline is regressed and subtracted [133]. The third step is to remove the fluctuation in the m/z values, which occurs due to the miscalibration of the machine. The alignment of the m/z values is done by shifting and scaling the m/z axis until the maximum alignment of intensity values is reached. The final step is to remove the variation among the intensity values, which occurs due to the changing of the levels of compounds or sometimes the sensitivity of the detector part in the machine. This is performed by normalising each spectrum using the area under the curve (AUC). As an example, Figure 3.3 shows the original spectrum and the spectrum after three steps of preprocessing of the low-resolution ovarian cancer dataset. An example of the result of the alignment of 10 spectra is shown in Figure 3.4.
PAN dataset: The first step done for this dataset is baseline correction. The baseline is estimated by segmenting the whole spectra into windows with a size of 200 m/z ratio intensities. Afterwards, the means of the intensity values under the windows are used as the baseline, and a regression of the baseline is performed using a piecewise cubic interpolation method [49].

The next step is to filter the noise. This is done using a Gaussian kernel filter. The last step is the normalisation of the spectra using the AUC method.

ARC dataset: The dataset is preprocessed by the providers by removing the fluctuation or the technical repeats by averaging them, then removing the baseline. Afterwards, smoothing the signals and alignment take place.
DS_a dataset: This dataset is an LC-MS dataset, where each sample con-

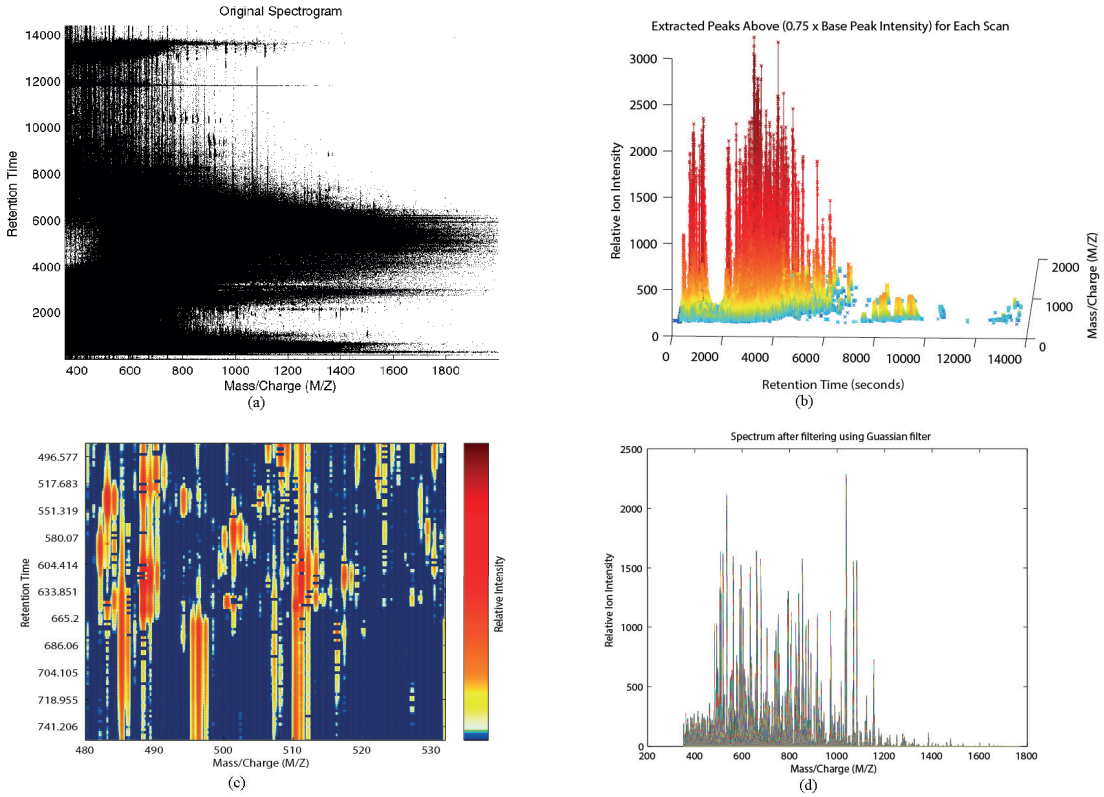


Figure 3.5: Preprocessing of the raw data spectrum of DS_a dataset. (a) The original raw spectrum. (b) Peak extraction step. (c) Alignment step (d) Filtering step.

sists of retention time, m/z ratios and their corresponding intensity [103]. Ideally, the same compounds detected by the same LC-MS should have the same abundances, m/z ratios and retention times, but this is not usu-

ally the case due to the experimental variation. Thus, the preprocessing of this dataset is different from that of the non-chromatographic MS datasets. The first step is the peak extraction, which selects the m/z features with reasonable intensities and signal to noise ratio. This step is done by clustering significant peaks and noisy peaks and removing the noisy peaks using the toolkit. Figure 3.5 shows the raw data form and the data after peak extraction, alignment and smoothing.

The alignment of the peaks is used to remove fluctuation or the small variation of the data. Finally, smoothing of peaks is done in order to remove noise with each scan. This smoothing is performed using a percentile of the base peak intensity, where the base peak is the most intense peaks found in each scan. The peak preserving resampling method is adopted to produce the centroid data. To perform a two-dimensional analysis, the total time ion count (TTIC) for each m/z value is calculated. TTIC is the sum of all intensity values of each m/z value at a specific retention time value. After preprocessing, the number of features becomes 800 for this dataset. The rest of the LC-MS datasets are available after preprocessing.can be analysed using the same computational framework. This is because the feature vector for both of them is the intensity vector and both the m/z and the retention values are the feature identities.

3.5 Results and Discussions

To evaluate the performance of our proposed method for feature selection and classification, we conducted a number of experiments on the four MS datasets and the nine LC-MS datasets. The classification performance of all the available features is used as a baseline, which is compared with that of the 20 top ranked features resulting from the proposed GP (feature selection and ranking) system, the 100 top ranked features from the IG method and the 100 top ranked features from the RF method.

The GP classifier was first used to evaluate the classification performance of the top ranked features. The detailed results are shown in Table 3.2. To test the classification performance of the top ranked features on other classifiers, NB, J48 decision tree, random forest and SVMs classifiers were also used in the experiments, where the results are shown in Table 3.3.

3.5.1 GP Feature Selection and Classification Performance

Table 3.2 shows the classification performance of the GP classifier using top ranked features resulting from the proposed GP system, the IG method and the RF method. In Table 3.2,

- *"ORG-GP"* means the GP classifier using all the original features for classification.
- *"IG-GP"* means the GP classifier using the 100 top ranked features obtained by the IG method.
- *"RF-GP"* means the GP classifier using the 100 top ranked features obtained by the RF method.
- *"IGRF-GP"* means the GP classifier using the 20 top ranked features obtained by the proposed GP feature selection and ranking approach.

Table 3.2 shows the classification accuracy over 30 independent 10-fold cross-validations runs (30×10=300 runs) for each dataset. The total number of runs performed are 15600 runs (300×13×4) for all thirteen datasets and all four methods. As shown in Table 3.2, the proposed GP approach with only the 20 top ranked features outperforms ORG-GP, IG-GP and RF-GP on almost all of the datasets.

This confirms our previous hypothesis that GP can produce a smaller set of features that can decrease the dimensionality and at the same time improve the classification performance. This is mainly because the GP

Table 3.2: Experimental Results

Dataset	Method	#Features	Ten folds cross validation (%)		
			Best Acc	Average Acc ±St.dev	
OVA2	ORG-GP	15000	92.80	86.23±3.8	
	IG-GP	100	96.80	94.64±1.21	
	RF-GP	100	96.80	93.71±1.73	
	IGRF-GP	20	98.40	96.07±1.49	§ * ‡
OVA1	ORG-GP	15154	87.14	81.44±3.58	
	IG-GP	100	93.34	88.14±2.13	
	RF-GP	100	75.24	68.47±2.97	
	IGRF-GP	20	93.34	89.62±1.89	§ * ‡
PAN	ORG-GP	6771	56.11	44.96 ±5.28	
	IG-GP	100	61.11	51.82±4.83	
	RF-GP	100	69.45	60.17±4.53	
	IGRF-GP	20	70.00	62.35±4.16	§ * ‡
ARC	ORG-GP	10000	73.00	67.87±2.81	
	IG-GP	100	77.00	72.25±2.27	
	RF-GP	100	79.5	75.48±2.31	
	IGRF-GP	20	82.00	76.38±2.7	§ *
DS_a	ORG-GP	800	80.00	44.00±15.89	
	IG-GP	100	80.00	55.67±13.57	
	RF-GP	100	80.00	55.67±13.57	
	IGRF-GP	20	100.00	71.33±14.56	§ * ‡
DS_b	ORG-GP	9889	100.00	88.00±13.49	
	IG-GP	100	100.00	91.33±9.37	
	RF-GP	100	100.00	98.33±3.79	
	IGRF-GP	20	100.00	99.33±2.54	§ * ‡
DS_c	ORG-GP	29529	100.00	43.67±30.68	
	IG-GP	100	100.00	83.67±14.67	
	RF-GP	100	100.00	94.00±9.32	
	IGRF-GP	20	100.00	94.00±9.32	§ *
DS_d	ORG-GP	29529	100.00	48.67±20.63	
	IG-GP	100	100.00	92.33±9.35	
	RF-GP	100	100.00	86.3±14.26	
	IGRF-GP	20	100.00	93.60±8.77	§ ‡
DS_e	ORG-GP	29529	100.00	56.67±17.24	
	IG-GP	100	100.00	90.67±9.44	
	RF-GP	100	100.00	86.50±10.76	
	IGRF-GP	20	100.00	95.17±5.49	§ * ‡
DS_f	ORG-GP	29529	100.00	54.78±16.53	
	IG-GP	100	100.00	92.89±7.67	
	RF-GP	100	100.00	92.33±7.54	
	IGRF-GP	20	100.00	94.11±6.35	§
DS_g	ORG-GP	29529	80.00	43.67±17.32	
	IG-GP	100	100.00	93.00±12.64	
	RF-GP	100	100.00	72.33±19.42	
	IGRF-GP	20	100.00	87.67±11.67	§ ‡
DS_h	ORG-GP	29529	80.00	54.00±15.26	
	IG-GP	100	100.00	82.50±15.41	
	RF-GP	100	95.00	76.50±13.66	
	IGRF-GP	20	100.00	90.33±8.6	§ * ‡
DS_i	ORG-GP	29529	80.00	49.11±14.62	
	IG-GP	100	100.00	82.11±10.26	
	RF-GP	100	100.00	84.67±9.91	
	IGRF-GP	20	100.00	89.11±6.19	§ * ‡

classifier has the capability to form high-Level features from the low-level features through its operators. The high-level features are the combinations of the low-level features and the functions of the function set. Therefore, these combinations help in discovering the hidden relationships between the low-level features and hence improve classification performance. Another possible reason could be that the combination of some high ranked features with some low ranked features can improve the classification performance. Moreover, IG and RF can select relevant features and some other less relevant features, thus GP as a search technique can select features from both metrics and form a better set of features. Finally, the frequency of a specific feature in the GP program indicates its importance and usefulness in classification. Therefore, making a new ranking scheme according to this hypothesis for the features can improve the classification performance.

Making a new ranking scheme benefits in many ways. Firstly, using 20 features instead of 200 features in the terminal set substantially decreases the dimensionality and search space. Secondly, using these 20 features improves the classification accuracy. It was noticed that the variance between the different runs, i.e. smaller standard deviation, is considerably lower in the case of the proposed approach in most of the datasets. This suggests that the proposed solution can be considered stable. Finally, in terms of biomarker detection, the proposed method can decrease the cost of experimental validation by reducing the number of features to be tested. In all the thirteen datasets, the baseline performance is worse than using the feature ranking approaches. The 100 top ranked features obtained by RF outperforms that of IG in five of the thirteen datasets.

To confirm the statistical significance of the results shown in Table 3.2, a T-test with 90% confidence level is performed between IGRF-GP and ORG-GP, IG-GP, or RF-GP. The results are presented as small symbols in Table 3.2:

- " § " means "IGRF-GP" is significantly better than ORG-GP.

- " * " means "IGRF-GP" is significantly better than IG-GP with 100 features.
- " ‡ " means "IGRF-GP" is significantly better than RF-GP with 100 features.

The proposed method (IGRF-GP) is shown to be significantly better than ORG-GP for all the datasets. In ten datasets, IGRF-GP is significantly better than using IG-GP (100 features) and in eight datasets, IGRF-GP shows significant better performance than RF-GP (100 features). In DS_g, the result of the T-test (p-value=0.0951) shows that the difference between "IGRF-GP" and IG-GP (100 features) is not really significant, which means that IG-GP is similar to IGRF-GP. Overall, it is clear that IGRF-GP achieved significantly better or similar performance compared to the other three methods on these datasets but with a smaller number of features.

The results suggest that GP can be used successfully as a feature selection method and also as a classifier, and that the early hypothesis of using GP to mix the advantages of multiple feature selection metrics is confirmed.

3.5.2 Using GP Features With Other Classifiers

For comparison, the original features, the 100 top ranked features of IG, the 100 top ranked features of RF, and the 20 top ranked features of the proposed GP system, were used in NB, J48, random forest and SVMs classifiers for classification. The results are shown in Table 3.3.

As shown in Table 3.3, the 20 top ranked features of the GP system either outperform or have a similar performance to the original features when used with the NB, J48 and random forest classifiers.

Using NB classifiers, the 20 GP features outperforms the 100 IG features in five datasets and have similar performance in the rest datasets.

This suggests that GP can shrink the search space and decrease the number of features by removing the redundant features and can increase

Table 3.3: Classification performance for the tasks using NB, J48, Random Forest and SVMs classifiers

Dataset	Method	#Features	NB	J48	Random Forest	SVMs
OVA2	Org.	15154	76.28	95.65	93.28	100.00
	IG	100	93.67	95.65	98.02	99.20
	RF	100	91.69	97.23	97.62	97.62
	GP	20	94.46	97.23	98.42	97.23
OVA1	Org.	15000	83.79	86.57	87.03	96.29
	IG	100	88.42	87.96	88.88	93.52
	RF	100	93.67	95.65	90.74	93.98
	GP	20	90.27	88.88	91.67	93.06
PAN	Org.	6771	51.38	50.82	58.56	62.43
	IG	100	57.45	63.53	64.64	66.29
	RF	100	63.53	65.74	65.19	62.42
	GP	20	65.74	64.08	74.03	71.27
ARC	Org.	10000	70.00	81.00	72.5	90.00
	IG	100	67.50	75.00	81.00	78.00
	RF	100	75.50	82.00	82.00	78.50
	GP	20	73.5	80.00	80.00	73.00
DS_a	Org.	800	70.00	90.00	40.00	70.00
	IG	100	80.00	90.00	90.00	70.00
	RF	100	80.00	90.00	90.00	70.00
	GP	20	80.00	90.00	90.00	90.00
DS_b	Org.	9889	80.00	90.00	70.00	50.00
	IG	100	100.00	90.00	100.00	100.00
	RF	100	100.00	90.00	100.00	100.00
	GP	20	100.00	90.00	90.00	100.00
DS_c	Org.	29529	90.00	90.00	90.00	100.00
	IG	100	100.00	100.00	100.00	100.00
	RF	100	90.00	70.00	100.00	100.00
	GP	20	100.00	100.00	100.00	100.00
DS_d	Org.	29529	66.67	58.33	41.67	58.33
	IG	100	100.00	100.00	100.00	100.00
	RF	100	100.00	100.00	100.00	100.00
	GP	20	100.00	91.67	100.00	100.00
DS_e	Org.	29529	75.00	91.67	86.67	83.33
	IG	100	100.00	91.67	100.00	100.00
	RF	100	95.83	91.67	100.00	100.00
	GP	20	100.00	91.67	100.00	100.00
DS_f	Org.	29529	73.33	90.00	86.67	83.33
	IG	100	96.67	96.67	100.00	100.00
	RF	100	100.00	93.33	100.00	100.00
	GP	20	100.00	96.67	100.00	100.00
DS_g	Org.	29529	50.00	41.67	41.67	58.33
	IG	100	100.00	91.67	100.00	100.00
	RF	100	100.00	100.00	100.00	100.00
	GP	20	100.00	100.00	100.00	100.00
DS_h	Org.	29529	83.33	79.16	77.27	86.36
	IG	100	100.00	79.16	100.00	100.00
	RF	100	95.83	87.5	100.00	100.00
	GP	20	95.83	91.67	100.00	91.67
DS_i	Org.	29529	70.00	90.00	83.33	76.67
	IG	100	100.00	90.00	100.00	96.67
	RF	100	96.67	90.00	100.00	96.67
	GP	20	100.00	90.00	100.00	100.00

or maintain the same performance. The only exception is DS_h where IG performs better than GP. The 20 GP features outperform the 100 RF features in three datasets and their performance are similar in nine datasets.

The classification performance of SVMs using all the original features is better than that of using the top ranked features of the GP system, IG and RF in three datasets. However, SVMs using the 20 GP features achieve better performance in the PAN datasets and DS_a. In addition, SVMs with the 20 GP features achieve the ideal classification performance in eight LC-MS datasets. Comparing Table 3.2 with Table 3.3, it can also be noticed that the best classification performance of the IGRF-GP is better than NB, J48 and random forest in most of the datasets. The performance of SVMs is very competitive with GP classifier because both SVMs and GP are good for binary classification. The good performance of GP in binary classification is due to the splitting of the program output to positive and negative for classifying the two classes. This is also due to the ability of GP to form high-level features from the low-level features.

3.5.3 Biomarker Detection

The LC-MS datasets contain a number of artificially spiked peptides, which should be detected as the biomarkers that differentiate one class from the other class. The number of peptides defined as the real biomarkers is 13 in DS_a (9 peptides with different charges). In DS_b, there are 38 defined biomarkers, while in rest of the datasets, the number of biomarkers defined is 151. We tracked the features selected by the proposed GP method to test the number of biomarkers detected. Figure 3.6 shows the number of biomarkers detected and the biomarker detection rate obtained by the proposed GP method, IG and RF.

As shown in Figure 3.6, the number of biomarkers detected by GP is larger than that of IG and RF in all the datasets. In DS_b, the biomarker detection rate of GP is 100%. The performance of GP for biomarker detection

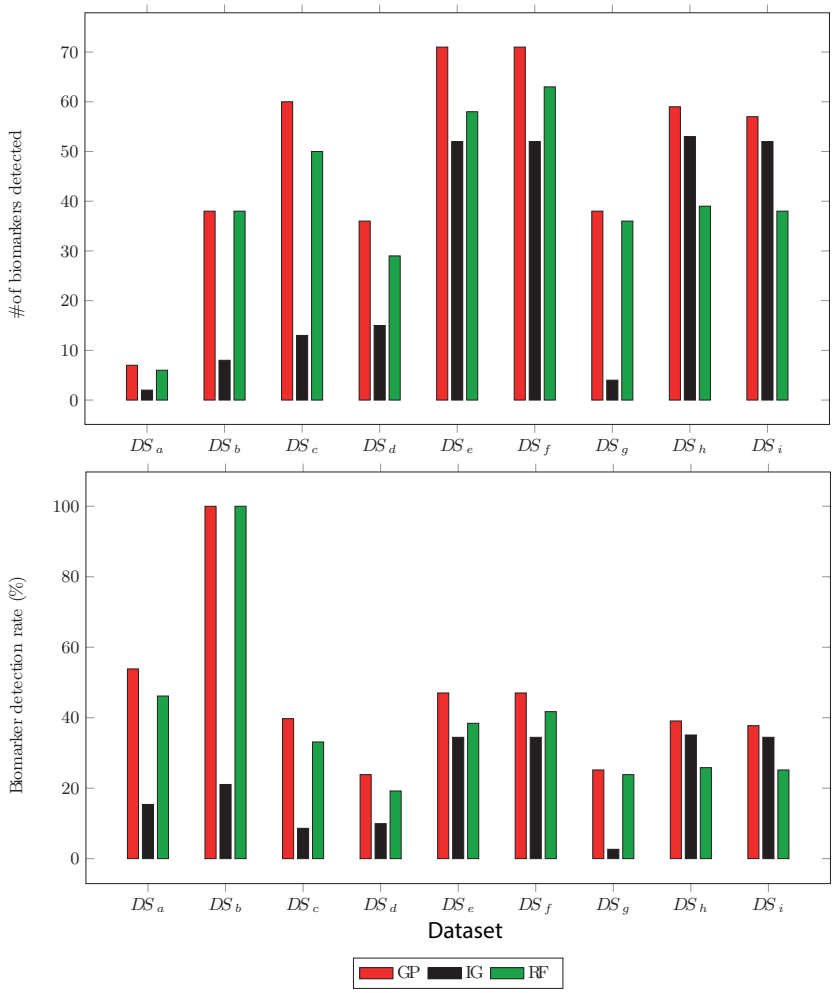


Figure 3.6: Biomarker detection of the proposed method in comparison with IG and RF.

tends to be stable regardless of the sample size, the high within-class variability or the low between-class variability. This can be a reason for GP to be advantageous for these types of tasks since only a very small number of samples are available from the mass spectrometer. The performance of IG is affected by the sample size to a larger extent. It can be observed that the performance of IG increases in the datasets DS_e, DS_f, DS_h and DS_i, where the number of samples is 24, 30, 24, and 30, respectively. In DS_a, DS_b and DS_c, where the between-class variability is high and the within-class variability is low, the performance of IG degrades although the biomarker detection task here is simpler. The performance of RF is more stable than IG across the datasets. In all cases, the proposed GP method merged the advantages of both metrics and detected more biomarkers than each metric individually.

3.6 Further Discussions

3.6.1 The 20 top ranked features from the proposed GP method, IG and RF using GP classifier

Another comparison with the top 20 features ranked by the proposed GP method, and the top 20 features ranked by both IG and RF in terminal set of a GP classifier is performed here. The results are shown in Table 3.4. The significance test is also performed and the results are shown with the following marks in Table 3.4.

- " *" means that IGRF-GP is significantly better than IG-GP with 20 features.
- "$\wedge$" means that IGRF-GP is significantly better than RF-GP with 20 features.

As can be seen in Table 3.4, the average classification accuracy of IG-GP using only the 20 features is worse than using the 100 features (as shown

in Table 3.2) in seven datasets. Compared with using the 100 top ranked features, the average classification accuracy of RF-GP, using only the 20 features, decreases in four datasets. Although in some cases, using 20 features from IG or RF improves the average accuracy over using the 100 features, IGRF-GP is still better than both IG-GP and RF-GP in almost all cases. Specifically, the average classification accuracy of IGRF-GP is better than IG-GP in all the cases and better than RF-GP in ten of the thirteen datasets (nearly the same in two datasets). The significance tests show that with 20 features, IGRF-GP is significantly better than IG-GP (20 features) in six datasets and significantly better than RF-GP (20 features) in eight datasets. According to the T-test results, IGRF is either significantly better or similar to IG-GP (20 features) and RF-GP (20 features) and it is never significantly worse than either of them.

3.6.2 The 20 top ranked features from the proposed GP method, IG and RF, using other classifiers

The 20 top ranked features from the proposed GP method, IG and RF are also used in NB, J48, random forest, and SVMs to test their classification performance of these classifiers. The results are shown in Table 3.5.

Comparing the GP method with IG, when using NB as the classifier, the classification performance of using the 20 features from GP is better than using the 20 features from IG in six datasets. They are the same in another six datasets, where both of them achieved the perfect performance (100%) in five of these six cases. The performance of J48, random forest and SVMs also show a similar pattern. In almost all cases, the classification performance of the 20 features from GP is better or at least the same as that of the 20 features from IG.

Comparing the GP method with RF, for all the four classifiers, the classification performance of the 20 features from GP is better or the same as that of the 20 features from IG in almost all cases. In most of the DS

Table 3.4: The performance of top 20 features selected by the GP method compared to the top 20 features by IG and RF with GP classifier.

			Ten folds cross validation (%)		
Dataset	Method	#Features	Best Acc	Average Acc ±St.dev	
OVA2	IG-GP	20	96.80	93.15±1.86	
	RF-GP	20	98.00	96.03±1.73	
	IGRF-GP	20	98.40	96.07±1.49	$\star$
OVA1	IG-GP	20	86.19	82.87±1.74	
	RF-GP	20	65.24	61.30±2.56	
	IGRF-GP	20	93.34	89.62±1.89	$\star \wedge$
PAN	IG-GP	20	61.67	51.59±5.26	
	RF-GP	20	46.67	42.17±2.39	
	IGRF-GP	20	70.00	62.35±4.16	$\star \wedge$
ARC	IG-GP	20	78.00	71.22±2.77	
	RF-GP	20	77.50	71.47±3.21	
	IGRF-GP	20	82.00	76.38±2.7	$\star \wedge$
DS_a	IG-GP	20	100.00	64.00±15.22	
	RF-GP	20	100.00	61.00±15.83	
	IGRF-GP	20	100.00	71.33±14.56	$\wedge$
DS_b	IG-GP	20	100.00	62.67±12.85	
	RF-GP	20	100.00	97.33±4.5	
	IGRF-GP	20	100.00	99.33±2.54	$\star \wedge$
DS_c	IG-GP	20	100.00	88.33±12.06	
	RF-GP	20	100.00	94.67±5.04	
	IGRF-GP	20	100.00	94.00±9.32	$\star$
DS_d	IG-GP	20	100.00	88.67±11.06	
	RF-GP	20	100.00	89.00±5.35	
	IGRF-GP	20	100.00	93.60±8.77	$\wedge$
DS_e	IG-GP	20	100.00	95.33±6.29	
	RF-GP	20	100.00	91.17±3.13	
	IGRF-GP	20	100.00	95.17±5.49	$\wedge$
DS_f	IG-GP	20	100.00	93.89±4.64	
	RF-GP	20	100.00	96.89±3.27	
	IGRF-GP	20	100.00	94.11±6.35	
DS_g	IG-GP	20	100.00	82.33±14.06	
	RF-GP	20	100.00	85.17±9.8	
	IGRF-GP	20	100.00	87.67±11.67	
DS_h	IG-GP	20	100.00	88.83±8.38	
	RF-GP	20	100.00	89.17±5.88	
	IGRF-GP	20	100.00	90.33±8.6	
DS_i	IG-GP	20	96.67	89.11±5.87	
	RF-GP	20	100.00	86.22±6.53	
	IGRF-GP	20	100.00	89.11±6.19	$\wedge$

Table 3.5: Classification performance of 20 features using NB and J48, random forest, and SVMs classifiers.

Dataset	Method	#Features	NB	J48	Random Forest	SVMs
OVA2	IG	20	89.72	93.67	96.83	97.23
	RF	20	96.44	96.83	96.05	96.44
	GP	20	94.46	97.23	98.42	97.23
OVA1	IG	20	85.18	86.11	86.57	89.35
	RF	20	88.89	86.57	89.35	88.89
	GP	20	90.27	88.88	91.67	93.06
PAN	IG	20	65.19	66.85	65.19	62.43
	RF	20	60.22	62.98	60.22	60.77
	GP	20	65.74	64.08	74.03	71.27
ARC	IG	20	69.00	73.50	80.00	65.50
	RF	20	67.50	71.00	73.50	71.00
	GP	20	73.5	80.00	80.00	73.00
DS_a	IG	20	80.00	90.00	90.00	80.00
	RF	20	80.00	90.00	90.00	90.00
	GP	20	80.00	90.00	90.00	90.00
DS_b	IG	20	60.00	80.00	90.00	100.00
	RF	20	100.00	100.00	100.00	100.00
	GP	20	100.00	90.00	90.00	100.00
DS_c	IG	20	100.00	90.00	100.00	100.00
	RF	20	100.00	90.00	100.00	100.00
	GA	45	40.00	60.00	40.00	60.00
	GP	20	100.00	100.00	100.00	100.00
DS_d	IG	20	100.00	100.00	100.00	100.00
	RF	20	100.00	100.00	100.00	100.00
	GP	20	100.00	91.67	100.00	100.00
DS_e	IG	20	100.00	100.00	100.00	100.00
	RF	20	100.00	100.00	100.00	100.00
	GA	381	54.16	87.50	83.33	66.67
	GP	20	100.00	91.67	100.00	100.00
DS_f	IG	20	100.00	90.00	100.00	100.00
	RF	20	100.00	90.00	100.00	100.00
	GP	20	100.00	96.67	100.00	100.00
DS_g	IG	20	91.67	100.00	100.00	100.00
	RF	20	100.00	91.67	91.67	100.00
	GP	20	100.00	100.00	100.00	100.00
DS_h	IG	20	100.00	95.45	100.00	100.00
	RF	20	100.00	86.36	100.00	100.00
	GP	20	95.83	91.67	100.00	91.67
DS_i	IG	20	100.00	96.67	100.00	96.67
	RF	20	100.00	90.00	93.33	96.67
	GP	20	100.00	90.00	100.00	100.00

datasets, their classification accuracies are the same, which is usually the perfect performance (100%).

3.6.3 The proposed GP method compared to GA

The top-ranked features from the proposed GP method are also compared with the features selected by genetic algorithms (GA) wrapped with SVMs for feature selection. The weka [66] package was used to run the GA feature selection method and the settings of GA are as follows:

to avoid the premature convergence, the population size is set 300 and the number of generations is set to 1000.

This makes the same number of evaluations. The crossover and mutation probabilities are to 0.8 and 0.033, respectively. The classification performance is evaluated by NB, J48, random forest, and SVMs, which can be seen in Table 3.6.

According to Table 3.6, it can be seen that the number of features selected by GA is significantly larger than 20 in the GP method. However, using only the 20 features from the GP method, NB, J48 and random forest can achieve better classification performance than using the much larger feature sets from GA in all the 13 datasets. For example, in the ARC dataset, GA selected 324 features, which is over 16 times larger than the 20 features from GP. However, the classification performance of NB, J48 and random forest with the 20 features ranked by GP are better than with the 324 features selected by GA.

SVMs were wrapped in the feature selection process of the GA method to evaluate the classification performance of the selected features. Therefore, the features selected by GA are expected to achieve good performance with SVMs. However, only in three of the 13 datasets, SVMs with the features selected by GA achieved better classification performance than SVMs with the 20 features selected by the proposed GP method. On the other 10 datasets, the classification performance of the GP method is sig-

Table 3.6: Comparison between IGRF-GP and GA

Dataset	Method	#Features	NB	J48	Random Forest	SVMs
OVA2	GA	189	86.56	89.32	92.88	99.20
	GP	20	94.46	97.23	98.42	97.23
OVA1	GA	229	85.18	83.33	88.42	94.90
	GP	20	90.27	88.88	91.67	93.06
PAN	GA	23	53.59	56.35	49.72	54.69
	GP	20	65.74	64.08	74.03	71.27
ARC	GA	324	68.00	69.00	77.50	78.50
	GP	20	73.5	80.00	80.00	73.00
DS_a	GA	10	70.00	50.00	50.00	90.00
	GP	20	80.00	90.00	90.00	90.00
DS_b	GA	62	40.00	50.00	70.00	60.00
	GP	20	100.00	90.00	90.00	100.00
DS_c	GA	45	40.00	60.00	40.00	60.00
	GP	20	100.00	100.00	100.00	100.00
DS_d	GA	62	33.33	50.00	83.33	66.67
	GP	20	100.00	91.67	100.00	100.00
DS_e	GA	62	58.33	79.16	83.33	66.67
	GP	20	100.00	91.67	100.00	100.00
DS_f	GA	62	66.67	96.67	60.00	76.67
	GP	20	100.00	96.67	100.00	100.00
DS_g	GA	62	66.67	41.67	33.33	58.33
	GP	20	100.00	100.00	100.00	100.00
DS_h	GA	62	41.67	79.16	62.50	75.00
	GP	20	95.83	91.67	100.00	91.67
DS_i	GA	62	50.00	86.67	76.67	70.00
	GP	20	100.00	90.00	100.00	100.00

nificantly better than that of the GA method. This further shows that the proposed GP method can outperform the GA method in terms of both the classification performance and the number of features. It also can be noticed that the performance of GA degrades when the number of examples is small. Therefore, GA might not be suitable for these datasets where the available number of examples is very small compared to the number of features.

3.6.4 Overlap between top-ranked features

Table 3.7: Percentage of overlap of the top 100 features of IG and RF.

Overlap (%)												
OVA2	OVA1	PAN	ARC	DS_a	DS_b	DS_c	DS_d	DS_e	DS_f	DS_g	DS_h	DS_i
46	0	41	11	20	6	22	19	48	56	16	43	39

Table 3.8: Percentage of overlap of the top 20 features across the 30 runs.

Feature	Overlap (%)												
	OVA2	OVA1	PAN	ARC	DS_a	DS_b	DS_c	DS_d	DS_e	DS_f	DS_g	DS_h	DS_i
1	76.67	60.00	100.00	76.67	100.00	70.00	100.00	56.67	70.00	73.33	66.67	70.00	90.00
2	67.67	56.67	93.33	76.67	100.00	56.67	93.33	46.67	63.33	63.33	53.33	60.00	56.67
3	63.33	53.33	73.33	53.33	100.00	50.00	83.33	46.67	60.00	46.67	46.67	50.00	50.00
4	63.33	53.33	50.00	50.00	100.00	50.00	70.00	46.67	50.00	43.33	46.67	46.67	40.00
5	56.67	50.00	50.00	46.67	100.00	50.00	66.67	43.33	46.67	40.00	46.67	43.33	36.67
6	50.00	46.67	50.00	46.67	100.00	50.00	66.67	43.33	46.67	40.00	46.67	43.33	36.67
7	46.67	43.33	46.67	46.67	100.00	46.67	66.67	40.00	43.33	40.00	46.67	43.33	36.67
8	43.33	40.00	43.33	43.33	100.00	46.67	63.33	40.00	43.33	40.00	43.33	43.33	36.67
9	43.33	40.00	40.00	43.33	100.00	46.67	63.33	40.00	40.00	36.67	40.00	40.00	36.67
10	43.33	36.67	40.00	43.33	100.00	46.67	60.00	40.00	40.00	36.67	40.00	40.00	36.67
11	40.00	36.67	36.67	43.33	100.00	46.67	60.00	40.00	40.00	36.67	40.00	40.00	36.67
12	36.67	33.33	36.67	40.00	100.00	46.67	60.00	40.00	40.00	36.67	40.00	40.00	36.67
13	36.67	33.33	36.67	40.00	100.00	46.67	60.00	40.00	40.00	36.67	40.00	36.67	36.67
14	36.67	33.33	36.67	40.00	100.00	46.67	60.00	40.00	36.67	36.67	40.00	36.67	33.33
15	36.67	33.33	36.67	40.00	100.00	43.33	56.67	36.67	36.67	36.67	40.00	36.67	33.33
16	36.67	33.33	33.33	40.00	100.00	43.33	56.67	36.67	36.67	36.67	40.00	36.67	33.33
17	36.67	33.33	33.33	40.00	100.00	43.33	56.67	36.67	36.67	33.33	36.67	36.67	33.33
18	36.67	33.33	33.33	40.00	100.00	43.33	56.67	36.67	36.67	33.33	36.67	36.67	33.33
19	36.67	33.33	33.33	40.00	100.00	43.33	53.33	36.67	36.67	33.33	36.67	36.67	33.33
20	33.33	33.33	33.33	36.67	100.00	43.33	53.33	0.13	36.67	33.33	36.67	36.67	33.33

To further analyse the top ranked features, the amount of overlap between the top 100 features from RF and IG is shown in Table 3.7. This overlap between the features of two metrics benefits in the proposed approach, where the features are ranked according to the frequency of occurrence of each feature in the best evolved programs of GP. The reason is that a feature selected by both RF and IG indicates that it could be an important feature. In IGRF-GP, an extra score is given to a feature if it ap-

pears in more runs. Therefore, the 20 top ranked features are the features that appear more in the best evolving programs in more runs.

The overlap percentage between the top 20 features used in the 30 different runs of the GP method is shown in Table 3.8. As shown in Table 3.8, the top 5 features appear in the four MS datasets in at least 50% of the 30 runs, which indicates these features are very important features. For the LC-MS datasets, the overlap between the features are also high, which explains why the performance of the top 20 features of those datasets can achieve the perfect classification performance.

3.7 Chapter Summary

The overall goal of this chapter was to investigate the capability of GP for processing multiple tasks, which are feature selection, feature ranking, and classification on the high dimensional MS data. This goal was fulfilled by developing a two-phase GP approach. In the first phase, GP was used to select a smaller set of features from the top ranked features obtained by IG and RF, and improve the feature ranking of these features. In the second phase, GP was used for classifying the data based on the new top-ranked features. The proposed GP approach works by embedding two feature ranking metrics, which are IG and RF, and taking the top ranked features by these metrics in the terminal set in order to produce a new and smaller set of features. The results show that the proposed GP approach selected a smaller number of features than IG and RF. The top 20 features ranked by GP resulted in a better classification performance than all the original features and the top 100 or 20 features ranked by IG and RF individually for most of the thirteen problems using the NB, J48, random forest and SVMs classifiers. GP as a classifier has the potential to outperform NB, J48, random forest classifiers on these datasets. The results also suggest that combining multiple feature selection metrics using GP can improve the classification performance.

In the biomarker detection task, GP managed to achieve a detection rate better than that of IG and RF in all the LC-MS datasets. These datasets are characterised by a small number of samples and different between-class and within-class variabilities. This indicates that GP is a promising approach for this type of challenging task.

Chapter 4

Multiple Feature Construction

4.1 Introduction

Feature construction is the process of transforming the original input features into new features [107]. The new features have the potential to improve the classification performance and also reduce the dimensionality of the input space.

GP has been widely used for feature construction [62, 93, 107, 122] with promising results in terms of improving classification accuracy. Most of the GP based feature construction approaches were based on constructing a single feature. This single feature is used for classification, or it is used along with the original set of features. Some other methods construct multiple features from multiple different independent runs. Using the single constructed feature alone might not achieve acceptable classification accuracy. The combination of a single constructed feature along with the original set of features will increase the dimensionality [60, 107] while using multiple individuals to construct multiple features might also increase the computational time.

In this chapter, a new GP approach to constructing multiple features [7] is presented. The proposed approach uses GP to select a good subset of features and automatically construct new features from a single run. The

new approach is expected to further decrease the dimensionality of the selected features and improve the classification performance. This method is also evaluated according to its performance for biomarker detection on MS data.

4.1.1 Chapter Goals

The goal of this chapter 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. 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 a single 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. Do the constructed features perform better than the low-level selected features?

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

Chapter Organisation: The rest of the chapter is organised as follows. Section 4.2 describes the new GP approach. The experiment setup and the datasets description are presented in Section 4.3. Section 4.4 reports

the experimental results along with discussions. Section 4.5 contains the chapter summary.

4.2 GP for Construction of Multiple Features

4.2.1 Algorithm Description

GP can automatically produce multiple outputs from its sub-trees and root nodes [190]. The use of subtree outputs (internal nodes) has shown to be effective for classification problems [190], which encourages us to use the internal nodes outputs for constructing new features. Unlike other approaches that use only the output of root node of the evolved tree as the constructed feature, the sub-tree nodes' output is also used as a high-level features here. This will help in the construction of more features from a single evolved tree and not from multiple trees (runs), and, therefore, reduce the computational cost. The multiple high-level features can also potentially 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 multiplication ($\times$) function, the constructed feature is the output of the multiplication of those two features. In the existing GP approaches, the final tree output from the tree node is the only constructed feature, but in our new approach, different branches of the tree are also treated as constructed 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 4.1. The steps of the proposed algorithm are described in Algorithm 2.

The process is as follows:
divide the datasets into a training set and a test set using ten-fold cross-

Algorithm 2 The GP multiple feature construction algorithm

Require: D, a dataset that contains a vector of instances with m original features.
Ensure: F, of a set of high-level features.

begin
Divide D into ten folds.
While maximum number of generations is not reached; **do**;
{

For j=1 to 10; **do**

{

Take the current fold as a test set and the rest folds as a training set

Randomly Initialise the population (P)

while Maximum generation is not reached **do**

Evaluate the fitness

Select the individuals using the selection method

Generate new population ($CHILD$) using the genetic operators

Save the high-level features from the best individual of each fold in F

end For

}

Compute the average of all the folds

}

end While
Remove the similar features from F
Use the features in F to project test set
return a vector F that contain the constructed features
Calculate the test set classification accuracy of the different solutions
end

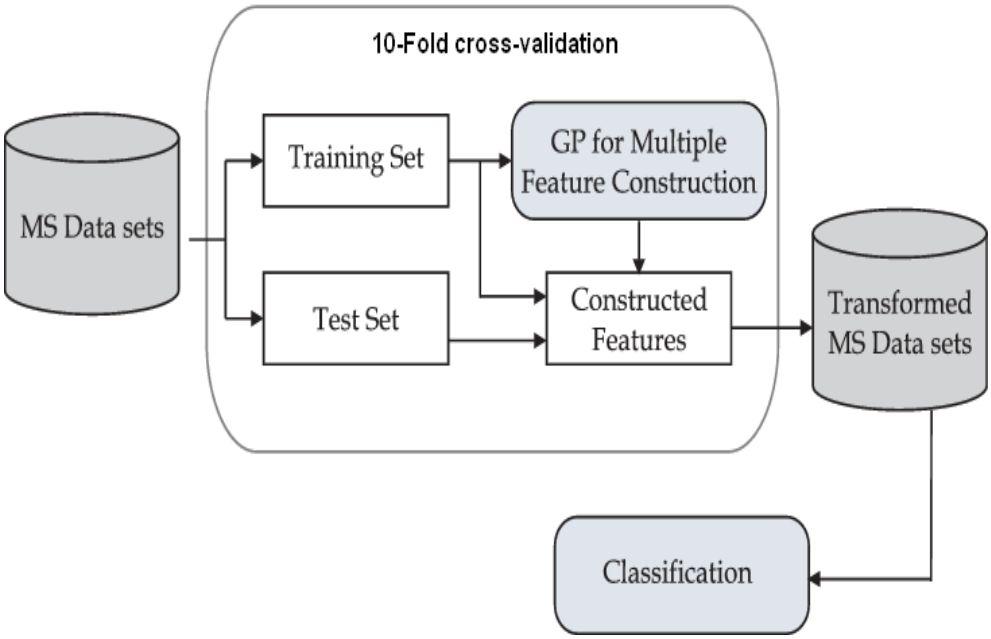


Figure 4.1: Overview of the GP-multiple feature construction system.

validation. Use the training set with GP to construct new features, where the quality of the features is measured using their discriminating power between the classes, which is calculated using Fisher 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 newly constructed features are used to project both the training and test sets, where different classification algorithms can be used to evaluate new features. Figure 4.2 shows an example of how the features are constructed from an evolved GP tree. As shown in Figure 4.2, 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.

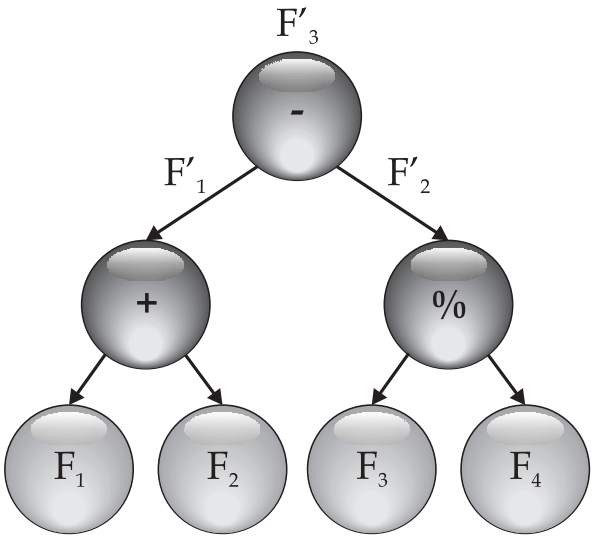


Figure 4.2: Example of how the features are constructed.

4.2.2 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 [62], but the computational cost is higher as it requires training a classifier for each individual of the population. Meanwhile, the classification performance depends more on the discrimination power of the classifier. Designing a fitness function as an embedded method can therefore avoid those disadvantages.

The Fisher criterion [39] works by maximising the *between-class* scatter and minimising the *within-class* scatter.

For a two-class problem, the Fisher criterion is defined as

$$\text{Fisher criterion} = \sum_{n=1}^{N} |\frac{\mu_i - \mu_j}{\sigma_i^2 - \sigma_j^2}| \tag{4.2.2.1}$$

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 calculated for each adjacent pair of classes based on Equation (4.4.2.1) and the summation of those pairs is the final value of Fisher criterion.

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

$$F_p = \frac{\text{Fisher criterion}}{P_{value}} \tag{4.2.2.2}$$

In Equation (4.4.2.2), 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 maximise the fitness. Therefore, during the evolution, the p-value is minimised, and the Fisher criterion is maximised (i.e. the between-class distance is maximised and the within-class distance is minimised).

4.3 Experiment Setup

This section explains the design of the experiments including the datasets that were used in the experiments, the terminal set, the function set, and the GP parameters.

4.3.1 Datasets and Preprocessing

To test the effectiveness of the new GP approach, eight MS datasets were used. In this section, the datasets characteristics and the preprocessing will be explained. Table 4.1 summarises the characteristics of the datasets.

Table 4.1: Datasets characteristics

Dataset	# Features	# Samples	#Classes
PAN	6771	181	2
OVA2	15,154	253	2
OVA1	15,000	216	2
Pros	15,000	322	4
TOX	7105	115	4
ARC	10,000	200	2
Apple-plus	773	40	4
Apple-minus	365	40	4

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

- PAN dataset [20]: This dataset is acquired using a SELDI-TOF system.The preprocessing steps include baseline subtraction where piecewise linear interpolation is used for the regression of the baseline. Afterward, filtering and normalisation are performed using Gaussian filter and area under the curve, respectively.

- OVA1 and OVA2 datasets [133]: Both of these datasets were analysed 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 PAN dataset, the preprocessing of these two datasets involves baseline adjustment, filtering, and normalisation. The final step performed is the alignment to remove the fluctuation in the m/z values. The ovarian cancer high-resolution dataset contains 121 cancer and 95 healthy samples, while the low-resolution dataset contains 162 cancerous samples and 91 healthy samples.

- Pros dataset [86]: Samples of three different stages of prostate cancer and healthy samples were analysed 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).

- TOX dataset [134]: Serum samples with toxicity-related biomarkers were analysed using the SELDI-TOF mass spectrometer. The dataset consists of four classes which are: definite positive (34 samples), definite negative (28 samples), probable positive (10 samples) and probable negative (43 samples). The Pros and TOX datasets were already baseline adjusted. Therefore, both of the datasets were only filtered and normalised.

  The above five datasets were downloaded from FDA-NCI Clinical Proteomics Program[1]. Those datasets are already binned. Therefore, the number of features remains the same after preprocessing. Matlab [114] bioinformatics toolbox was used to perform the preprocessing of the data.

- ARC dataset [64]: Three different MS datasets were combined to produce the ARC dataset that contains 100 samples of cancer patients and 100 healthy samples. The dataset is available after preprocessing, and it is downloaded from the UCI machine learning repository [64].

- Apple extracts datasets [169]: These two datasets are metabolomics datasets where twenty apples were analysed using LC-MS technology. Four classes are created from the twenty apples, each class containing ten samples. Three classes contain a mixture of known compounds (biomarkers) during the fourth class is not spiked-in with those compounds. The negative and positive ion modes form the

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

two different datasets. The total number of biomarkers is five and twelve in the negative and positive ion modes, respectively. The datasets are available in NetCDF format, and it is preprocessed using XCMS [152] with the settings described in [169].

4.3.2 GP Settings

The standard tree-based GP is used in the experiments where each node outputs a single floating point [17,145]. The initial population is generated using the ramped half-and-half method [92].

The m/z and retention time variables represent the feature identities of the compounds and the corresponding intensity are the feature value [176]. Therefore, the terminal set is composed of the intensity variables (features) which represent the abundance of the compound in the data and a constant value which is a randomly generated number between [-1,1]. For each sample in a dataset, a single floating-point value is produced by the program at the root of its evolved tree [92]. The function

Table 4.2: GP settings

Function set	$+, -, \times, \%, max, min, IFTE$
Variable terminals	features
Constant terminals	random numbers
Initialization method	Ramped Half-and-Half
Tree Depth	2-10
#Generations	50
Mutation rate	20%
Crossover rate	80%
Elitism	Yes%
Population Size	2048
Selection type	Tournament
Tournament Size	7

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 that returns zero for dividing by zero. All the function set members take two arguments except for $IFTE$, which takes three arguments and it returns the second argument if the first argument is negative and returns the third argument otherwise. The evolution terminates at a maximum number of generations of 50. The size of the population is set to 2048. The tree depth has been set between 2 and 10. The basic crossover and muation operator are used here. The crossover and mutation rates are set to 80% and 20%, 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, thus, keeping the performance is monotonically increasing during the evolution [153]. The ECJ [108] package was used in our experiments for running GP. Table 4.2 shows the various settings for the new method.

4.3.3 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 [67] 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 the artificial neural network (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 leaf 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 neighbours 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 Java implementation of the C4.5 decision tree classifier.

7. Decision table (DT): The possible subset of features is 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 [88].

4.3.4 Comparison Methods

The performance of the proposed GP method is compared with three methods. Firstly, the original set of features of each dataset are used with the seven classifiers for classification. Secondly, the proposed method 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. The features selected by the proposed method are compared with the features constructed by it. This method is annotated as Method_1. Finally, a GP-based feature selection method (Method_2) is also used for comparison [3] 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 datasets. Method_1 and Method_2 select the features that 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 [3]. 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) with 90% significance level is performed to compare the classification performances of the three methods.

4.4 Results and Discussions

In Table 4.3, 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 4.3. "Avg#" shows the average number of selected or constructed features from each method. The evaluation of the seven classifiers is done by means of ten-fold cross validation. The accuracy of using all the original features is also reported in the same table and is shown by "All".

In Table 4.3 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 8GBytes.

4.4.1 Comparison of the Constructed Features with All the Original Features

As shown in Table 4.3, for all the datasets 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 any result 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 datasets except

Table 4.3: 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	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**$^{\tau\star}$		98.88	95.13 ±2.00		97.83	94.11 ±1.51	
	K-NN	55.80		98.88	95.56 ±1.80$^{\tau}$		98.88	95.70 ±2.16		100.0	**95.73 ±1.61**	
	NB	51.38		77.77	**62.50 ±5.76**$^{\tau\dagger\star}$		64.44	57.26 ±3.44		56.52	54.57 ±0.91	
	J-48	50.82		92.77	87.82 ±2.77$^{\tau}$		94.44	**88.48 ±2.41**		92.93	88.42 ±2.16	
	DT	61.32		81.15	**72.36 ±5.33**$^{\tau\dagger\star}$		80.44	71.17 ±4.40		75.61	64.57 ±6.12	
OVA1	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**$^{\tau\dagger\star}$		100.0	98.33±0.92		100.0	97.17 ±1.08	
	K-NN	86.57		100.0	**99.85 ±0.43**$^{\tau\dagger\star}$		100.0	98.97 ±0.83		99.10	96.10 ±1.38	
	NB	83.79		100.0	**94.16 ±4.38**$^{\tau\star}$		98.13	93.97 ±2.36		93.11	88.22 ±3.34	
	J-48	86.57		100.0	**96.76 ±2.72**$^{\tau\dagger\star}$		98.59	95.28 ±2.00		97.71	93.93 ±1.77	
	DT	82.87		97.93	**94.74 ±2.16**$^{\tau\dagger\star}$		97.00	93.69 ±3.20		96.21	92.00 ±3.45	
OVA2	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$^{\tau\star}$		100.0	**99.73±0.37**		100.0	99.21 ±0.46	
	K-NN	92.09		100.0	**99.95 ±0.20**$^{\tau\dagger\star}$		100.0	99.76 ±0.47		100.0	99.01 ±0.68	
	NB	76.28		99.21	**96.91 ±1.42**$^{\tau\dagger\star}$		97.22	94.48 ±2.22		96.87	91.12 ±2.78	
	J-48	95.65		100.0	**98.76 ±1.06**$^{\tau\dagger\star}$		100.0	98.20 ±1.06		99.22	97.28 ±0.93	
	DT	92.49		100.0	97.97 ±1.39$^{\tau\star}$		100.0	**97.98 ±0.49**		100.0	97.23 ±2.43	
ARC	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$^{\tau\star}$		100.0	**97.50±1.27**		100.0	96.68 ±1.53	
	K-NN	84.50		100.0	**96.73 ±1.48**$^{\tau}$		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**$^{\tau\star}$		93.50	90.15 ±2.59		94.50	88.65 ±2.45	
	DT	71.50		92.00	83.51 ±4.64$^{\tau}$		93.67	84.15 ±3.50		94.00	**85.78 ±2.35**	
Pros	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**$^{\tau\dagger\star}$		99.37	97.72 ±0.98		100.0	97.74 ±1.04	
	NB	58.13		84.91	**75.37 ±6.13**$^{\tau\dagger\star}$		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**	
TOX	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**$^{\tau\dagger\star}$		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**$^{\tau\dagger\star}$		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	

for Apple-plus and Apple-minus datasets, where their performances 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 datasets excluding the LC-MS datasets (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 of using all the original features. On the OVA1, OVA2, Apple-plus and Apple-minus datasets, the constructed features achieved 100.0% accuracy with most of the classifiers. For other datasets, the improvement of the accuracy of the seven different classifiers is 25.97-41.44% on PAN, 14.5-27.5% on ARC, and 2.55-27.79% on Pros and TOX.

Also, to improve the classification performance, the proposed GP approach also helps in reduction of dimensionality. For example in Pros dataset, 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 TOX dataset with the 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 problem between the number of samples in each class and the embedded feature selection capability of J-48.

4.4.2 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, in the TOX dataset, GP-Constructed is significantly better than Method_1 on all the seven classifiers except for NB-tree, where their results are similar.

On the PAN dataset, the classification performance of the features constructed by the new approach (i.e. shown by GP-Constructed) is significantly better than that of the features selected by Method_1 (i.e. shown by Method_1) when used with MLP, NB and DT classifiers. On the OVA1 and OVA2 datasets, GP-Constructed is significantly better than Method_1 in four and seven classifiers, respectively. On the Pros dataset, GP-Constructed is significantly better than Method_1 when using RF, KNN, NB and DT in the Pros dataset. For the two Apple datasets, GP-Constructed is significantly better than Method_1 when using five and two classifiers, respectively.

In terms of the dimensionality reduction, GP-Constructed further decreased the number of features over Method_1 on all the eight datasets. The average number is reduced by 7.9-29.26 in different datasets. Meanwhile, GP-Constructed either significantly improved or maintain the similar performances of the low-level selected features in almost all cases.

Comparing the proposed method with Method_2: In almost all datasets, 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 OVA1, OVA2, and Apple-plus datasets with almost all the seven classifiers, and on the TOX and Apple-minus dataset with five of the seven classifiers.

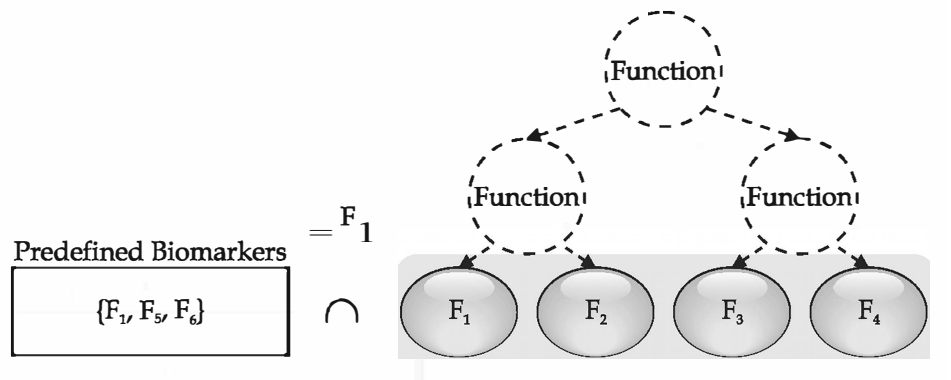


Figure 4.3: Biomarker detection approach.

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 datasets. With the smaller set of constructed features, the new approach still achieves similar or better classification performance than Method_2 in almost all cases.

4.4.3 Biomarker Identification

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

Figure 4.3 shows an example of the approach used to count the number of identified biomarkers. As shown in Figure 4.3, the intersection between the selected features in the terminal nodes of the tree and the predefined set of biomarkers are used as an evaluation of the biomarker identification task.

Table 4.4 shows the biomarkers in Apple-plus and Apple-minus datasets (positive and negative modes of the ions). 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

Table 4.4: Identified spike-in biomarkers by the proposed GP method and Method_1 for the Apple datasets. The biomarkers are identified using their m/z values.

m/z values in Apple-plus dataset (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 dataset (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

shown in Table 4.4. As shown in Table 4.4, GP identified the complete set of biomarkers in Apple-minus datasets. Method_2 detected only two biomarkers in 93.33% and 86.67% of the runs, respectively. For Apple-minus dataset, 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 dataset, nine out of the twelve biomarkers (75%) are detected by the proposed GP method. 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.

4.5 Chapter Summary

The goal of this chapter 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 maximising the significant discrimination between different classes. The performances of the high-level constructed features are compared to those of 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 datasets 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 datasets. 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 dataset and 75% of the predefined biomarkers in the Apple-plus dataset. Due to its better classification and biomarker identification performance, the new GP can be successfully applied to this task.

In the next chapter, multi-objective GP methods for feature selection and construction are proposed. The multi-objective feature construction is an extension of the method proposed in this chapter that aims to keep the trade-off between the number of high-level features constructed and the classification performance.

Chapter 5

Multi-Objective Feature Manipulation

5.1 Introduction

Many feature selection techniques have been proposed to detect the potential biomarkers in MS data [26, 102–104, 123]. Despite the promise of the previously proposed methods, none of these methods considered the number of features as an important independent objective to optimise. Some studies considered the relative importance of the number of features to classification accuracy in a single fitness function. The major limitation of these approaches is the prior specification of the relative importance of each objective into a single-objective fitness function. Multi-objective optimisation offers the solution to the optimisation of different conflicting objectives simultaneously without the need to consider the relative importance in advance. Section 5.2 of this chapter proposes the first attempt to use GP as a multi-objective approach to biomarker detection.

Although our previously proposed feature construction approach in Chapter 4 has shown the effectiveness of the new features on improving the classification performance, the number of features constructed is still high. Section 5.3 aims to extend the work in Chapter 4 to consider the

trade-off between the number of features constructed and the classification accuracy through the use of multi-objective optimisation.

5.1.1 Chapter Goals

The overall goal of this chapter is to develop GP-based multi-objective feature selection and construction approaches to classification of MS data. In feature selection, the proposed GP method uses ideas from NSGAII [27] and SPEA2 [192] to evolve models that keep the balance between the conflicting objectives. We notate these methods as $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$. The main goal here is to evolve a Pareto front of non-dominated solutions, which include a small number of selected original features and achieve a better classification accuracy than using the whole set of features.

In feature construction, a single evolved tree is used to construct multiple features by replacing the original features with the constructed features after combining them using the GP functions. Multi-objective optimisation is used to reduce the number of constructed features while keeping the high classification accuracy. We notate these methods as $NS\text{-}GPMOFC$ and $SP\text{-}GPMOFC$.

In both approaches, an embedded approach is used to take the advantages of the low computational cost and better classification accuracy.

Precisely, we will investigate the following:

- whether using GP as a multi-objective approach to feature selection can evolve better non-dominated solutions than using the single objective GP algorithm,

- whether using multi-objective GP feature selection methods can select feature subsets that improve the classification performance and reduce the number of features more effectively than using the traditional multi-objective algorithms,

- whether using multi-objective optimisation can reduce the number of constructed features and at the same time maintain the good classification accuracy, and

- whether the GP-based method for construction with multiple objectives can further improve the feature subset evolved by the multi-objective GP feature selection method.

Chapter Organisation: The rest of the chapter is organised as follows. Section 5.2 describes the GP-based multi-objective feature selection algorithm. Section 5.3 explains the GP-based multi-objective feature construction method. Section 5.4 gives an overview of the two systems. Section 5.5 describes the experimental design that includes the settings and the MS datasets used. Section 5.6 presents the experimental results of discussions. Section 5.7 concludes the chapter.

5.2 The GP Multi-objective Feature Selection Approach

In this section, we investigate a new approach to feature selection for MS data with the aim of biomarker detection using multi-objective GP, with two main objectives to explore the Pareto front of feature subsets. The objectives here are maximising the classification accuracy and minimising the number of features used in each individual of the population. As mentioned earlier, an embedded approach is taken in the proposed algorithm. GP is employed here as a classifier as well, and the number of correctly classified instances in the training set is stored in a memory list. The classification accuracy is used to assess the first objective. The second objective here is to minimise the cardinality of the selected features (number of features selected automatically in the GP tree). When a new solution is evolved, it is compared to the other solutions stored in the memory

list. If the evolved solution is not worse in both objectives and it is better than a solution in the list in at least one of the objectives, it will dominate that solution. Pareto optimal contains the set of non-dominated solutions where a specific solution can not improve any of the objectives without degrading at least one of the other conflicting objectives [126]. The non-dominated solution forms the Pareto front in which no solution can be judged better than the others.

5.2.1 Pareto Fitness Schemes in NS-$GPMOFS$ and SP-$GPMOFS$

In evolutionary multi-objective optimisation, solutions are usually ranked according to their performance on the different objectives to measure the Pareto dominance. The Pareto dominance is measured through the dominance rank or dominance count [192] (or both) of a certain solution. Dominance rank of a solution is the number of solutions that dominates this solution, while the dominance count is the number of solutions that a given solution dominates. A solution with a lower number of solutions that dominate it (lower rank) and a higher count is a better solution. We propose two mechanisms to measure the Pareto fitness. The first uses the dominance rank of a solution S_i for evaluating the fitness which is similar to the idea of NSGAII [27], i.e., the number of other solutions in the population that dominate S_i, and we annotate this method as NS-$GPMOFS$. Similar to SPEA2, the second mechanism uses both dominance rank and dominance count in the Pareto refined fitness, and this method is annotated as SP-$GPMOFS$.

5.2.2 Crowding Distance Measure

In addition to the previously mentioned Pareto dominance measures used in the fitness, a crowding distance measure is used to generate more diversity among the population [15]. The crowding distance used is the Manhattan distance between the solutions. This distance measure is used

only when two or more solutions have equal Pareto dominance measures, which means that if solutions have equal rank then the solution with better crowding distance is selected. The crowding distance is the average distance between the two solutions with each of the objectives, where a lower distance indicates a better result.

5.2.3 NS-$GPMOFS$ and SP-$GPMOFS$ Algorithms

Algorithm 3 shows the pseudocode of $GPMOFS$ algorithms. The input is D, the dataset, and the output is the Pareto front archive of solutions (PF). At each generation, the parent and offspring populations are merged. The fittest individuals (according to the two objectives) in this merged population acts as the new population ($CHILD$) in the next generation. The population is reduced to size N (original size of the population) using dominance rank and crowding distance for NS-$GPMOFS$. While for SP-$GPMOFS$ dominance rank, dominance count and the crowding distance are measured. The size of $CHILD$ is the same as the size of the original population and it is produced using the traditional genetic operators (crossover and mutation operators). In case of SP-$GPMOFS$, the size of PF (Pareto front solutions) is kept fixed while in NS-$GPMOFS$ it does not have a specific size. Another difference between using NS-$GPMOFS$ and SP-$GPMOFS$ is the use of elitism in SP-$GPMOFS$, which is not used in NS-$GPMOFS$. The non-dominated solutions in $CHILD$ are identified and copied to PF. These steps are repeated until the maximum number of generations is reached. At the end of the evolutionary search, the solutions of PF are used to project the datasets and passed for evaluation. The evaluation is done through both classification accuracy and the number of features used in each solution in the archive.

Algorithm 3 Pseudo-Code of $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$

Require: D, a dataset that contains a vector of instances with m original features.
Ensure: PF, a Pareto front (PF) of a set of solutions (low-level features).

begin
Divide D into training and test sets.
Initialise the population (P)
while Maximum generation is not reached **do**
Evaluate the two objectives of each individual { // Acc, $|F|$}
Select the individuals using the selection method
Generate new population ($CHILD$) using the genetic operators
if $NS\text{-}GPMOFS$ is used **then**
Non-dominated sorting of the individuals based on ranking and the crowding distance
else if $SP\text{-}GPMOFS$ **then**
evaluate the individuals based on ranking, count, and the crowding distance
end if
Copy both $CHILD$ and P to $Archive$
Identify the individuals who have non-dominated solutions in $Archive$ and add to Pareto front (PF)
Select a population of size N based upon ranking and crowding distance
Generate new population ($CHILD$) using the genetic operators
end while
Use the solutions in PF to project the test set
Calculate the test set classification accuracy of the different solutions
Calculate the number of selected features in each solution in PF
return a vector S that contain the number of features and classification accuracy of each solution in PF

5.3 The GP Multi-objective Feature Construction Approach

The difference between $GPMOFS$ and $GPMOFC$ is that instead of using the original selected features; $GPMOFC$ constructs new high-level features from the original features (resulted features from the tree branches). In addition to the features constructed from the branches, the final feature constructed from the root node of the tree is also used.

5.3.1 SP-$GPMOFC$ and NS-$GPMOFC$ Algorithm

Algorithm 4 describes the algorithms of SP-$GPMOFC$ and NS-$GPMOFC$. The two algorithms are similar to the feature selection algorithms (SP-$GPMOFS$ and NS-$GPMOFS$) except for the final feature set. The difference here between the two algorithms for feature selection and construction is that instead of using the original features selected, the high-level features are constructed to optimise the second objectives.

5.4 Overview of the Two Systems

As shown in Figure 5.1, after preprocessing of the MS spectra datasets, the system for $GPMOFS$ starts by dividing the dataset into training and test sets.Each program in the population uses a subset of features in its tree terminal nodes and generates the fitness value. The fitness value (classification accuracy) is measured by GP individual classifier's accuracy that is passed as a fitness value to measure the dominance. Dominance rank, dominance count and crowding distance are used to measure the dominance of the solutions. After the fitness calculation, the fitness value of each solution is compared to the Pareto front archive. If the solution in the archive is dominated by the new solution, the new solution will replace it in the archive. Each solution in the Pareto front has a subset of features

Algorithm 4 Algorithm of $NS\text{-}GPMOFC$ and $SP\text{-}GPMOFC$

Require: D, a dataset that contains a vector of instances with m original features.
Ensure: PF, A Pareto front (PF) (solutions with high-level features).

begin
Divide D into 50% for training and 50% testing.
Randomly Initialise the population (P)
while Maximum generation is not reached **do**
Save the high-level features resulting from the branches and the root of the individual tree
Evaluate the number of constructed features and Acc of each individual
Select the individuals using the selection method
Generate new population ($CHILD$) using the genetic operators
if $NS\text{-}GPMOFC$ **then**
Non-dominated sorting of the individuals based on ranking and the crowding distance
else if $SP\text{-}GPMOFC$ **then**
Non-dominated sorting of the individuals based on ranking, count and the crowding distance
end if
Copy both $CHILD$ and P to $Archive$
Identify the individuals who have non-dominated solutions in $Archive$ and add to Pareto front (PF)
Select a population of size N based upon ranking and crowding distance
Generate new population ($CHILD$) using the genetic operators
end while
Use the solutions in PF to project test set
Calculate the test set classification accuracy of the different solutions
Calculate the number of high-level features in each solution in PF
return a vector S that contain the number of high-level features and classification accuracy of each solution in PF
end

that were selected in the terminal nodes. The Pareto front solutions are used to project the datasets, therefore if the size of the archive is n, there will be n projected datasets. To test the subsets of features, the test set is evaluated using GP classifier. As explained earlier, the main difference between $GPMOFS$ and $GPMOFC$ is the use of low-level and high-level features.

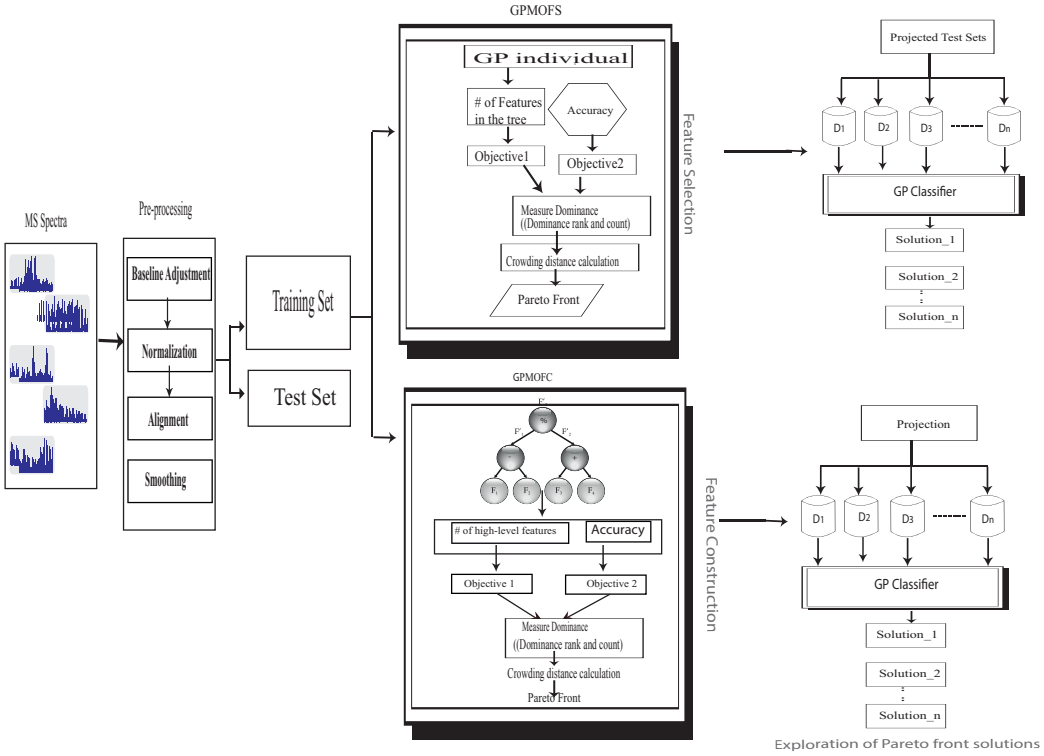


Figure 5.1: General overview of the multi-objective approaches

5.4.1 Fitness Function

For both algorithms, the fitness values used as objectives are the following: Firstly, the overall classification accuracy (Acc) and secondly, the number of features either selected or constructed. Acc is defined as:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

Table 5.1: Summary of the Datasets

Dataset	No. of spectra	No. of features
OVA1	216(121+95)	15000
OVA2	253(162+91)	15154
PAN	181(80+101)	6771
ARC	200 (100+100)	10000
TOX	62(28+34)	45200
HCC	150 (78+72)	36802
DGB	128 (78+25)	16075
Pros	253 (63+190)	15000
Appleminus	40(10+10+10+10)	365

where TP, TN, FP and FN are the true positives, true negatives, false positives, and false negatives, respectively. For each instance of the training set, if the output of the program is less than or equal to zero then the instance is classified as class 1, otherwise it is classified as class 2.

Thus, in both $GPMOFS$ and $GPMOFC$, the first objective is to maximise the classification accuracy. The second objective used is to minimise the cardinality of the feature subset selected or constructed by each GP tree in the terminal nodes $|F|$.

5.5 Experiments Design and Settings

This section explains the MS datasets used to test $GPMOFS$ and $GPMOFC$, GP operators and parameters, benchmark algorithms used for comparison reasons, and the evaluation criteria.

5.5.1 MS Datasets

To test the effectiveness of the proposed GP multi-objective approach, nine different MS datasets are used. Table 5.1 summarises the details of the datasets used in the experiments.

- OVA1 and OVA2 [133]: OVA1 is composed of 216 spectral instances where 121 spectra are cancerous samples and 95 spectra are healthy ones, while OVA2 consists of 253 spectra with 162 spectrum in the cancer class and 91 in the unaffected class. The number of features is 15000 and 15154 in OVA1 and OVA 2, respectively.

- PAN [20]: The dataset has 181 spectral examples, where 80 are in the affected class and 101 are in the healthy class. The number of features in each spectrum is 6771.

- ARC [22]. ARC is generated from three merged MS datasets (two prostate and one ovarian cancer dataset) with 100 spectra from cancerous class and 100 from normal class. Each spectrum has 10000 features in each spectrum.

- TOX [134]: The dataset consists of 62 spectra (28 in the positive and 34 in the negative class) and each spectrum has 45200 m/z readings.

- HCC [140]: HCC has 150 spectra (78 affected and 72 non-affected) with 36802 features in each spectrum.

- DGB [22]. This dataset contains three groups of samples (78 healthy control samples, 25 hepatocellular carcinoma and 25 chronic liver samples). The total number of features is 16075.

- Pros dataset [86]: This dataset 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). The number of features in Pros is 15000. For DGB and Pros datasets, we used only two classes of instances.

- Appleminus: This dataset is composed of 365 features with ten instances of each class. Three classes contain five predefined biomarkers, and the last class is not spiked-in. Only one of the spiked-in classes and the non-spiked class are used in our algorithms.

Several preprocessing steps were applied to each of the datasets. ARC datasets are available after preprocessing. The preprocessing of MS data is important to convert the data to a homogeneous matrix which can be used for feature selection and classification of the data [22]. The preprocessing steps used in our experiments include baseline adjustment, spectrum normalisation, alignment and filtering with different parameters for each dataset. The baseline removal is used to remove the low-range noise. The baseline is estimated by passing a window on the spectra and the minimum m/z values are calculated. A piecewise linear interpolation method is used for the regression of the baseline. To make the intensity values range the same, normalisation is performed. The normalisation of the spectra is done by calculating the area under the curve [134] and rescaling the spectra to have a maximum intensity value of 300. This is done by using the $msnorm$ function in the Matlab toolbox [114]. After normalisation is performed, alignment of the peaks is performed to match the similar peaks across all the spectra. Finally, smoothing of the spectra is done to remove the low signal fluctuation. Smoothing is done via a Savitzky-Golay filter. Pros and TOX datasets were already baseline adjusted. Therefore, both of the datasets were only filtered and normalised. Table 5.2 shows the running parameters of the preprocessing steps used with each of the datasets. The parameters are selected based on the original papers of the datasets [20,22,134,140]. For the spike-in Appleminus dataset, the preprocessing is previously explained in Chapter 4.

5.5.2 Performance Evaluation

GP as a classifier is used to test the selected features in each solution in the archive on the test sets.

5.5.3 Genetic Operators and Parameters

In the experiments, we adopt the standard tree-based GP which produces a single floating point number as a result of the fitness evaluation [92]

Table 5.2: Preprocessing running parameters

	OVA1 & OVA2	TOX	PAN	HCC	DGB	Pros
Window size for baseline removal	500	-	200	50	200	-
Smoothing frame size	5	6	3	6	6	3
Maximum intensity after normalisation	300					

for each instance in the dataset. Each of the output values is then used to determine the relevance of the subset of features used in the program and the classification accuracy of the genetic program. The initial population is generated using the ramped-half-and-half method [124]. The function set consists of the four standard elementary mathematical operators $\{+, -, \%, \times\}$ and also a square root $\sqrt{}$ operator. The $\%$ and $\sqrt{}$ are "protected" where $\%$ returns zero for division by zero and $\sqrt{}$ returns zero for negative numbers. The terminal set has only variable terminals that are the feature values. The population size is set to 1024. Crossover and mutation probabilities are 0.8, and 0.19, respectively, and tournament selection is used with the size of 7. The GP, NSGAII and SPEA2 implementations used in the experiments are based on the Evolutionary Computing Java-based (ECJ) package [108]. Other parameters for NSGAII and SPEA2 are set as the default values in the ECJ library. The evolution terminates at a maximum number of 20 generations.

For each dataset, the experiment is repeated for 30 independent runs with 30 different random seeds. Each run outputs a set of non-dominated solutions in the Pareto front. The 30 sets of non-dominated solutions from the 30 runs are combined to one set by removing the dominated solutions from the different sets.

Table 5.3 shows the various running parameters of the new GP method.

Table 5.3: GP running parameters

Function set	$+, -, \times, \%, \sqrt{}$
Variable terminals	features
Initialization method	Ramped half-and-half
Tree Depth	8-17
#Generations	20
Mutation rate	0.19
Crossover rate	0.8
Population Size	1024
Selection type	Tournament
Tournament Size	7

5.5.4 Benchmark Algorithms

$GPMOFS$ and $GPMOFC$ are compared to the following benchmark algorithms:

1. Standard (Single-Objective) GP method which is the standard GP classification framework using a fitness of the overall classification accuracy as a single objective to maximise. The features selected in the terminal nodes of the tree are treated as the selected features.

2. NSGAII: Multi-objective optimisation using NSGAII and Fisher criterion based class separability for feature selection [157]. The evaluation is done through both the higher Fisher criterion and the smaller number of features. The first objective which is maximising the Fisher criterion or the class separability, that is defined as,

$$\text{Fitness function = Fisher criterion} = \sum_{n=1}^{N} |\frac{\mu_i - \mu_j}{\sigma_i^2 - \sigma_j^2}| \tag{5.5.4.1}$$

where μ_i and μ_j are the means, 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. The second objective is minimising the number of features.

3. SPEA2: Multi-objective optimisation using SPEA2 and Fisher criterion to evaluate the selected features.

Similar to $GPMOFS$ and $GPMOFC$, the population size is set to 1024 and the number of generations is 20. For both NSGAII and SPEA2 , each individual is encoded as a binary vector. The length of the vector is equal to the total number of features in the dataset. Hence, if the bit is 0, this means that the feature is unselected.

5.6 Results and Discussions

This section provides the results and discussions of the proposed algorithms. Figure 5.2 shows the results of $GPMOFS$ compared to using the single objective GP method, and the SPEA2 and NSGAII, while Figure 5.3 shows the results of $GPMOFC$ compared to $GPMOFS$. The multi-objective methods have different numbers of non-dominated solutions. The results are the non-dominated solutions obtained from the 30 independent runs. The x-axis refers to the number of features selected by each method whereas the y-axis indicates the classification accuracy. Each figure is divided into a number of sub-figures where each sub-figure represents the results of each dataset.

5.6.1 Performance of $GPMOFS$

It can be noticed from Figure 5.2 that using SP-$GPMOFS$ has the potential to evolve solutions, which have better classification performance and a smaller number of features than using NS-$GPMOFS$ in seven out of the nine datasets. The proposed method also outperformed the single objective GP approach and the two benchmark multi-objective methods SPEA2

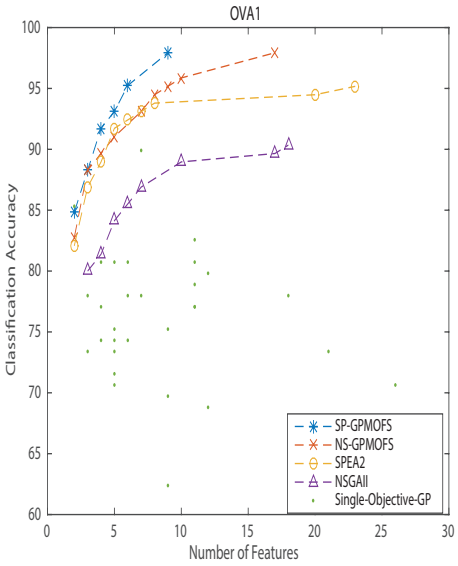
and NSGAII, on all the nine datasets. This supports our hypothesis that using multi-objective GP can improve the feature selection performance from both the classification accuracy and the number of features point of views.

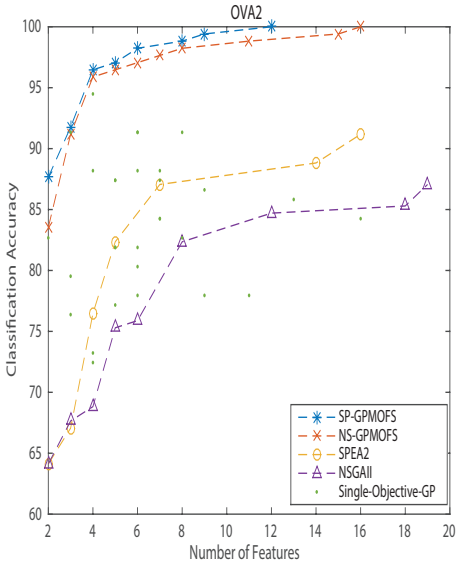
In some cases, $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$ have common solutions such as in the ARC, TOX, and HCC datasets during the left region of the front. Only in the TOX dataset, $NS\text{-}GPMOFS$ evolves solutions at the right region of the frontier which have better accuracy, but the number of features in these solutions are larger. In the Appleminus dataset, $NS\text{-}GPMOFS$ is the best followed by $SP\text{-}GPMOFS$. The single-objective GP method for the Appleminus dataset has evolved solutions with a large number of features and lower accuracy compared to the multi-objective approaches.

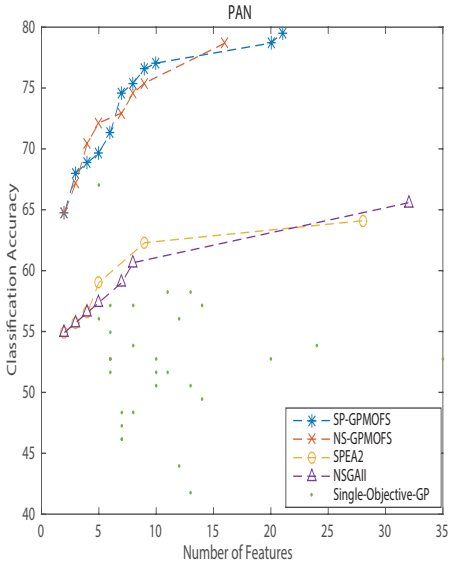
The multi-objective approaches SPEA2 and NSGAII for feature selection are both used with Fisher criterion for comparison to the proposed method. Comparing $NS\text{-}GPMOFS$ and $SP\text{-}GPMOFS$ with SPEA2 and NSGAII, it is clear that GP has improved the performance of both NSGAII and SPEA2 for feature selection. This can be explained by the GP capability to select the subsets of features that are more relevant to classification. Using multi-objective optimisation along with GP improve both objectives of reducing the number of features and having a better performance. This suggests that GP improves the capability of the multi-objective approaches through its ability to select the better subsets of features.

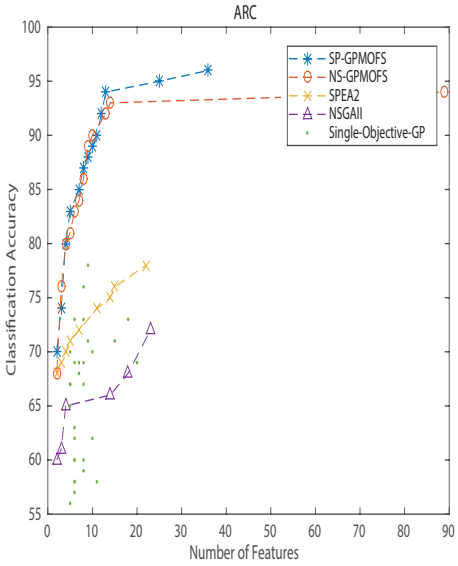
5.6.2 Comparison of $GPMOFS$ and $GPMOFC$

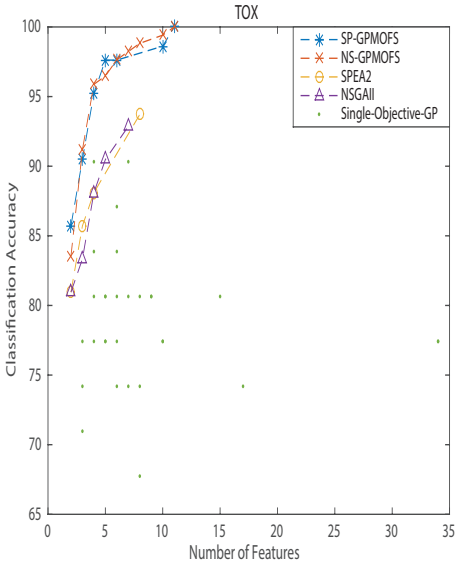
Considering the experimental results of $GPMOFC$ that are shown in Figure 5.3, it can be noticed that the multi-objective feature construction is better than the multi-objective feature selection in most cases. For OVA1, $SP\text{-}GPMOFC$ is the best with a smaller number of features. For PAN and HCC, feature construction approaches evolve better solutions than

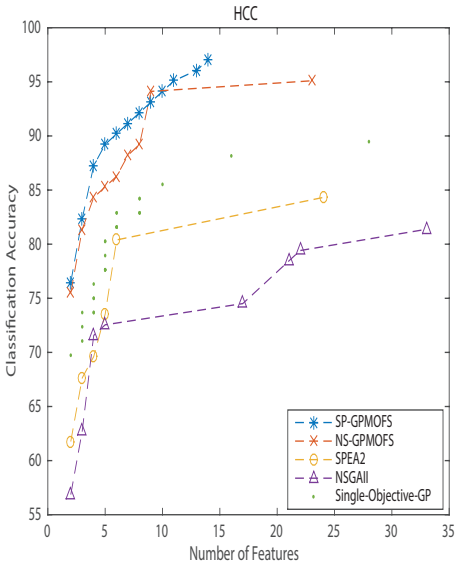

OVA1
Classification Accuracy
Number of Features
SP-GPMOFS
NS-GPMOFS
SPEA2
NSGAII
Single-Objective-GP


OVA2
Classification Accuracy
Number of Features
SP-GPMOFS
NS-GPMOFS
SPEA2
NSGAII
Single-Objective-GP


PAN
Classification Accuracy
Number of Features
SP-GPMOFS
NS-GPMOFS
SPEA2
NSGAII
Single-Objective-GP


ARC
Classification Accuracy
Number of Features
SP-GPMOFS
NS-GPMOFS
SPEA2
NSGAII
Single-Objective-GP


TOX
Classification Accuracy
Number of Features
SP-GPMOFS
NS-GPMOFS
SPEA2
NSGAII
Single-Objective-GP


HCC
Classification Accuracy
Number of Features
SP-GPMOFS
NS-GPMOFS
SPEA2
NSGAII
Single-Objective-GP

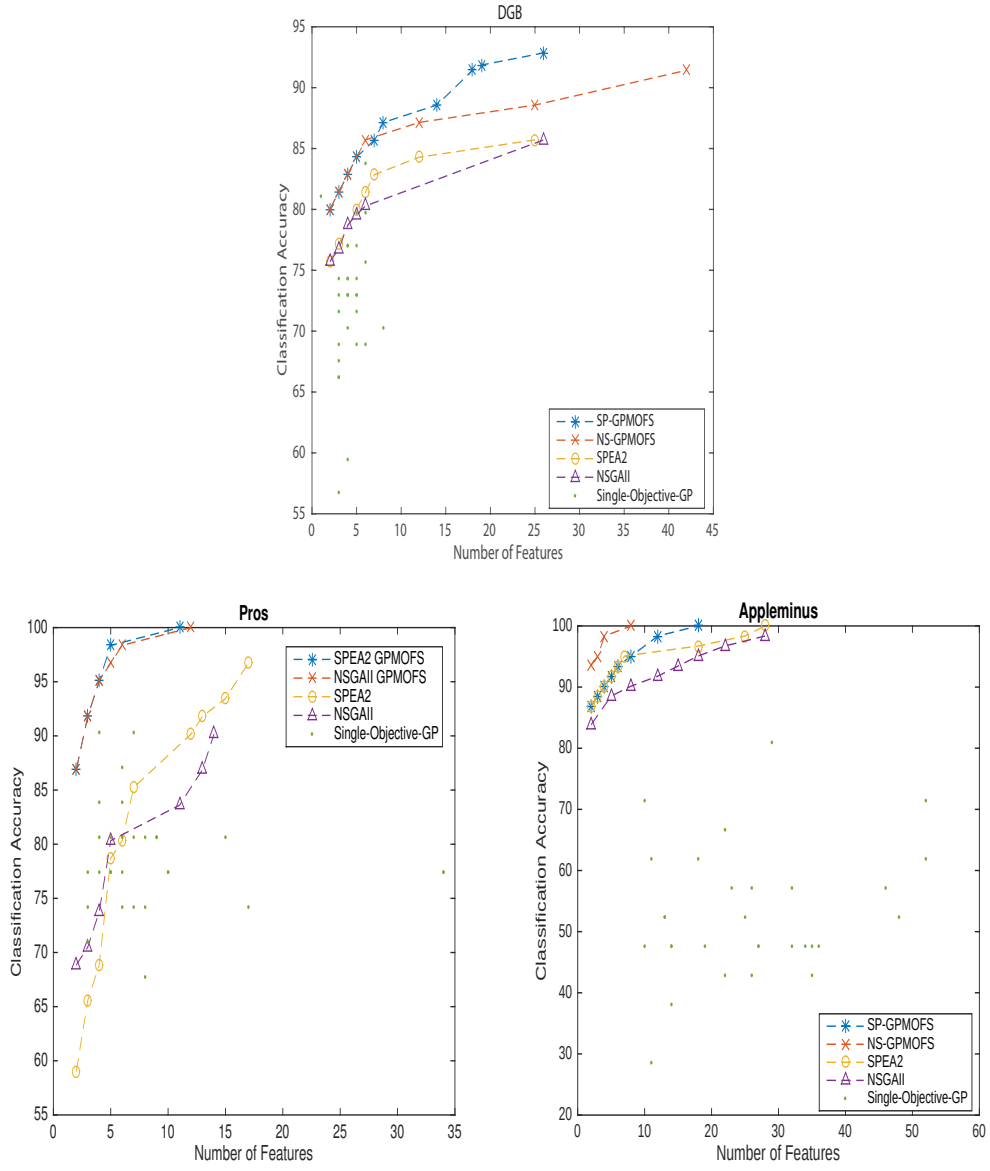
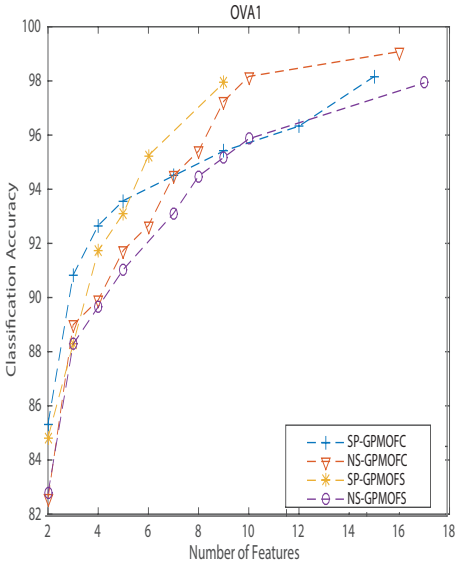
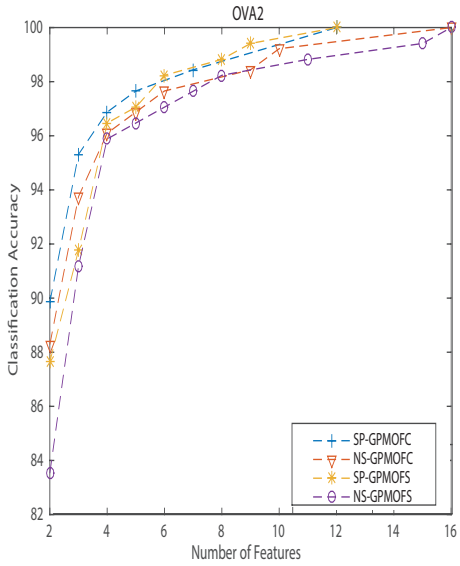
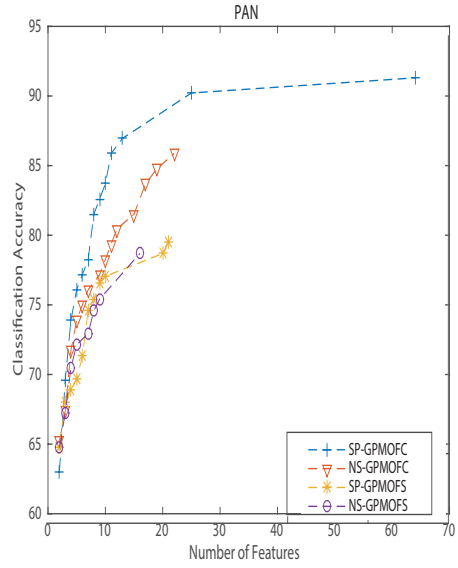
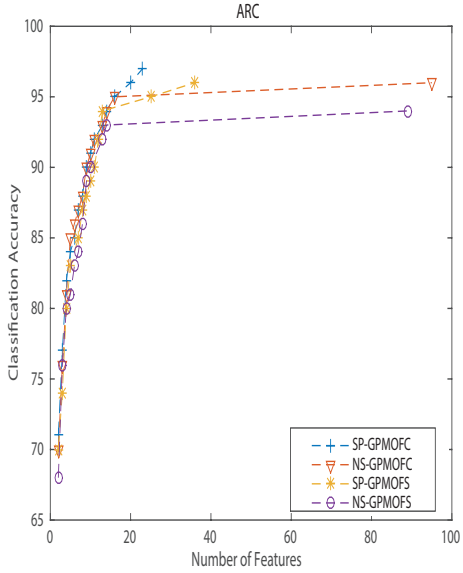


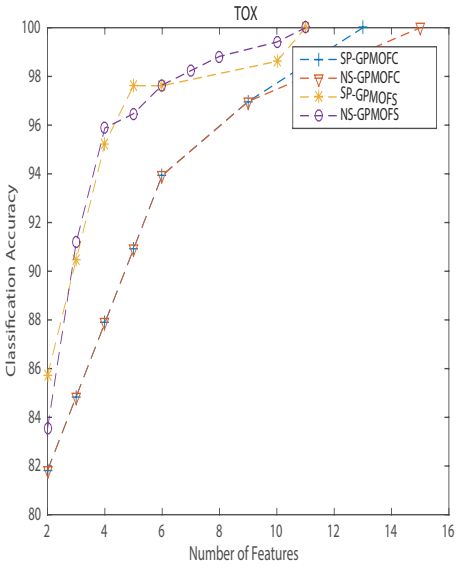
Figure 5.2: Experimental Results for $GPMOFS$

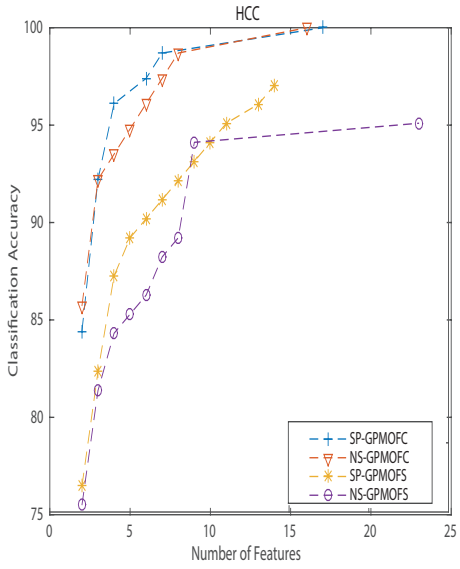

OVA1
Classification Accuracy
Number of Features
SP-GPMOFC
NS-GPMOFC
SP-GPMOFS
NS-GPMOFS


OVA2
Classification Accuracy
Number of Features
SP-GPMOFC
NS-GPMOFC
SP-GPMOFS
NS-GPMOFS


PAN
Classification Accuracy
Number of Features
SP-GPMOFC
NS-GPMOFC
SP-GPMOFS
NS-GPMOFS


ARC
Classification Accuracy
Number of Features
SP-GPMOFC
NS-GPMOFC
SP-GPMOFS
NS-GPMOFS


TOX
Classification Accuracy
Number of Features
SP-GPMOFC
NS-GPMOFC
SP-GPMOFS
NS-GPMOFS


HCC
Classification Accuracy
Number of Features
SP-GPMOFC
NS-GPMOFC
SP-GPMOFS
NS-GPMOFS

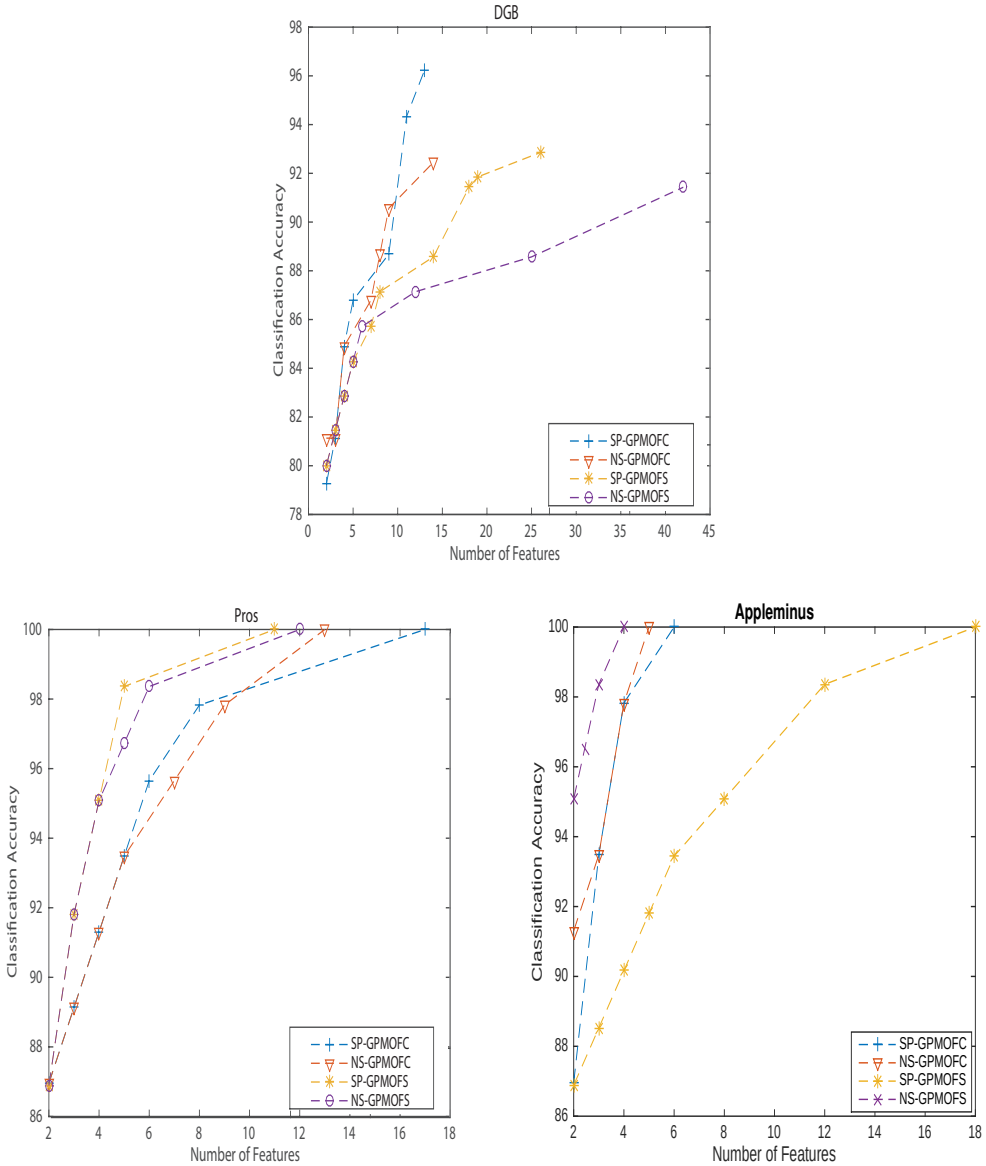


Figure 5.3: Experimental Results for $GPMOFC$

the feature selection algorithms. However, for ARC, using all algorithms the solutions are similar when the number of features is smaller, but SP-$GPMOFC$ outperforms the other algorithms in its last two solutions. In dataset OVA2 (Figure 5.3 (b)), SP-$GPMOFC$ is equivalent to SP-$GPMOFS$ and it outperforms NS-$GPMOFS$.

The results suggest that multi-objective feature construction tends to achieve the balance between reducing the dimensionality and improving the performance better than multi-objective feature selection. This supports our first hypothesis that feature construction can further improve the multi-objective feature manipulation performance through the construction of high-level features that identify the interactions and relations between the original low-level features.

The exceptions to the conclusion mentioned above that multi-objective feature construction can achieve better results than the multi-objective feature selection on these datasets are the TOX and Pros datasets. For Appleminus dataset, NS-$GPMOFS$ has the best set of solutions and the two feature construction methods come next. For these two datasets, $GPMOFS$ is better than $GPMOFC$. $GPMOFC$ tries to reduce the number of constructed features and decreases the dimensionality better than $GPMOFS$ in these two datasets, but this came on the account of the classification performance. However, the gap between the selection and construction is very small. Both selection and construction can achieve 100% accuracy with a number of features of 10-12 for feature selection and 12-16 for feature construction, from over 15,000 features in Pros and 45,000 in TOX.

5.6.3 Comparison of $GPMOFC$ to single objective GP, SPEA2 and NSGAII approaches

Comparing Figure 5.2 and Figure 5.3, $GPMOFC$ is outperforming both SPEA2 and NSGAII in all the cases. If the results of $GPMOFC$ and the

single objective GP are compared, it is also clear the multi-objective construction is better with all the tasks.

This indicates the increased effectiveness of using the high-level features over the selected original features, and gives more credibility to GP as a feature construction approach.

5.6.4 $GPMOFC$ vs single objective GP for feature construction

Considering the method proposed in Chapter 4 in the thesis for feature construction, the number of high-level features is still an issue that needed to be considered. $GPMOFC$ managed to keep the good performance of the algorithm and at the same time significantly reduced the number of constructed features.

In Table 4.3 (Page 102), the average number of features of OVA1 constructed by single objective GP feature construction approach, for instance, is 27.26. $GPMOFC$ managed to keep the maximum number of features of 14 which means reducing the dimensionality by approximately 50%. For TOX, the average number of constructed features using the single objective GP is 37.1 while $GPMOFC$'s maximum number of features is 12. For the rest of the datasets, the same scenario happens, which means that our goal of improving classification through feature construction and reducing dimensionality has been successfully achieved.

5.6.5 Biomarker Detection

We tested the features selected from the Appleminus dataset to check the number of detected predefined biomarkers by each method. Table 5.4 shows the selection status of the biomarker by each of the multi-objective feature selection methods. From the table, it is clear that SP-$GPMOFS$ has outperformed the other three methods and managed to detect the five biomarkers. SPEA2 detected four biomarkers while both NS-$GPMOFS$

and $NSGAII$ detected three out of the five biomarkers. This suggests that $SP\text{-}GPMOFS$ has better performance in terms of biomarker detection as well as higher accuracy solutions with a smaller number of features

Table 5.4: Identified spike-in biomarkers by $SP\text{-}GPMOFS$, $NS\text{-}GPMOFS$, $SPEA2$ and $NSGAII$

m/z value	$SP\text{-}GPMOFS$	$NS\text{-}GPMOFS$	$SPEA2$	$NSGAII$
(5 Biomarkers)				
463.0	✓	✓	✓	✗
447.09	✓	✓	✓	✓
273.03	✓	✓	✓	✓
435.13	✓	✗	✗	✗
227.07	✓	✗	✓	✓

5.7 Chapter Summary

This chapter proposes the first multi-objective biomarker detection approach for MS data. Moreover, the chapter also presents the first multi-objective feature construction algorithm that is applied to MS data. The goal here was to develop a new GP multi-objective feature manipulation algorithm.

In Section 5.2 of the chapter, $GPMOFS$, a GP multi-objective feature selection method is proposed, which manages the trade-off between the classification accuracy and the cardinality of features. According to the results, $GPMOFS$ evolves non-dominated solutions, which has the potential to solve the problem of high dimensionality and a small number of examples in MS data. The method outperforms the single-objective feature selection GP method in terms of both objectives. The method uses the embedded capability of GP to select features with the dominance rank, dominance count and crowding distance to evaluate the solutions. The pro-

posed method also outperforms both SPEA2 and NSGAII multi-objective feature selection approaches using Fisher criterion.

The second part of the chapter presents $GPMOFC$, the first multi-objective feature construction method on MS data. The method is extending the GP multiple feature construction method proposed in Chapter 4 of the thesis. For the construction of multiple high-level features, the features generated from the branches of the evolved GP tree in addition to the root features are used. This generates a number of new high-level features, which has the potential to improve the classification performance. To reduce the dimensionality by generating a smaller number of features, $GPMOFC$ uses ideas from SPEA2 and NSGAII to keep the trade-off between the number of features and the classification performance. The results show that $GPMOFC$ outperformed $GPMOFS$ in almost all the cases, and hence, it was better than SPEA2 and NSGAII approaches and the single objective GP feature selection method. The results also show that the number of constructed features are greatly reduced over the method proposed in Chapter 4, and, therefore, it can be more suitable for dealing with the high dimensional MS data.

Chapter 6

Biomarker Verification

6.1 Introduction

There have been significant efforts to compare proteome of diseased and control samples to discover biomarkers or understand the pathogenesis of diseases. These efforts lead to long lists of biomarkers associated with a wide range of diseases and these biomarkers require validation. However, the experimental validation of these biomarkers is extremely challenging due to the high complexity and heterogeneity of the human samples [9, 111] and the high cost of the process. Since the evaluation of the large number of candidate biomarkers is a critical gap in the biomarker discovery process, an efficient method is needed to select a few candidates who will be passed to experimental validation. Measuring the detectability of peptides can be used to relieve the bottleneck between the biomarker detection and experimental validation through selecting the high responding peptides (referred to as the quantifiable surrogates [23], [5], [6]). This process of measuring the peptide detectability can act as a biomarker verification step.

Peptide detectability, which is the probability that a certain peptide can be observed in a mass spectrometer, is used in this chapter for biomarker verification.

Peptide detection is a classification problem where the task is to classify the flyer vs. the non-flyer peptides in the mass spectrometer. Biomarkers are the flyer peptides that can be detected in the mass spectrometer. The existing approaches to this task did not make use of the GP's advantages, which include flexibility in building models and automatic feature selection [46, 48, 164]. The fact that the power of GP for peptide detection has not been fully used motivates us to investigate its capabilities.

6.1.1 Chapter Goals

The goal of this chapter is to develop a new GP peptide detection system that finds the high responding peptides and selects the important properties necessary for peptide detection. This peptide detection method is used for verification of biomarkers. Moreover, the proposed method should take into account the class imbalance problem of the data. The combination of GP capabilities of feature selection, evolving classification models and also handling the class imbalance problem constitutes the peptide detection system. Specifically, we are interested in investigating the following:

1. The effectiveness of GP to find the important peptide's physicochemical properties while considering the relations with other properties rather than selecting them independently.

2. The requirement to deal with the class imbalance problem while designing the GP peptide detection method.

3. The performance of the proposed method compared to the common benchmark classification and feature selection methods.

4. The comparison of the proposed method to Enhanced Signature Peptide Predictor (ESP-Predictor) [46], which is considered a popular method in this area.

Chapter Organisation:

The rest of the chapter is organised as follows. Section 6.2 describes the proposed GP peptide detectability method. The design of the experiments is explained in Section 6.3. The results and discussions are provided in Section 6.4. Section 6.5 gives the further analysis of the results. The steps for verification of biomarkers using the proposed method are explained in Section 6.6. Section 6.7 concludes and summarises the chapter.

6.2 GP for Measuring Peptide Detectability

6.2.1 Method Overview

The proposed method is designed to perform multiple tasks. The first is feature selection, which is done automatically in GP since it can select the important features in the evolved classification model. This constitutes an advantage of using GP, where it is useful for selecting the relevant features and ignoring the redundant ones. The second task is the prediction of the detected peptides. This is done through building a classification model that predicts whether the peptides can be detected in the mass spectrometer. Mostly the peptide datasets are facing an imbalance problem, and the number of peptides in the observed or the detected class is much smaller than the peptides in the non-observed class. This problem makes the classifier more biased to the class with a large number of instances. To tackle this problem we designed a fitness function, which takes into account the class imbalance problem.

Figure 6.1 shows the process of generating the data. In Figure 6.1, protein samples are digested into peptides and passed to LC-MS/MS to produce the LC-MS/MS spectra. The MS/MS raw data are searched using SEQUEST software (using the suitable database) to perform the peptide and protein identification. The identified proteins are filtered to include only the high confident peptides. The minimum number of the high confi-

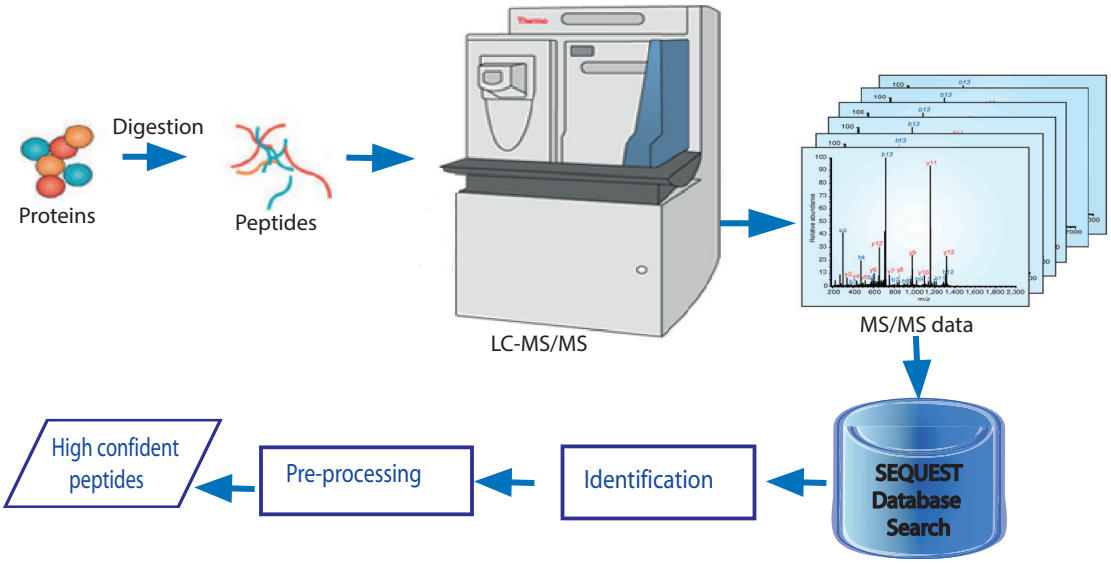


Figure 6.1: Data generation

dent peptides assigned to a protein is one. Figure 6.2 shows an overview of the proposed GP system, named as PEP-GP. The system takes the peptide data as an input, and each peptide is labeled as observed if it is found in at least 50% of the LC-MS/MS experiments in the dataset [38]. Otherwise, it is labeled as a non-observed peptide. The dataset is passed to the feature extractor, which generates the physiochemical properties of the peptides. The features are extracted from the AAindex database, which contain a set of 544 physiochemical properties for each amino acid. The physiochemical properties of each amino acid in the peptide are averaged to generate the feature vector of each peptide instance. Thus, the dataset will construct a matrix of peptides represented by properties along with the class labels of observed or non-observed. The dataset is randomly divided into training and tests where 50% of the data is used for training, while the other 50% is held as unseen data (test set). The imbalance ratio between the two classes (observed and non-observed) is preserved during the process of dividing the data into a training set and a test set. The training set is passed to the GP peptide detection system to build a prediction model for pep-

tide detection. Since some of the features (544 physiochemical properties) might be redundant or not relevant to the classification, feature selection is needed. The proposed GP method performs both tasks of feature selection and classification, and hence, the method outputs the important features which are automatically selected along with the prediction model. The prediction model is examined on the unseen data. Finally, the important features selected are evaluated with other classification algorithms and the prediction probability of the GP is tested.

6.2.2 Feature Vectors

The datasets were obtained in the form of peptides sequences (amino acids) and the class label. Hence, to use those peptides with the selected machine learning techniques, they need to be transformed to a numerical feature vector. The physicochemical properties of the peptides have shown to be related to their detectability [1]. Therefore, for each peptide, 544 properties were calculated to transform the peptide data into numerical feature vectors. The 544 properties were extracted from *AAindex* database [81] and for each peptide sequence the average of the property values of each individual amino acid is calculated over the whole peptide. The physicochemical properties include, for example, mass, alpha-helical (which is the predicted percentage of the secondary structure), hydrophobicity, gas phase basicity, and isoelectric point. Therefore, each peptide is an instance used for training and testing the GP algorithm, where each instance is modeled by 544 feature values along with either observable or non-observable class label.

6.2.3 Fitness Measure for Unbalanced Peptide Data

Since the peptide detection problem is facing a high ratio of imbalance between the observed and non-observed classes, the measure of fitness should take into account the accuracy of the minority and majority classes

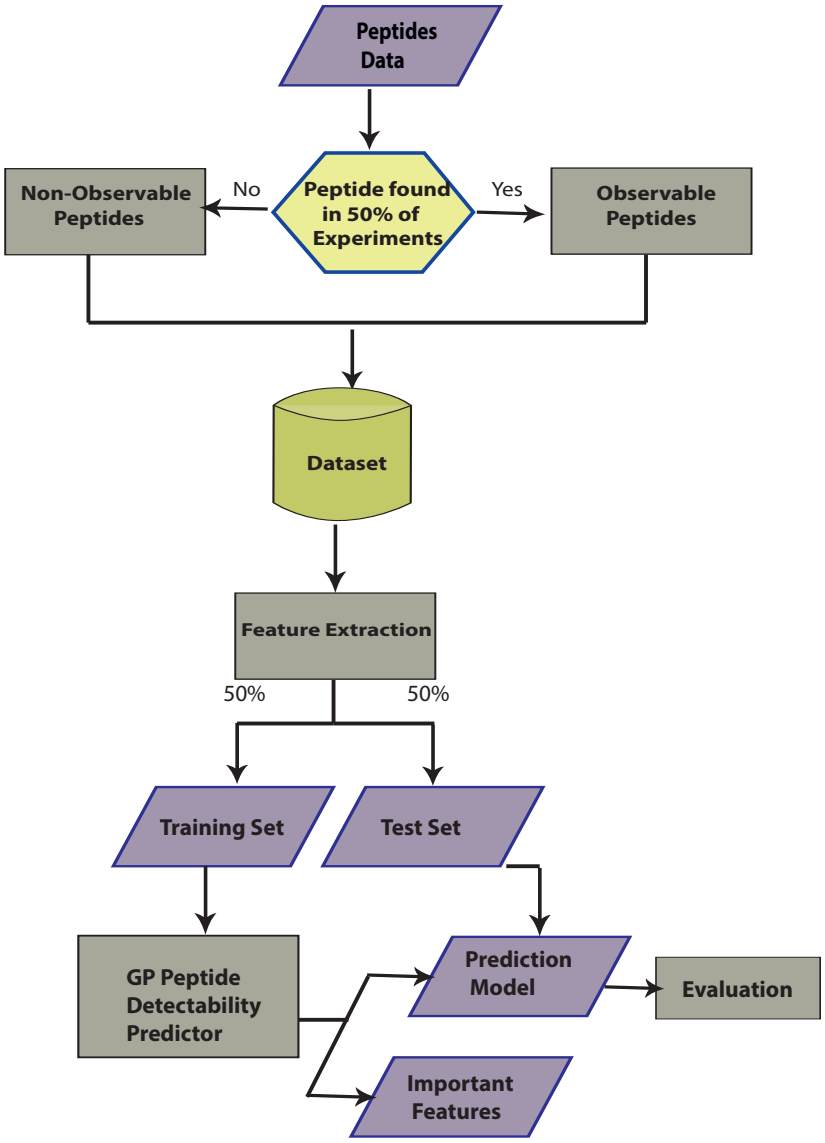


Figure 6.2: The GP peptide detection system

carefully. Hence, we designed a fitness function that tackle this issue by maximising the sensitivity and specificity of classification both with equal weights. The outputs of classification as true or false are the following: true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

The accuracy of majority class (sensitivity) is given by:

$$Sensitivity = MinorityAcc. = \frac{TP}{TP + FN} \tag{6.2.3.1}$$

Additionally, the accuracy of the minority class (Specificity) is given by:

$$Specificity = MajorityAcc. = \frac{TN}{TN + FP} \tag{6.2.3.2}$$

To avoid the bias of the evolved classifiers toward the classification of the majority class, the average classification accuracy is used to evaluate the fitness of the evolved models. The fitness function used is:

$$Fitness\_Fn = 0.5(MajorityAcc. + MinorityAcc.) \tag{6.2.3.3}$$

6.2.4 Evolving Peptide Detectability Models

The measure of peptide detectability is solved as a classification problem, where a classification model is first evolved using a training set, and this model is used to make the prediction. Each genetic program (which is a mathematical expression) outputs a single floating point number for each instance in the dataset that will be classified. Our system performs classification by dividing the output space into two decision intervals. A zero threshold is used to separate the prediction of each class. Therefore, a particular instance will be assigned to the observed class if its classifier's output is negative. Otherwise, it will be classified as a non-observed peptide. The dataset of peptide detection problem is in the form of $D = (v, c)$, where $v = (v_1, v_2, ..., v_m)$ is a vector of m instances which has n peptide's properties and c is the vector of class labels for each corresponding peptide in v. Suppose (φ) is the model GP evolved to separate a class $c^\star$ from rest of classes. We define the binary detectability model $PD_{c^\star}$ as:

$$PD_{c^\star} = \begin{cases} Observed, & \text{if } \varphi_{c^\star}(v_1, v_2, ..., v_m)<0 \\ Non\_observed, & \text{otherwise} \end{cases}$$

Hence, a peptide is predicted as observed if O= $\varphi_{c^\star}(v_1, v_2, ..., v_m)$ (i.e. O is the GP program output) falls in the interval of $c^\star$ which is less than 0, otherwise, it is predicted to be non-observable.

6.2.5 Selecting Important Properties

Feature selection is performed to find the most relevant features of the detection task and to remove the non-informative features. The process of selecting features in our approach is done automatically along with building the prediction model. Hence, the features will not be selected independently, but through discovering the relationships between the features. One of the advantages of GP is that it evolves a tree for classification that has the selected features in its terminal nodes. Therefore, the two processes of feature selection and classification are not separated. This will help in both improving the prediction performance and reducing the computational cost while also considering the interaction between the features.

6.2.6 Summary of the Algorithm

Algorithm 5 explains the steps involved in computing the fitness, which is capable of balancing the accuracy of both the majority and the minority classes. The algorithm requires a dataset (D), which contains the peptides' instances and the class label as observed or non-observed. It also requires the GP program φ which acts as a function to transform the data, and finally the desired class label $c^\star$ for which the classifier should be built. The algorithm outputs a fitness value through which one can tell if an individual is better than another, and hence, the method of selection can perform its task easily. The algorithm starts with using φ for transforming the data

into a single floating point number for each of the m instances in D. The algorithm then counts the number of instances correctly and incorrectly classified and computes the average accuracy of the classification.

Algorithm 6 shows the steps of evolving the GP program that can predict the detectability of peptides and select the important properties. The input to the algorithms is D, which is a dataset of peptide instances with n original features (properties) and a class label for each instance from the vector of the class label c. The output is the best individual ($BestProgram$) and the set of selected features.

F, where the set of selected features is stored, is firstly empty, and at the end it will contain the selected features in the terminal nodes of the $BestProgram$. In each evolved program, the fitness is concerned with separating a specific class label from the other one. Finally, the best individual is the one with the highest fitness value. The $BestProgram$ is updated according to the fitness where the higher, the better. The maximum fitness f_{max} is first initialized to zero and updated to the fitness of the $BestProgram$. The fitness value (f) is calculated by calling the $Evaluate\text{-}Fitness$ function. If its value is greater than f_{max}, then f_{max} is changed to the value of f. The main GP loop is terminated either by reaching the maximum number of generations ($MaxGen$), or f_{max} reaches the optimal value that is 1. Finally, the $BestProgram$ and the set of selected features F are returned as outputs.

6.3 Design of Experiments

This section explains the experiments' set-up that includes the peptide datasets preparation, the GP settings, evaluation process and methods for comparisons.

Algorithm 5 *Evaluate-Fitness (D,φ,c*)*

Require: D, a dataset.
Require: φ, a GP program which acts as a function.
Require, c*, the class label for which the classifier should be built.
Ensure, fit, a value showing the fitness of the GP program (the higher, the better and the maximum is 1).
O[i] $\Leftarrow \varphi$[x$_1$(i),x$_2$(i),..., x$_m$(i)], $\forall_i \in$ {1,2,,..n}
$fit \Leftarrow$ 0;
for i$\Leftarrow$1 to n **do**
  if (O(i)$>$ 0) **then**
    if c[i]= c* **then**
      TP$\Leftarrow$TP+1;
    else
      FN $\Leftarrow$ FN+1;
    end if
  end if
  if (O(i)$\leq$ 0) **then**
    if c[i]$\neq$ c* **then**
      TN$\Leftarrow$TN+1;
    else
      FP$\Leftarrow$ FP+1;
    end if
  end if
end for
$fit \Leftarrow$ 0.5*($\frac{TP}{TP+FN}$+$\frac{TN}{TN+FP}$);
return fit;

Algorithm 6 *Predict-Peptide-Detectability (D)*

Require: D, a dataset that contains a vector of instances of m original features of each instance in the dataset and vector of class labels c.

Ensure, $BestProgram$ & F, a pair containing the best performing GP program which outputs the prediction probability and the vector of selected features of size n.

$F \Leftarrow \{\}$
for c$^\star \in$ c **do**
  P $\Leftarrow$ initialize the population
  $f_{max} \Leftarrow 0$;
  while $CurrGen < MaxGen \wedge f_{max} < 1$ **do**
    for indvidual $\in$ P **do**
      $f \Leftarrow$ Evaluate-Fitness(D,individual,c*)
      if $f > f_{max}$ **then**
        $f_{max} \Leftarrow f$;
        $BestProgram \Leftarrow$ individual;
        $F \Leftarrow F \bigcup \{BestProgram\}$
      end if
    end for
  end while
end for
return ($BestProgram$,F);

6.3.1 Peptide Datasets

Dataset 1 (DS_1)

Sample Collection and Preparation

This dataset was obtained in-house (unpublished data) from the Victoria University of Wellington's proteomic lab. Samples from healthy and treated HepG2 cells were collected, where the treated cells were fed with a high concentration of fatty acids and had lipids accumulated in the cell. Proteins from both cells were extracted, reduced and alkylated according to Mast et al. [113]. Afterwards, the samples were digested using trypsin.

Dionex UltiMateTM 3000 RSLCnano system (Thermo Scientific, USA) coupled with Linear Trap Quadrupole (LTQ) and coupled with an Orbitrap XL mass spectrometer (Thermo Fisher Scientific, USA) is used. The peptides were separated on a 300 min LC gradient constructed from 0.1% formic acid in 80% acetonitrile acid (solvent B). Fourier transform mass spectrometry (FTMS) in the Orbitrap at a resolution of 30,000 was used to collect the precursor ions with m/z range of 200-1800. Data-dependent ms/ms of the top 6 intense ions was dynamically selected for collision-induced dissociation (CID) fragmentation and detection. Each sample from the healthy or treated class was run five times, and hence, each class has one sample with five replicates.

Peptide and protein identification

The spectra were searched against the human proteome database (915565 protein sequences) obtained from UniProt Knowledge-base using the SEQUEST algorithm in Proteome Discoverer (v 1.2.0.208, Thermo Fisher Scientific). A mass range of 350-5000 Da was searched with a signal to noise threshold of 1.5 and allowing 2 missed trypsin cleavages [113].

The spectra were searched with a fragment ion mass tolerance of 0.80 Da and a parent ion tolerance of 10.0 ppm. Trypsin was specified as the

digestion enzyme with two maximum missed cleavages. Other protein modification parameters are: carbamidomethylation of cysteine was set as a fixed modification (+57.021). Dynamic side chain modifications were: oxidation +15.995 Da at M; carbamylation +43.066 Da at K; acetylation +42.011 Da at K; deamidation +0.984 Da atnd N, and Q and R; phosphorylation +79.966 Da at R, S, and T and Y. The N-terminal modification was carbamylation +43.066 Da. The files were searched against a decoy database, with a strict false discovery rate (FDR) of 1% and relaxed FDR of 5%. The search resulted in 706 proteins with 4985 peptides, where 3629 are classified as non-observed class and 1356 are labeled as observed. This dataset is notated as DS_1.

Dataset 2 (DS_2) and Dataset 3 (DS_3)

Two other tryptic peptide datasets originating from yeast are used to test the performance of the proposed GP method. The datasets were obtained from [38] and [146], and analysed using LC-ESI-MS (Liquid Chromatography-Electro Spray Ionization-Mass Spectrometry).

Dataset 2 was produced from 24 yeast experiments originally downloaded from *PeptideAtlas* [30]. The total number of proteins is 2733. The peptides' length (number of amino acids) ranges from 6 to 42 residues with 0-2 missed cleavage. Each peptide was assigned an observed class label if it was detected in the 24 experiments. Otherwise, it was assigned a non-observed class label. The total number of peptides in this dataset is 21515 in which 2121 peptides are in the observed class and 19394 are in the non-observed class. We annotate this dataset as DS_2. More details about the datasets acquisition and preprocessing can be found in [38]. The third dataset (Dataset 3) used in our experiments was generated from 15 proteins. The proteins were searched against the NCBInr database [139] using Mascot server [87] (Matrix science) to confirm identity and elution time. Extracted ion chromatograms were generated for the peptides that did not yield tandem MS data. Each peptide contains at least five amino

acids and generated by either 0 or 1 missed cleavage. The class label as observed or non-observed was set by counting the number of its occurrences in the dataset, where if the peptide is found in 50% of the experiments it is assigned as observed. This dataset is annotated as DS_3 and contains 809 peptides, where 657 belong to the non-observed class while 152 belong to the observed class. The three datasets used in this study are summarised in Table 6.1

Table 6.1: Datasets

Dataset	# peptides in the observable class	# peptides in the non-observable class
DS_1	1356	3629
DS_2	2121	19394
DS_3	152	657

6.3.2 Program Representation

Th experiments were conducted using the tree-based GP, where each program is a hierarchy of nodes. The program forms a mathematical expression where each of the tree's nodes produces a single floating point value. The root node of the tree generates the final output of the evolved program, which is used to decide the class label. The program takes variable terminal nodes and constant terminal nodes as inputs to the program. The variable terminal nodes take the values from the original features while the constant terminal nodes take their values randomly from the range of [-1,1]. The function set consists of the four standard arithmetic operators and an absolute operator ($+$, $-$, $\times$, $\%$, Abs). The division operator is protected which returns zero for division by zero. The absolute operator returns the absolute value of its input. Table 6.2 shows the function set operators with their inputs and outputs.

Table 6.2: Function Set

Function	Arguments	Description
$+$	Double, Double	Performs addition of the two arguments
$-$	Double, Double	Performs subtraction of the two arguments
$\times$	Double, Double	Performs multiplication of the two arguments
$\%$	Double, Double	Performs protected division of the two arguments
Abs	Double	Returns the absolute of the argument

6.3.3 GP Parameters

The method used to generate the initial population is the ramped-half-and-half method [136]. The population contains 512 individuals. The crossover, mutation and reproduction probabilities are 80%, 19% and 1% , respectively. The tree can grow up to a depth of 10. The selection is performed using the tournament method with a size of 4. The number of generations is 100. The evolution stops if the maximum number of generations is reached, or the best fitness (100%) is achieved.

The evolutionary process is repeated for 30 independent runs where at each run a different random seed is used. For running GP, the Java-based Evolutionary Computation research system (ECJ) [108] package was used. Table 6.3 shows the evolutionary running parameter values used.

Table 6.3: GP evolutionary parameters

Initialization method	Ramped Half-and-Half
Max. Tree Depth	10
Max.# of Generations	100
Mutation probability	19%
Crossover probability	80%
Reproduction probability	1%
Population size	512
Selection method	Tournament
Tournament Size	4

6.3.4 Training and Testing Process

The evaluation is performed on the test set, which was not used during training. As there was no separate test set available in the three datasets. Each dataset is randomly divided to 50% for training and 50% for testing while preserving the original ratio of imbalance between the observed and non-observed classes. The weighted average accuracy, as well as the sensitivity and specificity, are used as evaluation measures to avoid the class imbalance learning bias. The specificity and sensitivity are used in order to show the accuracy of each class (majority and minority) independently.

6.3.5 Methods for Comparison

For the methods of comparisons, we used GP with standard classification accuracy as a fitness function using the original features. This is used as a baseline for comparison to the proposed method. The baseline GP method is shown by $baseline\text{-}GP$. The GP settings and parameters of the baseline method are set to be the same as the proposed method. Moreover, different benchmark feature selection and classification algorithms are used for comparison. The same training and test sets are used with all the other

methods. Waikato Environment for Knowledge Analysis (WEKA) package [67] is used for running the benchmark of comparison.

In addition to $baseline\text{-}GP$, the following methods are also used for comparison:

Classifier learning Methods

A range of various classifier learning algorithms are used for comparison that are shown as follows:

1. Support Vector Machines (SVMs): SVMs form some hyperplanes and classify the instances according to the side of the hyperplane to which the instance belongs [180].

2. Decision Tree (J48): J48 is the Java implementation of the C4.5 decision tree inducer. J48 classifies instances through sorting them in a tree which is composed of a hierarchy of nodes. The root node first tests the value of the feature and then moves to the child nodes until the label node is reached [68].

3. Voted Perceptron (VP): VP is based on the perceptron algorithm and uses kernel functions to build hyperplanes as decision boundaries [43].

4. Conjunctive Rule (CR): CR builds a single conjunctive rule to predict the class labels. It uses the ”AND" logical operator to determine the correlation of features and classes [180].

5. OneR: OneR performs classification like a 1-level decision tree [75].

6. CART: CART generates partial decision trees several times to infer rules [41].

The following feature selection methods are also used for comparison:

Feature Selection Methods

We choose four common feature selection methods to compare with the GP's selected features with that selected by benchmark methods. The methods are described as the following:

1. Information Gain (IG): determines the amount of information gained about a class when a certain feature exists or not [147]. It is defined as follows:

$$IG(\grave{f}, c_i) = \sum_{c \in \{c_1, c_2\}} \sum_{\grave{f} \in \{f, \bar{f}\}} P(\grave{f}, c) \frac{\log P(\grave{f}, c)}{P(\grave{f}) P(c)} \tag{6.3.5.1}$$

   where f and $\bar{f}$ denotes the presence and the absence of a feature and the healthy and diseased classes are denoted by c_1 and c_2. The probability of the occurrence features $\grave{f}$ in a specific class is represented by $P(\grave{f}, C)$. $P(\grave{f})$ and $P(c)$ represent the probability of the observation of a feature and class, respectively.

2. Information Gain Ratio (IGR) feature evaluation: the attributes are evaluated by measuring the gain ratio with respect to the class [40]. The gain ratio is the ratio between the total entropy of the attribute and the intrinsic value.

3. Relief-F (REL-F): REL-F ranks the features according to their capability to distinguish instances of different classes. This is done by searching the k nearest neighbors of instances between the same and different classes with respect to the attribute value [161].

4. Chi Square (χ^2) feature evaluation: in statistical analysis methods, χ^2 test is used to measure the independence of two events. χ^2 measures the association between the features and classes. A score is given to each feature according to its χ^2 statistics with respect to the class [40].

6.4 Results and Discussions

In this section, the results of the proposed method (PEP-GP) as a classifier and as a feature selection method are given.

6.4.1 Classification Results

Table 6.4 presents the results of the proposed method (PEP-GP) on the unseen data (test set) compared with the other classifiers. In Table 6.4, the first column indicates the dataset while the second column gives the method used for peptide detection. The first six classifiers used are deterministic methods and hence they only have a single solution, while the two GP methods run 30 times. Hence, the mean, best, and standard deviation ($\mu \pm \sigma$) are presented. The average of the majority and minority accuracies ($Avg.Acc.$) are given in the third column. The overall accuracy of classification, and accuracies of each class (majority and minority classes) are presented.

The statistical information is provided in the last two columns where $d_{avg.acc.}$ gives the difference in the average accuracies between the proposed method and the methods of comparison while the p-$value$ gives the p-$value$ of the z-test. The z-test is performed with three different confidence intervals which are 95%, 99%, 99.9%. Each of the $\star$ in the last column indicates that the PEP-GP is significantly better using the different confidence intervals.

According to Table 6.4, PEP-GP is highly effective in improving the average accuracy ($Avg.Acc$) of all the datasets. In terms of $Avg.Acc$, it can be noticed that PEP-GP has outperformed the seven classifiers for all the three datasets. Testing the significance of the results with three confidence intervals shows that PEP-GP is significantly better than that of the benchmark classifiers.

Although the overall accuracy (Overall Acc.) of the benchmark classifiers are more than PEP-GP, as discussed before the overall accuracy is

very sensitive to the imbalance of the data. This means that the overall accuracy is only a good-looking result while the accuracy of majority class is much larger than the minority class accuracy. It is clear from majority and minority classes accuracies of DS_1 , DS_2 and DS_3 (where the imbalance ratio is high) that $PEP\text{-}GP$ kept the balance between the two classes and is not biased by the majority class so much. For example, in DS_1, the minority's class accuracy of the proposed method is better than the other classifiers by 48.7-61%. For DS_2, the performance of the minority class of $PEP\text{-}GP$ is improved over 35.5%-71% than the other six classifiers.

In case of DS_3, the mean and the $best$ of 30 runs of $Avg.Acc.$ (μ) are better than all other classifiers. It can be noticed that the baseline-GP has the same attitude with unbalanced data which is being biased to the majority class. This suggests that the proposed method is more suitable for measuring the peptide detectability than the existing benchmark classification methods.

6.4.2 Feature Selection Results

Table 6.5 shows the average and overall classification accuracy of using the features selected by $PEP\text{-}GP$. In Table 6.5, the first column refers to the dataset, the third column shows the method used for feature selection. The average and overall accuracies ($Avg.Acc$ and $OverallAcc.$) are shown when the features selected by each method are used with each classifier. The last column shows the number of features used. The average number of features selected by $PEP\text{-}GP$ is used as a reference for selecting the number of top ranked features used for each method.

According to Table 6.5, the proposed method performs either the same or better than IG, IGR, χ^2 and RLF-F. For DS_1, all classifiers with all the feature selection methods have a similar performance. For DS_2, the performance of SVMs, VP, and CR are similar when using the different feature selection methods. Moreover, the best of $PEP\text{-}GP$ as a feature selection

Table 6.4: Classification results of $PEP\text{-}GP$ and other classifiers.

dataset	Method	Avg. Acc.	Overall Acc.	Majority Acc	Minority Acc.	Statistical Test	
						$d_{avg.acc}$	p-value
DS_1	SVMs	50.00	72.82	100	0.00	8.00	$\prec$0.0001$\star\star\star$
	J48	50.55	70.13	93.40	7.7	8.00	$\prec$0.0001$\star\star\star$
	VP	50.00	72.82	100.0	0.0	8.00	$\prec$0.0001$\star\star\star$
	CR	50.00	72.82	100.0	0.0	8.00	$\prec$0.0001$\star\star\star$
	OneR	51.25	69.01	90.2	12.30	6.75	$\prec$0.0001$\star\star\star$
	CART	50.05	72.17	98.6	1.50	7.95	$\prec$0.0001$\star\star\star$
	Baseline-GP($\mu(best) \pm \sigma$)	50.00(50.01)$\pm$0.001	72.67(73.00)$\pm$0.001	100.00(100.00)$\pm$0.0	0.0(0.0)$\pm$0.0	7.99	$\prec$0.0001$\star\star\star$
	PEP-GP($\mu(best) \pm \sigma$)	58.00(66.41)$\pm$0.05	56.00(66.67)$\pm$0.04	55.00(74.13)$\pm$0.14	61.00(80.88)$\pm$0.08		
DS_2	SVMs	50.00	90.14	100.00	0.00	21.00	$\prec$0.0001$\star\star\star$
	J48	64.55	87.91	93.60	35.50	6.45	$\prec$0.0001$\star\star\star$
	VP	50.00	90.14	100.00	0.00	21.00	$\prec$0.0001$\star\star\star$
	CR	50.00	90.14	100.00	0.00	21.00	$\prec$0.0001$\star\star\star$
	OneR	55.75	90.04	98.5	13	15.25	$\prec$0.0001$\star\star\star$
	CART	57.95	89.70	97.5	18.4	13.05	$\prec$0.0001$\star\star\star$
	Baseline-GP($\mu(best) \pm \sigma$)	50.15(51.5)$\pm$0.01	90.00(90.17)$\pm$0.001	100.00(100.00)$\pm$0.0	0.0(0.0)$\pm$0.0	20.75	$\prec$0.0001$\star\star\star$
	PEP-GP($\mu(best) \pm \sigma$)	71.00(76.09)$\pm$0.04	71.00(78.3)$\pm$0.05	70.6(82.4)$\pm$0.07	71.00(75.15)$\pm$0.02		
DS_3	SVMs	50.00	81.38	100.00	0.00	11.00	$\prec$0.0001$\star\star\star$
	J48	53.65	71.46	82.00	25.30	7.35	$\prec$0.0001$\star\star\star$
	VP	50.00	81.38	100.00	0.00	11.00	$\prec$0.0001$\star\star\star$
	CR	50.00	81.38	100.00	0.00	11.00	$\prec$0.0001$\star\star\star$
	OneR	48.35	77.17	93.90	0.04	12.65	$\prec$0.0001$\star\star\star$
	CART	48.25	68.48	80.5	0.16	12.75	$\prec$0.0001$\star\star\star$
	Baseline-GP($\mu(best) \pm \sigma$)	50.27((52.67)$\pm$0.012	81.00(81.63)$\pm$0.01	99.00(100.00)$\pm$0.01	1.55(5.33)$\pm$0.015	10.73	$\prec$0.0001$\star\star\star$
	PEP-GP($\mu(best) \pm \sigma$)	61.00(85.71)$\pm$0.14	54.00(69.70)$\pm$0.18	66.00(90.47)$\pm$0.13	56.42(100.00)$\pm$0.31		

Table 6.5: Feature Selection Results

dataset	Method	SVMs		J48		VP		CR		OneR		CART		# of Features
		Avg Acc.	Overall Acc.	Avg Acc.	Overall Acc.	Avg Acc.	Overall Acc.	Avg Acc.	Overall Acc.	Avg Acc.	Overall Acc.	Avg Acc.	Overall Acc.	
DS_1	IG	50.00	72.82	50.00	72.82	50.00	72.82	50.00	72.82	53.5	69.34	50.00	72.82	7
	IGR	50.00	72.82	50.00	72.82	50.00	72.82	50.00	72.82	52.5	68.25	50.00	72.82	
	χ^2	50.00	72.82	50.00	72.82	50.00	72.82	50.00	72.82	53.5	69.34	50.00	72.82	
	RLF-F	50.00	72.82	50.00	72.82	50.00	72.82	50.00	72.82	52.45	68.33	50.00	72.82	
	PEP-GP (best)	(50.00)	(72.82)	(50.00)	(72.82)	(50.05)	(72.86)	(50.00)	(72.82)	(53.72)	(71.29)	(50.00)	(72.82)	
	($\mu \pm \sigma$)	(50.00)±0.0	(72.82)±0.0	49.95±0.01	72.63±0.35	50.00±0.001	72.78±0.09	50.00±0.0	72.82±0.0	50.65±0.01	69.25±0.89	50.00±0.0	72.77±0.72	
DS_2	IG	50.00	90.14	57.30	90.12	50.00	90.14	50.00	90.14	52.00	89.65	56.20	90.03	8
	IGR	50.00	90.14	57.30	90.03	50.00	90.14	50.00	90.14	51.95	89.61	56.20	90.04	
	χ^2	50.00	92.22	59.25	90.13	50.00	92.22	50.00	92.22	54.95	90.14	62.55	90.12	
	RLF-F	50.00	92.22	54.80	90.14	50.00	92.22	50.00	92.22	51.6	89.65	51.9	90.35	
	PEP-GP (best)	(50.00)	(90.15)	(60.51)	(90.59)	(52.97)	(90.17)	(50.00)	(90.15)	(52.3)	(90.23)	(63.77)	90.57)	
	($\mu \pm \sigma$)	50.00±0.0	90.15±0.0	53.12±0.04	89.95±0.4	50.33±0.01	90.06±0.2	50.00±0.0	90.15±0.0	51.41±0.01	89.81±0.25	52.21±0.04	90.03±0.43	
DS_3	IG	49.85	80.44	50.00	80.69	50.00	80.69	50.00	80.69	49.45	78.21	50.00	80.69	15
	IGR	49.85	80.44	50.00	80.69	50.00	80.69	50.00	80.69	49.45	78.21	50.00	80.69	
	χ^2	49.85	80.44	50.00	80.69	50.00	80.69	50.00	80.69	49.45	78.21	50.00	80.69	
	RLF-F	49.85	80.44	50.00	80.69	49.85	80.44	50.00	80.69	49.45	78.21	50.00	80.69	
	PEP-GP (best)	(50.00)	(81.39)	(58.05)	(81.11)	(50.70)	(81.39)	(50.00)	(81.39)	(51.71)	(80.89)	(52.00)	(81.38)	
	($\mu \pm \sigma$)	50.00±0.0	81.38±7.22e^{-14}	50.00±0.25	81.39±1.28	50.20±0.004	81.38±0.045	50.00±0.0	81.38±7.22e^{-14}	50.48±0.035	78.37±1.69	49.95±0.008	81.16±0.81	

method outperformed the other feature selection methods using J48, OneR and CART. In DS_3, the proposed method is better than the four benchmark feature selection methods when used with SVMs, J48 and OneR. The feature selected by PEP-GP improved the performance of the three classifiers in terms of both average accuracy and overall accuracy. The performance of the CR and CART is near similar for all the feature selection methods. Using VP, the average accuracy of using PEP-GP outperformed the four methods, however, the overall accuracy of using these methods is slightly better. It can be noticed that all the feature selection methods are not successful in improving the average accuracy. The classifiers' performances are biased towards the majority class. The possible reason for this is that the problem of imbalance of the data depends mainly on the classifier and not on the feature selection method. Another possible reason is the separation of the two processes of feature selection and classification is not useful for improving the performance. This suggests that the proposed method is more suitable for solving this problem as it selects important features and performs classification, while also taking into account the imbalance problem.

GP-selected features vs. all features

Comparing Table 6.4 with Table 6.5, PEP-GP managed to select a smaller number of features which either improved or preserved the same performance of the classifiers using the original number of features. For example, in DS_1 using only 7 features, SVMs features achieved similar average and overall accuracies as the original features. In the same dataset, the overall accuracy of J48 using the features selected by PEP-GP tends to be better. The same scenario happened in DS_2. For example, VP and CART using a smaller number features have better average accuracies. In most cases, the smaller number of features achieved similar performances with the whole set features when used with the other four classifiers. Finally for DS_3, SVMs performed similarly with a smaller number of features, while

the best of J48 in terms of the average accuracy is better than all the original features. However, the mean of overall accuracy of J48 using the PEP-GP features is far better than the whole set of features. For the other classifiers, the performance is never worse than using all the original features.

6.5 Further Discussions

In this section, resampling of the datasets is performed to balance the number of examples between the two classes. The different classifiers are applied to the datasets after resampling for further comparison of PEP-GP to the classifiers. A comparison of the proposed method to ESP-predictor [46] (a state-of-the-art method for peptide detection) is also discussed in this section. Furthermore, an analysis of the complexity of PEP-GP's evolved models is performed. Finally, this section includes a discussion of the important properties which are selected by PEP-GP compared to ESP-predictor.

6.5.1 Comparison after Data Resampling

A resampling of the three datasets is performed on the training set. The resampled data is used with the different classifiers to further compare PEP-GP performance (without resampling) to the performance of the classifiers after resampling of the training data. The filter resampling algorithm Synthetic Minority Oversampling Technique (SMOTE) implemented in WEKA [67] is used where the percentage of instances for SMOTE to create in the minority class is 300% of the original instances.

Table 6.6 shows the results of the test set using PEP-GP and the other classifiers. As shown in Table 6.6, PEP-GP succeeded in generally outperforming the six classifiers after resampling of the datasets in terms of average accuracy. Although the resampling helped in enhancing the performance of some of the classifiers, the average accuracy of PEP-GP with-

Table 6.6: Classification results of $PEP\text{-}GP$ and other classifiers after resampling of the data.

dataset	Method	Avg. Acc.	Overall Acc.	Majority Acc	Minority Acc.	Statistical Test	
						$d_{avg.acc}$	p-value
DS_1	SVMs	56.75	82.69	92.00	21.5	1.25	$\prec$0.001$\star\star$
	J48	51.70	81.84	92.26	10.80	6.30	$\prec$0.0001$\star\star\star$
	VP	53.65	70.74	76.90	30.40	4.35	$\prec$0.0001$\star\star\star$
	CR	62.65	69.58	72.00	53.30	-4.65	
	OneR	46.70	79.89	91.70	1.7	11.30	$\prec$0.0001$\star\star\star$
	CART	51.10	83.10	94.5	7.7	6.90	$\prec$0.0001$\star\star\star$
	PEP-GP($\mu \pm \sigma(best)$)	58.00(66.41)$\pm$0.05	56.00(66.67)$\pm$0.04	55.00(74.13)$\pm$0.14	61.00(80.88)$\pm$0.08		
DS_2	SVMs	64.42	87.38	93.10	35.4	6.57	$\prec$0.0001$\star\star\star$
	J48	61.80	85.56	91.10	35.20	9.20	$\prec$0.0001$\star\star\star$
	VP	65.90	86.72	91.80	40.00	5.10	$\prec$0.0001$\star\star\star$
	CR	63.65	69.44	70.90	56.40	7.35	$\prec$0.0001$\star\star\star$
	OneR	52.10	87.72	96.5	7.7	18.90	$\prec$0.0001$\star\star\star$
	CART	60.70	88.32	95.10	26.30	10.30	$\prec$0.0001$\star\star\star$
	PEP-GP($\mu(best) \pm \sigma$)	71.00(76.09)$\pm$0.04	71.00(78.3)$\pm$0.05	70.6(82.4)$\pm$0.07	71.00(75.15)$\pm$0.02		
DS_3	SVMs	50.40	71.96	84.80	1.60	10.60	$\prec$0.0001$\star\star\star$
	J48	53.05	70.47	80.80	25.30	7.95	$\prec$0.0001$\star\star\star$
	VP	54.55	68.73	77.10	32.00	6.45	$\prec$0.0001$\star\star\star$
	CR	49.35	75.93	91.80	6.70	5.95	$\prec$0.0001$\star\star\star$
	OneR	55.05	71.215	91.80	6.7	11.65	$\prec$0.0001$\star\star\star$
	CART	50.60	66.65	75.90	25.30	10.40	$\prec$0.0001$\star\star\star$
	PEP-GP($\mu(best) \pm \sigma$)	61.00(85.71)$\pm$0.14	54.00(69.70)$\pm$0.18	66.00(90.47)$\pm$0.13	56.42(100.00)$\pm$0.31		

out resampling is better than that of these classifiers. For example, SVMs's Avg.Acc. had been improved by 6.75%, 14.42% and 0.40% after resampling, however, $PEP\text{-}GP$ is still outperforming SVMs by 1.25%, 6.6% and 10.6% in DS_1, DS_2 and DS_3, respectively. The statistical tests also show that $PEP\text{-}GP$ is significantly better than the other classifiers with resampling in the three datasets. The only exception is CR in DS_1, where its average accuracy is better than $PEP\text{-}GP$. The results suggest that the proposed method has the potential to perform the task of peptide detection effectively without resampling of the data. The minority class accuracy using $PEP\text{-}GP$ is much higher than using other classifiers which means that $PEP\text{-}GP$ is more suitable to perform peptide biomarker verification (biomarkers are the flyer peptides).

6.5.2 Comparison to ESP-Predictor

In this section the performance of $PEP\text{-}GP$ is compared to a benchmark method, Enhanced Signature Peptide Predictor (ESP-Predictor) [46], which is a leading tool for this problem. ESP-Predictor's input is a list of peptides and outputs a probability of detection for each peptide. In ESP-Predictor, the random forest classifier is used to calculate the probability of detection of each peptide. Similar to our approach, ESP-Predictor generates the feature set from AAindex. The same unseen data used to test the proposed method is used with ESP-Predictor. Figure 6.3 shows a comparison between the performances of $PEP\text{-}GP$ and ESP-Predictor. The first bar graph shows the $Avg.Acc.$, while the second one shows the Overall Acc. of the two methods. It can be noticed from the first bar chart that the proposed GP method is far better than ESP-Predictor in terms of Avg. Acc. The performance of ESP-Predictor is biased towards the non-observed class (majority class). This suggests that $PEP\text{-}GP$ is better than ESP-Predictor for this task where the observed class (class where the biomarkers belong) is usually the minority class.

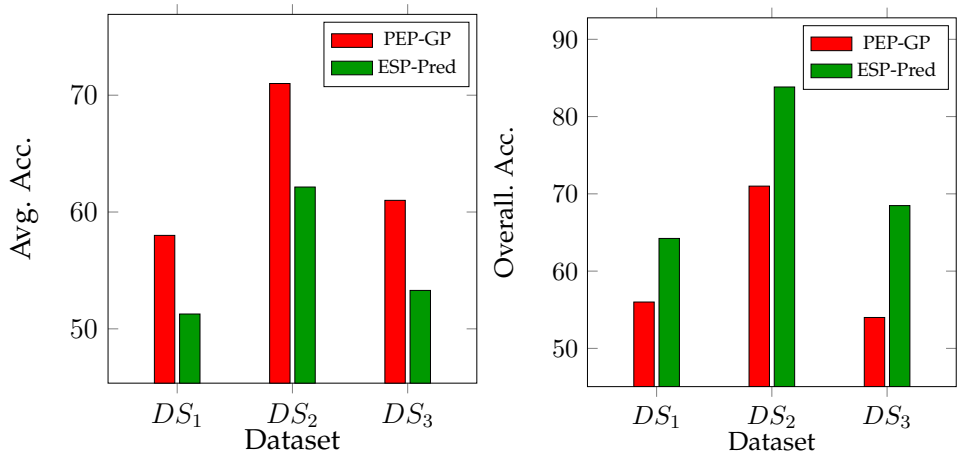
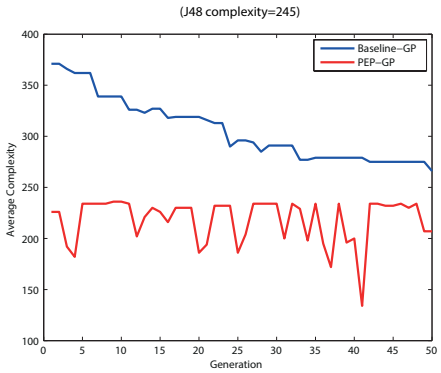


Figure 6.3: Comparison of $PEP\text{-}GP$ with ESP-Predictor

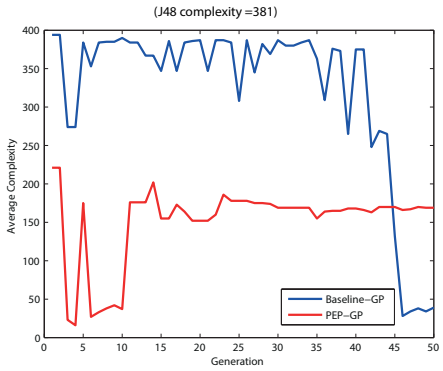
Complexity of the Evolved Models

We also compared the complexity (# of nodes) of $PEP\text{-}GP$ to the complexity (the average number of nodes) of baseline-GP and J48. Figure 6.4 shows the average complexity per generation for the 30 runs of both $PEP\text{-}GP$ and the baseline-GP for each of the datasets. The red line indicates $PEP\text{-}GP$, while the blue line indicates the $baseline\text{-}GP$. The number between the brackets is the complexity of J48. The average number of nodes of the 30 runs for each generation is calculated for each of $PEP\text{-}GP$ and $baseline\text{-}GP$. It can be noticed that $PEP\text{-}GP$ has reduced the complexity for DS_1. For DS_2, in the first 40 generations, the complexity of the proposed method is less than the $baseline\text{-}GP$. However, in the last 10 generations $baseline\text{-}GP$ drops the complexity. DS_2 imbalance ratio is higher than the other two datasets, and hence, the $baseline\text{-}GP$ reduced the complexity as it tends to create small models completely biased towards the majority class.

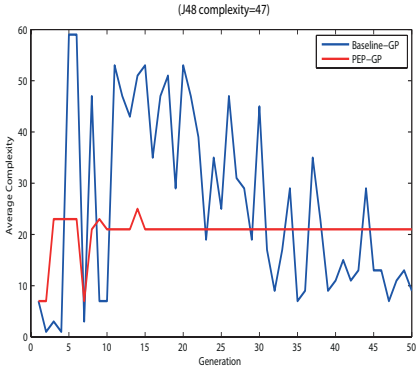
Finally for DS_3, the same scenario occurred where the complexity of $PEP\text{-}GP$ is smaller than $baseline\text{-}GP$ in the first generations and keeps smooth till the end, while $baseline\text{-}GP$ has more divergent complexity.



(a) Average complexity per generation for DS_1



(b) Average complexity per generation for DS_2



(c) Average complexity per generation for DS_3

Figure 6.4: Average complexity per generation of PEP-GP vs. Baseline-GP

This can be explained by the high ratio of imbalance of this dataset which makes the complexity of $baseline$-GP degrades easily as it is generating models which are more biased to the majority class. Another possible explanation of the stable complexity of PEP-GP for DS_3 is that the method is trapped to a local optima, which means that the method needs more divergence to be introduced between the individuals. Comparing PEP-GP and J48, it can be noticed that the maximum complexity of PEP-GP is much smaller than J48 on the three datasets.

6.5.3 Important Properties for Detectability

Another ranking is performed on the features selected by the proposed method where the more frequently selected by the GP tree, the higher their rank. Moreover, only features selected in more than 50% (15 runs) of the runs are considered important. The properties selected by PEP-GP for the three datasets, which are important for peptide detectability, are shown in Table 6.7.

In Table 6.7, the first column gives the description of the physicochemical property selected, the second column indicates the percentage of overlap between the 30 GP runs and the third column indicates if the property has been identified by the domain experts or not. We examined the important properties from [46] to test the common properties between our top properties and the domain experts. In [46], the authors ranked 35 properties as the most important properties for detection, while our method identified 14 properties. When comparing the proposed method and the method in [46] (ESP-Predictor), PEP-GP outperformed ESP-Predictor as shown in section 6.8.2. This indicates that the 14 properties are more important than the 35 properties ranked, and the rest of 35 features are redundant. As shown in Table 6.7, 10 properties are common between our method and the domain experts which suggests that PEP-GP not only identified the important features which can detect the flyer peptides, but

Table 6.7: Important properties for peptide detectability

Property	**Percentage of overlap between the GP runs**	Domain Experts
Retention coefficient in pH2	53.33	✓
Fraction of site occupied by water	50.00	✓
Transfer energy organic solvent water	50.00	✓
Hydrophobicity-related index 3.0 pH	93.33	✓
Isoelectric point	56.67	✓
Mass	53.33	✓
Absolute entropy	53.33	✓
Gas phase basicity	56.67	✓
Normalised composition of mt protein	53.33	✓
Length	80.00	✓
Normalised frequency of alpha-helix	50.00	✗
Normalised frequency of beta-sheet from LG	56.67	✗
Normalised frequency of C-terminal non beta region	50.00	✗
Slopes tripeptide, FDPB VFF neutral	50.00	✗

is also more efficient in selecting a smaller number of important properties.

6.6 Biomarker Verification Steps

The input to the proposed GP biomarker verification algorithm is peptide sequences which then act as examples in the test set. Training is already performed using the various datasets, and hence, the biomarkers candidates will be the input to the algorithm as the test set's instances. Using the verification algorithm, a biomarker is verified if it is classified as a flyer

or observed peptide. Otherwise, it is classified as a non-verified or non-trusted biomarker.

The steps involved for the biomarker verification are the following:

1. The biomarker detection approaches output biomarkers in the form of m/z to the intensity value. Hence, the first step is to perform identification of the biomarker candidates using the m/z values to convert them into peptide sequences. These m/z values are used to perform the identification of the resulting peptides (peptide mass fingerprinting).

2. Convert the peptide sequence of each biomarker candidate to numerical feature vector that are the physiochemical properties of amino acids

3. Using the converted biomarker peptides, directly apply them to the GP peptide detection method for biomarker verification as a test set.

4. Output the list of peptides in the observed class as the verified biomarkers.

6.7 Chapter Summary

The overall goal of this chapter was to develop a new GP system for measuring peptide detectability. This goal was successfully achieved by developing PEP-GP, a GP method which predicts if a specific peptide is observable or non-observable in the mass spectrometry experiment. The proposed method performs both feature selection and classification simultaneously and also takes into account the class imbalance problem in the data. According to the results, PEP-GP succeeded in keeping the balance between both the observed and the non-observed classes and outperformed the benchmark classification methods used for comparison on

these datasets. Furthermore, the proposed method achieved better performance than ESP-Predictor on the datasets used. For feature selection results, neither PEP-GP nor the other feature section algorithms used for comparison as a separate stage improved the average accuracy of the classification. This suggests that separating the feature selection from classification for the class imbalance problems is not useful for these tasks. In addition, PEP-GP is better than the classifiers used for comparison even when data resampling is performed. This means that GP is more suitable for peptide detection as it has the potential to detect the flyer peptides which lie on the minority class, and can deal with the data without prior resampling.

Chapter 7

GP for Multiple Alignment of MS data

7.1 Introduction

The preprocessing of MS data, specifically the alignment step can directly affect the biomarker detection process. This chapter presents some initial results of the use of GP for alignment of MS data. A new GP method is proposed here and tested on a number of benchmark datasets. The results show that the new GP method has outperformed five different benchmark alignment methods in most of the datasets that indicate the potential of GP to solve the various complex tasks involved in the MS data analysis.

7.1.1 Chapter goals

The goal of this chapter is to develop a GP based method for multiple alignments of LC-MS peak maps that can correct the distortion of RT in multiple maps simultaneously. The method aims at aligning the MS data features which will help in improving the biomarker detection process. The proposed method is composed of two main phases: the first is to match the peaks across multiple maps and the second is to find the best

dewarping function for the RT of the matched peaks. The method is tested on one proteomics dataset and two metabolomics datasets and compared against five benchmark algorithms. Specifically, we will perform the following:

- develop an appropriate peak matching approach across multiple LC-MS maps with different numbers of peaks;
- design a GP method to perform multiple-output regression;
- model the terminal set of GP, to perform multiple regression simultaneously; and
- investigate whether the new GP method outperforms the conventional alignment methods on these datasets.

Chapter Organisation: The rest of the chapter is organised as follows. Section 7.2 gives brief background of the MS alignment problem. Section 7.3 describes the proposed approach and the new GP method. The experimental design, the datasets' description and preprocessing are presented in Section 7.4. Section 7.5 reports the experimental results along with the discussions. The conclusions and future work are presented in Section 7.6.

7.2 Background

The LC-MS spectrum is a 3D map, called LC-MS map, which consists of mass to charge ratio (m/z), retention time (RT) and ion intensity count (Int). LC-MS can be used for providing quantitative and qualitative information about the proteins in a biological sample [171]. Such information is useful in several applications including system biology, functional genomics and biomarker detection. For these applications to be successful, ideally the m/z and RT of the same molecule at different spectra among the LC-MS replicate runs detected in the same LC-MS platform should be

the same. However, this is not always the case. In particular, there is a large shift and sometimes distortion in RT between different runs [171]. Also, the m/z values show smaller distortion that introduces ambiguity in peak matching in comparative analyses. Moreover, the variations in RT may show non-linear deviations and can be greater than predicted [98]. Therefore, an effective algorithm is required to address two main tasks. The first is to match the peaks arising from the same peptides at different runs within certain m/z and RT windows, and the second is to find the correct transformation of the RTs to make comparison [99] between the intensity values effectively.

The methods for alignment of LC-MS spectra can be classified into two groups. The first group is the raw-based methods, which select the set of significant peaks from raw data and use these peaks as a reference for aligning the data. These methods can avoid the errors due to feature detection but they have high computational cost [72]. The second group is the peak based methods, where the alignment is done by extracting features and grouping corresponding features (peaks) from different LC-MS runs [171]. However, feature extraction and centroidisation can introduce some errors [72]. Therefore, the quality of the alignment algorithm will depend on mainly the quality of these preprocessing paradigms.

Examples of raw-based methods include the Hidden Markov Models (HMMs) approach presented in [104], where the alignment of RT and the normalisation of the peak intensities were done at the same time. HMMs were used to represent the correct retention times and the parameters of the model were estimated using the maximum likelihood estimation. A star-wise manner alignment of either raw or feature maps was depicted [98] in the open source platform *OpenMS*. In the first phase, features were matched together using pose clustering followed by linear regression to correct the retention time distortion. In the second phase, the dewarped maps were combined into a consensus map by using the nearest neighbour search. Li et al. [101] developed a pairwise feature-based alignment

algorithm in the open source software suite *SpecArray*. The algorithm computes all the pairwise alignments and combines them to a consensus map. The RT was corrected using a calibration curve and continuously aligning the pairs of features with similar m/z values. In [13] a star-wise alignment approach to feature maps was proposed, and the algorithm starts with pairing the most intense features with similar m/z values. After that, smoothing spline regression was used for dewarping and finally divisive clustering was used to obtain the consensus map. The RANdom SAmple Consensus (RANSAC) algorithm was used in the *MZmine2* [135] framework to find features that fit a non-linear model within a user-supplied m/z and RT tolerances. A locally weighted scatter plot smoothing regression method was used on all the points obtained from RANSAC. Genetic algorithms were used in [130] to predict the RT dewarping function.

Most of these approaches for alignment of LC-MS data focus on solving the pairwise alignment problem, which produce somehow suboptimal results for multiple alignment problems.

GP has been successfully used for alignment and forecasting of time series data [2] and achieved good results. In particular, GP is well known for symbolic regression, which provides a potential for aligning LC-MS data. However, GP has not been used for the alignment of LC-MS datasets to date.

7.3 The new GP Alignment algorithm

In this section, a new algorithm for alignment of MS data using GP is proposed. This work is the first work using GP for alignment of MS data.

The objective of the alignment of LC-MS maps (we refer to each sample or run as a map) is to produce a consensus map that contains matching peaks of the same molecules after transformation of RTs. In other words, the aim is to produce peak lists that have similar m/z and RT values in order to perform comparisons of intensity values effectively [4].

The alignment approach proposed here works with peak data which has a much smaller amount of data than the raw maps. Therefore, it can be used to develop faster dewarping techniques. Figure 7.1 shows an overview of the proposed alignment approach, which starts with taking the peak lists as inputs. The main aim of alignment is to find the possible transformations that map the RT points of one map (reference map) $(r_1, r_2, ..., r_n)$ to the corresponding points of the other maps $(m_1, m_2, ..., m_x)$. To achieve this objective, the most matched partners must be detected by the peak matching approach that is used as an intermediate step to allow GP to search for the optimal transformation. The peak lists that have a different number of peaks are passed to the peak matching phase to detect the matched peak lists between the reference map and the other maps $((r_1, m_1), (r_2, m_2)....(r_n, m_n))$.

For pairwise alignment, GP can be used directly to evolve transformation functions. However, the multiple alignments of multiple maps require a different structure of the evolved programs of GP to determine the transformation of the multiple maps. Therefore, a new GP multi-branch tree approach is developed for correcting RTs of multiple maps simultaneously. Finally, GP outputs the corrected peak lists. The two phases of the alignment approach are described below. For presentation convenience, the new approach is called GPMS.

7.3.1 Peak Matching

The first phase of the approach is to identify the significant matching peaks across all maps. The criteria for peak matching is the distance between the m/z and RT in the reference map and the other maps. The procedure for peak matching is as follows:

1. Randomly select a map from the dataset as a reference map $R = (r_1, r_2, ..., r_n)$.

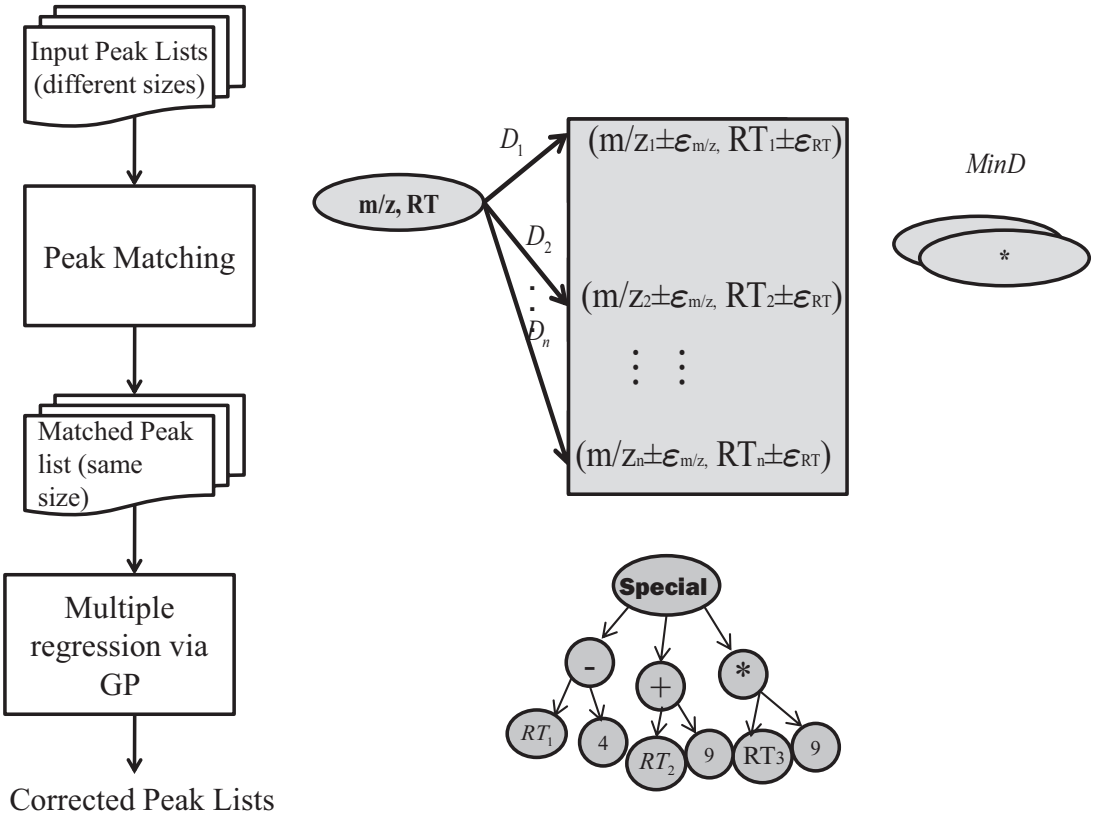


Figure 7.1: Overview of the alignment approach.

2. For each peak $(m/z_i, RT_i, Int_i)$ in the reference map, find the list of peaks in the next map $M = (m_1, m_2, ..., m_n)$ within a predefined m/z ($m/z_i \pm \varepsilon_{m/z}$) and RT ($RT_i \pm \varepsilon_{RT}$) tolerances and with the same charge.

3. Select the nearest neighbour (1-NN) peak from the list of peaks in the current map with respect to m/z, RT, and Int, and add the two peaks as significant peaks of the reference and current maps into the consensus map. The distance between the peaks is measured using the Euclidean distance between m/z, RT, and Int. A larger weight is given to m/z because RT and Int are much more tolerable than m/z. The Euclidean distance is given by:

$$ED = \sqrt{(W_1^2 * (R_{m/z} - M_{m/z})^2 + W_2^2 * (R_{RT} - M_{RT})^2 + W_3^2 * (R_{Int} - M_{Int})^2)}$$

where ED is the Euclidean distance between the two peaks of the reference (R) and the current (M) maps and W_1=0.7, W_2=0.2 and W_3=0.1.

4. Mark the selected peak on the current map as a processed peak so that it will not be selected again as the nearest neighbour to another peak.

5. Repeat steps 2-4 on all the maps until all the peaks in all maps are processed. If there is no corresponding peak found in half of the maps, all significant peaks related to this peak are removed from the significant peak lists.

After identifying the matching peaks across all maps, the list of matching pairs is passed to GP to correct the RT values.

7.3.2 GP Multi-Branch Regression for Multiple Alignment

Unlike most of the previous RT alignment algorithms, our GP method corrects RTs of all maps simultaneously. The main advantage of this regression GP technique is that it can work efficiently. Another advantage is not having the requirement of a specific *gold standard* reference map for alignment of the rest of the maps. In other words, any map can be selected as a reference to align the rest of the maps. In this approach, we use the tree-based GP [190] for this task but we modified the tree structure as a multi-branch tree. In the multi-branch GP approach, each individual is composed of several branches and each branch is responsible for evolving a part of the solution [143, 190]. The final solution is integrating all these partial solutions through a special node which represents the root node [28, 190]. The number of children of the special node is equal to the number of maps to be aligned. The children of the root node are the functions. The function node can also take other function nodes as its children. The terminal nodes of each branch are the RTs of a specific map or random

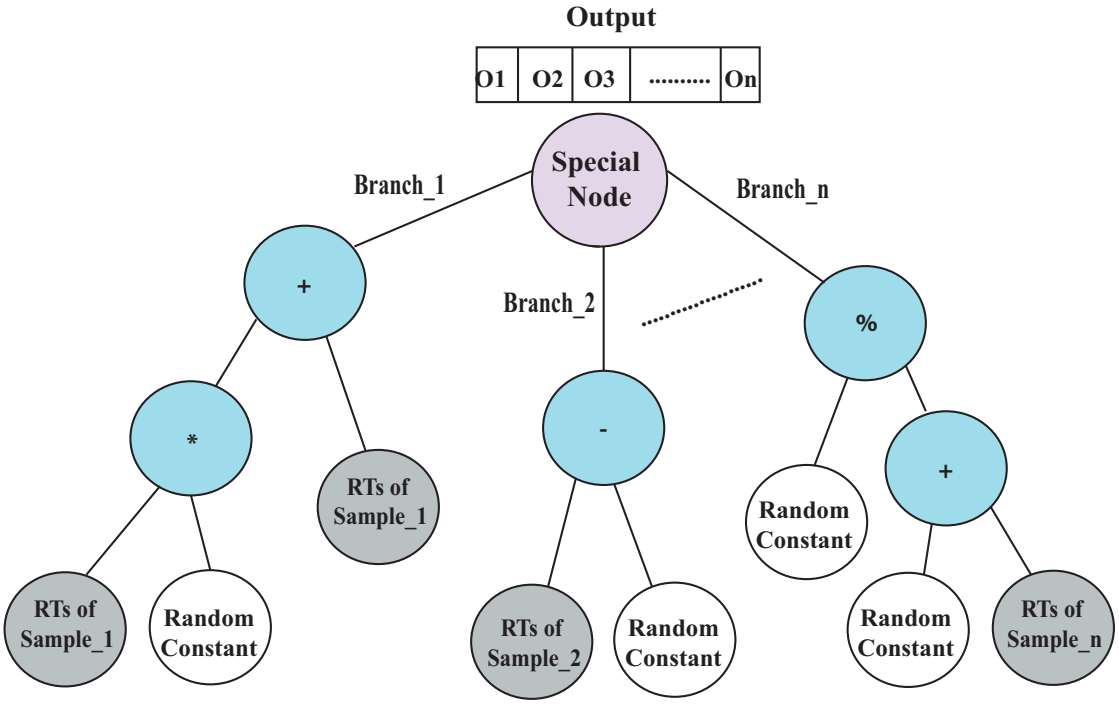


Figure 7.2: Tree structure in the Multiple Alignment GP.

constants. The same branch cannot contain RTs from different maps. The structure of the multiple-output regression tree is shown in Figure 7.2.

In the rest of the section, we will describe the terminal set, the function set and the fitness function of the new GP method.

7.3.3 Terminal and Function Sets

An LC-MS sample is a 3D map composed of the m/z values, RTs, and the intensity counts (Ints). The objective here is to correct the RTs of all maps to the corresponding RTs of the reference map. Therefore, the terminal set is composed of the RTs of N maps. We consider each input to GP as N RT dimensions (equal to the number of maps). For example, if we have three maps, each input to the terminal set is composed of three RT variables. We also used a random generated constant in the range of [-10,10] in the

terminal set. Hence, our terminal set is composed of RT values of all maps and random constant values. The function set used for this problem is $F = \{+, -, \times, \%, cos\}$, where % is the protected division operator, which returns zero if the division is by zero. The aim of using the *cos* operator is to evolve non-linear function for prediction and regression of the complex RT deviations. The outputs (O_i) of each map are collected by the special node that is the root of the tree.

7.3.4 Fitness Function

For function approximation tasks, the performance can be measured as an error between the predicted and the real target values. As we have multiple outputs, with each output corresponding to RTs of one map in the dataset, we calculate the sum of errors between the multiple outputs (which are the estimated outputs of the genetic programs) and the reference map output. The root mean square error (RMSE) is used as a fitness function. Thus, the GP framework is to minimize the fitness so that the generated programs lead to the minimum error between the RTs to be predicted. The RMSE fitness function is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(RT_{ij} - \hat{RT}_{ij})^2}{N}}$$

where N and M are the number of maps and the number of RTs to be corrected in each map respectively. RT_{ij} is the i^{th} real RT value of the j^{th} map while $\hat{RT}_{ij}$ is the i^{th} estimated RT value of the j^{th} map by the GP program.

7.4 Experiments Design

7.4.1 Datasets

We tested the proposed approach on one proteomics dataset (P_1) and two metabolomics datasets (M_1, M_2) obtained from the Open Proteomics Database

(OPD) [138] and Lange et al. [98]. Dataset P_1 contains two LC-MS runs with six different fractions and it originates from an *E.coli* sample. For this dataset, each fraction is composed of pairs of LC-MS runs. The dataset was analysed using LC/MS/MS with an ESI ion trap mass spectrometer (ThermoFinnigan Dexa XP Plus). It was exported into mzXML centroided mode and preprocessed using TOPP tools [90] to produce the peak lists which consist of the m/z, RT, intensity values and ignoring the charge states. The numbers of peaks in each fraction run were between 400 to 5800. A partial ground truth was produced using the first fraction of the dataset by linking the LC-MS spectra to the MS/MS of the SEQUEST search. More details about the steps for dataset preparation, analysis, preprocessing, and parameters optimisation can be found in [98]. For the two metabolomics datasets, Arabidopsis thaliana leaf tissues were analysed using two different LC-MS setups. An API QSTAR Pulsar i (Applied Biosystems/MDS Sciex) was used to produce 44 spectra for the M_1 dataset and a MicrOTOF-Q (Bruker Daltonics) to produce 24 spectra for the M_2 dataset. Peak extraction was done using XCMS software [152] resulting in 4000 to 17600 peaks in each spectrum. The ground truth was generated in the same study by selecting the high confident peaks. Those were the peaks found in more than four runs, having the same RT and also showing a high correlation in their peak shapes. Table 7.1 summarises the datasets used in the approach.

Table 7.1: Datasets used in the approach

Dataset	**Number of LC-MS runs**	**Number of peaks**
P_1	2 on 6 fractions	400-5800
M_1	44	4000 -17600
M_2	24	4000 - 17600

7.4.2 Genetic Operators and Parameters

The initial populations of GP are generated using the ramped half-and-half method. Each population consists of 1024 individuals to reduce the early convergence probability. The tournament selection method is used to select the individuals that can perform well for reproducing the new generations. The size of the tournament is set to 5. The standard crossover and mutation are used here with ratios of 80%, and 19% respectively. Elitism is also used with a ratio of 1%. The depth of each individual is kept between 2 and 8. Each evolutionary process stops at the maximum generation of 30 unless a perfect error of zero is found. The process is repeated for 30 independent runs. The random seeds for each of the 30 runs in each set of experiments are different. The peak matching phase parameters are as follows: the m/z tolerance and RT tolerance are set to 1.5, and 100, respectively for dataset P_1 for all the fractions. For datasets M_1 and M_2 the m/z tolerance and RT tolerance are set to 0.011, and 20, respectively, for both datasets. Those parameters were selected via initial search, and they achieved good results.The GP implementation used in our experiments is the Evolutionary Computing Java-based (ECJ) package [176]. Table 7.2 describes the parameters used in the experiments.

7.4.3 Benchmark Algorithms

We compared our approach with previously published results of five publicly available benchmark algorithms for alignment of LC-MS maps that are: msInspect [13], MZmine [80], SpecArray [101], XAlign [189] and XCMS [152]. msInspect [13] works in a star-wise manner which aligns all maps with respect to a specific reference map, which is the map with the minimum number of peaks. The process starts with the selection of the most intense peak within a certain RT tolerance and the removal of the rest of the peaks. After that, pairing the remaining peaks with peaks of similar m/z is performed. Smoothing spline regression is used for dewarping, and fi-

Table 7.2: GP parameters

Parameter	**Value**
Initialization method	Ramped Half-and-Half
Initial tree Depth	2
Maximum tree depth	8
Generations	30
Mutation probability	19%
Crossover Rate	80%
Elitisim	1%
Population Size	1024
Selection type	Tournament
Tournament Size	5
m/z tolerance	1.5, 0.011,0.011 for P_1, M_1, M_2 respectively
RT tolerance before correction	100, 20, 20 for P_1, M_1, M_2 respectively

nally divisive clustering is used to obtain the consensus map. The main disadvantage of this approach is the removal of less intense peaks, which might lead to the loss of many important peaks. MZmine [80] works by scoring the similarity of all features against a master list and if the score is "good enough", the feature is assigned to the best matched row. MZmine does not perform any transformation of RT. The SpecArray [101] schema works as pairwise alignment and combines the pairwise aligned maps into a consensus map until all maps are aligned. SpecArray is not applicable to datasets with a big number of maps. XAlign [189] also works in a star-wise manner and selects the most intense peaks within a user defined m/z and RT tolerance, and the map with the minimum difference to the average RTs is chosen as a reference map. After dewarping the RT, the features with high correlation coefficients are selected to form the consensus map. XCMS [152] works as a multiple alignment approach where peak matching is performed in the first phase by using a fixed interval bin and us-

ing kernel density estimation to determine the distribution of the features. Boundaries of regions with features that have similar RTs are selected. Finally, non-linear regression is used to correct RTs.

7.4.4 Performance Evaluation

The performance of the proposed approach is measured through the precision (PR) and recall (RE) measures. Precision is the probability that a found item is relevant, which in our case is the percentage of correctly aligned peaks among all the peaks aligned by the approach.

$$PR = \frac{\text{Number of correctly aligned peaks}}{\text{Total number of peaks aligned}}$$

Recall is the probability that a relevant item is found (the percentage of the correctly aligned peaks among the peaks in the ground truth [174]).

$$RE = \frac{\text{Number of correctly aligned peaks}}{\text{Total number of peaks in the ground truth}}$$

The harmonic mean of the precision and recall is measured through the F-measure [174].

$$\text{F-measure} = \frac{\text{2*PR*RE}}{\text{PR+RE}}$$

Precision and recall of alignment were calculated using the evaluation script provided by Lang et al. [98].

7.5 Results and Discussions

7.5.1 Effectiveness Performance

GPMS is initially tested for the pairwise alignment on P_1 which is available in six different fractions. P_1 shows a large deviation in RT values which is a challenge for the alignment tool to correct the RT. Tables 7.3 and 7.4

show the results of the five conventional approaches compared to our approach notated as GPMS. As shown in Tables 7.3 and 7.4, GPMS achieved much better performance than msInspect and SpecArray in all the three datasets. GPMS outperformed all other methods in three fractions of P_1. For the first fraction (00), the mean of the 30 runs of GPMS is better than msInspect by 44% in terms of precision, 30% in terms of recall and 38% in terms of F-measure. For the other approaches, GPMS improves the precision by 1-25%, the recall and F-measure by 1-21%. For fraction (20), GPMS achieves similar performance as XCMS and has the third rank after MZmine and XAlign. GPMS performs better than msInspect, SpecArray and XCMS for fraction 40. Furthermore, our new method is the third best after MZmine and XAlign for the same fraction. For fractions (60) and (100), GPMS outperforms all other methods in terms of precision (which reaches 1.00 for the fraction (100)) and F-measure. The proposed method has the best recall in fraction (60) while in the fraction (100) it has the third best recall after Xalign and XCMS. Finally for the fraction (40), the performance of GPMS was slightly better to XCMS, and it is the second best after MZmine. In general, for P_1 the proposed method outperforms the other methods in three fractions, the second best in two fractions and third best in one fraction.

For datasets M_1 and M_2, which contain 44 and 24 maps respectively, the challenge for the alignment approach on these complex metabolomics datasets is to assign the most suitable matches and to correct the RT distortion across multiple maps. SpecArray did not manage to produce any results for these complex alignment tasks. As shown in Table 7.4, GPMS appears to be more powerful in aligning a large number of maps as in the dataset M_1 (44 maps). For M_1, it has better performance than other methods by 1-31% in terms of precision and 2-49% with respect to F-measure. This suggests that the proposed method can be more powerful for multiple map alignment. The performance of GPMS outperforms msInspect in terms of precision by 41.87%, XCMS by 1% and it is equal to XCMS for

Table 7.3: Proteomics dataset P_1 alignment results

Fraction	Measure	msInspect	MZmine	SpecArray	XAlign	XCMS	GPMS		
							Min	Max	Mean ±St.Dev.
00	Precision	0.38	0.81	0.61	0.82	0.58	0.82	**0.83**	**0.83±0.003**
	Recall	0.52	0.75	0.61	0.82	0.62	0.82	**0.83**	**0.82±0.004**
	F-measure	0.44	0.78	0.61	0.82	0.60	0.82	**0.83**	**0.82±0.004**
20	Precision	0.45	**0.88**	0.62	0.85	0.80	0.80	0.82	0.81±0.0100
	Recall	0.56	**0.87**	0.62	0.85	0.81	0.80	0.80	0.80±0.0000
	F-measure	0.50	**0.87**	0.62	0.85	0.80	0.80	0.81	0.81±0.0060
40	Precision	0.48	**0.90**	0.75	0.87	0.80	0.83	0.84	0.84±0.002
	Recall	0.63	**0.87**	0.75	**0.87**	0.81	0.81	0.81	0.81±0.0
	F-measure	0.54	**0.88**	0.75	0.87	0.80	0.82	0.82	0.82±0.003
60	Precision	0.54	0.84	0.71	0.87	0.75	**0.91**	**0.91**	**0.91±0.000**
	Recall	0.73	0.79	0.71	0.87	0.78	**0.92**	**0.92**	**0.92±0.000**
	F-measure	0.62	0.81	0.71	0.87	0.76	**0.91**	**0.91**	**0.91±0.005**
80	Precision	0.57	**0.94**	0.74	0.90	0.88	0.90	0.90	0.90±0.000
	Recall	0.70	**0.92**	0.74	0.90	0.89	0.89	0.89	0.89±0.0000
	F-measure	0.63	**0.93**	0.74	0.90	0.88	0.90	0.90	0.90±0.0040
100	Precision	0.56	0.92	0.77	0.96	0.96	**1.00**	**1.00**	**1.00±0.000**
	Recall	0.82	0.94	0.77	**0.96**	**0.96**	0.94	0.94	0.94±0.000
	F-measure	0.67	0.93	0.77	0.96	0.96	**0.97**	**0.97**	**0.97±0.000**

Table 7.4: Metabolomics datasets M_1 and M_2 alignment results

Fraction	Measure	msInspect	MZmine	SpecArray	XAlign	XCMS	GPMS		
							Min	Max	Mean ±St.Dev.
M_1	Precision	0.46	0.74	-	0.70	0.70	**0.77**	**0.77**	**0.77±0.003**
	Recall	0.27	0.89	-	0.88	**0.94**	0.89	0.91	0.9±0.004
	F-measure	0.34	0.81	-	0.78	0.80	**0.83**	**0.83**	**0.83±0.001**
M_2	Precision	0.47	**0.84**	-	0.79	0.78	0.79	0.79	0.79±0.001
	Recall	0.23	**0.98**	-	0.93	**0.98**	0.90	0.90	0.90±0.000
	F-measure	0.31	**0.90**	-	0.85	0.87	0.84	0.84	0.84±0.001

M_2. In terms of recall, GPMS performed better than msInspect by 53% and outperformed SpecArray which did not manage to achieve results in terms of F-measure. Overall, the performance of GPMS is the second best with respect to precision, third best with respect to recall and F-measure in M_2. In general, GPMS is among the top two methods or even performs best (for 00, 60, 100 of P_1, M_1).

7.5.2 Efficiency Performance

Another comparison is done in terms of the run time of each of the methods, and the results are shown in Table 7.5. For all the datasets, the average run time of GPMS is much better than all other approaches. The computational cost (in terms of time) of GPMS is lower than the rest of the methods, which represents another advantage of GPMS. For all the datasets, GPMS improves the efficiency by an order of magnitude over the rest of the methods except for XCMS. GPMS is also more efficient than XCMS in terms of computational time for P_1 and M_2. Moreover, the efficiency of GPMS for M_2 in one of the runs is also better than XCMS.

Table 7.5: Comparison of run time of GPMS with other approaches (in seconds)

Dataset	**msInspect**	**MZmine**	**SpecArray**	**XAlign**	**XCMS**	**GPMS**		
						Min	Max	Mean ±St.Dev.
P_1	60	40.2	111	69	54	**4.1**	**9.8**	**6.1±1.20**
M_1	720	1200	-	3060	54	**36.34**	64.92	64.92±4.97
M_2	2160	2640	-	2100	348	**81.10**	**94.20**	**87.37±3.23**

7.5.3 Interpretation of the Evolved Regression Models

Table 7.6 shows two examples of the evolved models for fractions (00) and (100). SPE refers to the special node that is the root node collecting the

Table 7.6: (a) An evolved model for fraction (00) with some examples of inputs and outputs of the model. (b) An evolved model for fraction (100).

(SPE T_0 (- (- T_1 9.05) (*cos* T_1)))

Input		**Output**	
T_0	T_1	T_0	T_1
1263.95	1271.96	1263.95	1263.89
1307.84	1315.58	1307.84	1307.09
1708.72	1717.28	1708.72	1708.10

(a)

(SPE T_0 (+ T_1 17.56))

Input		**Output**	
T_0	T_1	T_0	T_1
182.95	165.425	182.95	182.98
111.45	94.12	111.45	111.68
455.08	438.12	455.08	455.68

(b)

multiple outputs of the tree. T_0 refers to the RTs of the first map while T_1 refers to the RTs of the second map. The first map (T_0) is selected as the reference map in which the RTs of both maps should be corrected according to it. The dewarping functions of both inputs are determined simultaneously through the multiple branches. As shown in Table 7.6 (a), GP managed to determine the correct amount of the shift for the RTs of the second map (T_1) through a nonlinear dewarping model in the second branch of the tree. The RTs of the first map (T_0) (the first branch of the tree) is kept the same as it has been selected as the reference map. Some examples are shown in the same figure where the inputs to the models and the mapped outputs after correction show that GP has successfully aligned the maps with respect to the reference map. The evolved model for fraction (100) is shown in Table 7.6 (b) where the GP dewarping function has managed to correct the distortion of RTs through a linear function. Examples of inputs and outputs of fraction (100) are also shown in Table 7.6 (b), where it is clear that GP managed to correct the distortion between the inputs and the outputs.

7.6 Chapter Summary

In this chapter, a new method is proposed for multiple alignments of LC-MS peak data. The proposed method has two phases. The chapter represents preliminary results for the first work of GP for multiple alignments of LC-MS data. In the first phase, the partner peaks across multiple maps are detected to form the matched peak lists. In the second phase, the matched peak lists are passed to GP to perform the correction of RTs of all maps simultaneously. The new GP approach is depicted by dividing the tree into multiple branches, where each branch produces the output dewarping function of each map with respect to the reference map. The proposed GP-based method (GPMS) was tested on one protoemics dataset of six different fractions. The results show that GPMS achieves better pre-

cision, recall and F-measure than five other LC-MS benchmark alignment methods for three fractions of the proteomic dataset and one metabolomic dataset which has larger number of maps. This suggests that GPMS is more powerful in the multiple alignments of LC-MS data. The proposed method also shows very competitive results in the rest of the datasets. GPMS, in general, is always either the best or among the two top methods for these datasets. Furthermore, the proposed GP method is much more efficient in terms of computational time than the benchmark methods.

Chapter 8

Conclusions

8.1 Introduction

The main goal of this thesis was to improve the biomarker detection process for classification of MS data through the use of GP. The thesis focuses on investigating the capability of GP for different tasks, namely feature manipulation for biomarker detection and biomarker verification (peptide detection) through classification and feature selection of unbalanced peptide data. This overall goal was fulfilled through developing new GP approaches to feature selection, feature construction and classification of the high dimensional MS data to discover the relevant features for classification and construct new high-level features for the aim of improving the classification performance and reducing the number of features. The goal was extended to the verification of the detected features (biomarkers) using GP.

The thesis developed a number of new GP algorithms to automatically select subsets of features and construct a number of high-level features. The proposed methods were tested on a number of MS datasets, and compared to the existing methods. The results show that GP can effectively be used for the challenging task of MS data analysis. The proposed meth-

ods succeeded in improving the MS biomarker detection and verification performances.

Chapter Organisation: The rest of this chapter is organised as follows. Section 8.2 explains the achieved objectives of the thesis and the main findings from each chapter. Section 8.3 gives conclusions for each of the thesis research objectives. Section 8.4 presents some potential research areas for future work.

8.2 Achieved Objectives

This thesis has achieved the following research objectives:

- Proposes a new ensemble selection method and a new ranking mechanism in GP for single objective embedded feature selection in MS data for biomarker detection. The proposed ensemble method uses the advantage of GP for automatic feature selection to further select features from two different feature ranking metrics.

  The proposed ranking scheme uses the frequency of occurrences of features in the evolved tree to generate a new rank for each of the selected features. By combining the new ensemble and ranking mechanisms, the proposed GP based algorithm can significantly reduce the number of features by further selecting from the top features produced from two different metrics, and improve the classification performance over using all the original features. It also outperforms the two traditional ranking metrics individually and a standard wrapper genetic algorithm method for feature selection. The use of GP as a classifier also outperforms standard classification algorithms. The method also detected most of the predefined biomarkers of the spiked-in datasets.

- Proposes a new GP based approach to embedded single objective multiple feature construction. Unlike existing GP multiple feature

construction algorithms which use multiple individuals to construct multiple new high-level features, the proposed approach uses a single GP tree to construct multiple features during the evolutionary process. The algorithm uses the capability of GP to automatically combine the original features with the functions from the function set to find the new high-level features.

The proposed approach can successfully find a set of high-level features which significantly improve the classification accuracy. It can also detect the true biomarkers of the MS datasets. The method is compared to using the original selected features from the same method before the combination using the functions. It is also compared to another GP method for feature selection.

After applying the algorithm on a number of benchmark MS datasets, the results show that the proposed method has a significant potential to reduce the number of features more than feature selection, and select a much smaller feature set than the original set of features. At the same time, it successfully improved the classification performance more than the methods used for comparison on the used datasets and managed to detect the predefined biomarkers.

- Proposes the first multi-objective embedded feature selection approach using GP for the high dimensional MS data. The proposed multi-objective approach aims to minimise the number of features, and maximise the classification performance of GP using the smaller subset of features.

  The proposed approach has the potential to successfully evolve a set of non-dominated solutions that are composed of feature subsets. The solutions in the Pareto front have a smaller numbers of features and better classification performance on GP than the traditional GP based single objective algorithm. After comparing the algorithm to the benchmark multi-objective approaches, it is shown

that it can outperform these methods effectively on the used datasets in terms of both classiifcation performance and the number of predefined biomarkers detected.

The multi-objective approach for feature selection is modified for feature construction in order to examine the construction of high-level features using the multi-objective optimisation. Based on the method proposed for multiple feature construction, the proposed approach extends the single objective algorithm feature construction to a multi-objective one. The proposed approach managed to reduce the number of evolved constructed features and simultaneously increase the classification accuracy. The methods show further improvement of both the number of features and classification performance over the aforementioned proposed multi-objective feature selection approach in most of the datasets used, and hence, outperform the single objective approach and the benchmark multi-objective approaches for feature selection.

- Proposes a new GP biomarker verification method through peptide detection. The method acts as a verification stage for the biomarker candidates, represented as peptides, through the examination and prediction whether they are going to be observed in the mass spectrometer. The proposed approach solves the problem of the unbalanced peptide data, where usually the biomarkers peptides are in the minority class. Using the flexibility of GP to build classification models that are not biased by the majority class, the proposed approach succeeds in evolving classifiers which improve the minority class's accuracy. Moreover, the proposed approach uses the automatic feature selection capability in GP to discover the important features represented as the physiochemical properties for detection of the peptides. After testing on a number of datasets, including an in-house dataset, and comparing the algorithm to benchmark classi-

fiers and feature selection algorithms, it is clear that GP for peptide detection can make better progress. The method is also compared to ESP-Predictor, which is a benchmark peptide detection method and shows improved performance.

- Presents some initial work for the use of GP for alignment of MS data. Alignment of MS data is a main preprocessing step in the data analysis and can directly affect the biomarker detection process.

8.3 Main Conclusions

This thesis finds that GP can effectively address the biomarker detection problem through feature selection and construction in single objective or multi-objective ways. The use of GP has been found to be useful for handling the challenges in MS data, which includes high dimensionality (MS data that typically has thousands of features), small number of examples, and a high percentage of noise.

GP showed to be successful for this task and managed to select and construct features of high capability for improving the classification performance. The thesis also finds that GP can be used effectively for verification of biomarkers through classification of unbalanced peptide data.

This section presents and discusses the main conclusions drawn from the four contributions of the thesis.

8.3.1 Ensemble and Ranking of Features Mechanisms in GP

Chapter 3 proposes a new single objective GP feature ranking approach which uses a new ensemble mechanism for selecting and ranking features using an embedded approach.

Ensemble feature selection mechanism in the GP embedded approach

The combination of different metrics for feature ranking is found to be useful for improving the feature selection process. This is because each metric has its advantages in ranking the top features, and the use of GP to further select from the top ranking features and generate better feature combinations.

Biomarker detection was usually performed using a single feature ranking metric which has the risk of ignoring the important relationships between features. Using more than one metric provides a better combination of features and has the advantage of reducing the search area for these metrics. GP can automatically select features and reduce the features to be used in classification. Using an embedded approach can also evolve the classification model along with the feature selection process. This is found to be useful because it combines the advantage of the wrapper approaches and filter approaches of using a classifier for evaluation of features and avoiding the high computational cost. Another advantage of using the embedded approach is the better understanding of feature interactions through the linking of the feature selection and the classification processes.

Ranking mechanism in GP

The tree-based GP may select the same feature more frequently than another feature. It is found that the more frequently selected features can be more relevant to classification and hence, they should be given a higher rank. In addition, GP as an evolutionary algorithm starts from a random seed, and therefore, it must be run several times. The appearance of the feature in several runs gives it more score. The new rank for the automatically selected features has been found to be useful for improving the performance of classification.

Generality of embedded approach

Although the performance of GP as a classifier is better with its selected features, the embedded approach for feature selection has the potential to be generalised to other classifiers. Testing the features of the embedded approach feature selection with other classifiers has been found to be effective in improving their performance.

8.3.2 Single objective Multiple Feature Construction

This thesis proposes the first multiple feature construction approach for biomarker discovery in MS data (Chapter 4). From Chapter 4, it is found that the use of the high-level features can significantly influence the performance of the classifier. GP can construct multiple high-level features automatically from a single GP tree. These features successfully reduce dimensionality and improve classification accuracy more than the selected features. The new fitness function (which is considered as computationally cheap) proposed also managed to construct features that have more power in distinguishing the classes.

Applying the method on datasets with predefined biomarkers confirms its capability of detecting the biomarkers defined.

8.3.3 Multi-objective Feature Manipulation

The first GP multi-objective biomarker detection is proposed in this thesis (Chapter 5). It is found that the multi-objective optimisation using GP can successfully be used for feature manipulation in the high dimensional MS data. Either feature selection or construction involves a large search space in MS data, which is characterized by high feature-to-sample ratio. This makes selection or construction of features in MS data a challenging and complex task.

Multi-objective Feature Selection

The thesis finds that GP using the ideas from SPEA2 or NSGAII can select non-dominated solutions from the original set of features. These non-dominated solutions have a smaller number of features, a better classification accuracy and a better biomarker detection rate than the single objective GP and either SPEA2 or NSGAII individually for feature selection.

Multi-objective Feature Construction

This thesis proposes the first multi-objective feature construction approach for biomarker detection. Examining the multi-objective approach for feature construction finds that the achieved Pareto Front contains new high-level features which significantly improve the classification performance and reduce the number of generated features from the evolved tree more effectively than the single objective proposed feature construction approach discussed in Chapter 4.

The approach further improves the classification performance over the aforementioned multi-objective feature selection and hence, it outperformed the single objective and the benchmark multi-objective approaches for feature selection.

8.3.4 Biomarker Verification

The thesis is the first to link the biomarker detection and their verification as an intermediate stage before the experimental validation stage. In Chapter 6, the thesis proposes the first use of GP for biomarker verification. It is also the first verification algorithm which solves the imbalance problem in the peptide data.

It is found that GP for biomarker verification can significantly improve the accuracy of the minority class which include the predicted peptides to be observed in the mass spectrometer. These observed peptides are classified as the verified peptide biomarkers.

8.4 Future Directions

This section provides some possible future directions in the use of GP for biomarker detection and MS data analysis.

8.4.1 Single Objective GP for Feature Construction

The GP-based system proposed in this thesis produces multiple high-level features from a single evolved program which depends on the tree depth. To reduce the dimensionality of the evolved features, the arithmetic simplification of the tree might be considered. This can help in evolving a smaller number of features.

8.4.2 Building a multi-class GP classification system for MS data

The classification of MS data in case of multiple classification is more difficult than binary classification. This due to the high dimensionality, small number of samples in each class and often imbalance of the data. Sometimes, the MS data involves more than two classes (e.g. Stage 1, Stage 2 and Healthy classes). In the thesis, fixed interval thresholds are used for multiple classes. As future work, one might consider building pairwise GP classifier models or using multiple output GP for this task.

8.4.3 GP for Quantification of Proteins and Peptides

The thesis discusses a biomarker verification method that predicts the peptide's detectability. The system can be used for peptide quantification by using the detection probability as an absolute intensity of the peptide. Determining the absolute quantity of peptides, which will help in predicting the real concentration of the specific peptide, can be more accurate by us-

ing GP for solving a symbolic regression problem for regressing the actual intensity of these peptides.

8.4.4 GP for discovering the Pathways of the Diseased Metabolites

Modeling the chemical reactions of the biomarkers in the disease chemical network is useful for understanding the disease process. Metabolic pathways are the chemical reactions that take place within the cell. Modeling metabolic underlying biochemical processes is difficult and is not systematically approached. This analysis of the metabolic pathway is also an essential tool for metabolic biomarker validation.

Another future direction can be developing a GP method for regression of the expected pathway of the detected peptides. This method will help validate the metabolic biomarker detected by the GP feature selection method and compare these pathways with those in the metabolic pathways databases, such as the KEGG database.

8.4.5 GP for Alignment and Peak Extraction in MS data

The thesis presents some initial work on the use of GP for MS data alignment and peak extraction. This work can be extended to the use of GP for clustering the matching peaks. This will relate to another interesting but challenging research direction, i.e. using GP for peak matching through a clustering approach that can match the partner peaks better.

8.4.6 GP for Feature Selection in Unbalanced data

In peptide detection problem and generally in MS data, the datasets might suffer from an imbalance problem. GP has been successfully used in the thesis to evolve classifiers that can handle the imbalance of the peptide dataset. However, selecting the features from the unbalanced data is still

an issue. Hence, in the future, one might take into account the convergence of bagging and boosting with balanced bootstrap sampling in GP to handle this problem.

8.4.7 Further Selection from Multiple Solutions

The fact that GP produces multiple solutions might be challenged by biologists (a single solution is required). This disadvantage is related to all evolutionary algorithms and other stochastic algorithms such as neural network. Hence, further selection mechanisms to help biologists make a good choice need to be considered and developed in the future.

8.4.8 Verification and Experimental Validation of the Detected Biomarkers

In future work, the candidate biomarkers detected by the proposed GP biomarker detection approaches will be verified using the GP verification method presented in Chapter 6. This will help reduce the cost of the experimental validation through passing a smaller number of candidates. The verified biomarkers will be passed to the final experimental validation.

8.4.9 More MS and LC-MS Datasets

In future work, these GP methods would be evaluated on more biomarker detection tasks using both MS and LC-MS datasets.

Bibliography

[1] ABBATIELLO, S., MANI, D., KESHISHIAN, H., AND CARR, S. Automated Detection of Inaccurate and Imprecise Transitions in Peptide Quantification by Multiple Reaction Monitoring Mass Spectrometry. *Clinical Chemistry 56* (2010), 291–305.

[2] AHALPARA, D. P. Improved forecasting of time series data of real system using genetic programming. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation* (New York, NY, USA), GECCO, (2010), ACM, pp. 977–978.

[3] AHMED, S., ZHANG, M., AND PENG, L. Enhanced feature selection for biomarker discovery in LC-MS data using GP. In *Proceedings of 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico* (2013), pp. 584–591.

[4] AHMED, S., ZHANG, M., AND PENG, L. GPMS: A genetic programming based approach to multiple alignment of liquid chromatography-mass spectrometry data. In *Applications of Evolutionary Computation - 17th European Conference, EvoApplications, Granada, Spain* (2014), pp. 915–927.

[5] AHMED, S., ZHANG, M., AND PENG, L. Prediction of detectable peptides in MS data using genetic programming. In *Genetic and Evolutionary Computation Conference, GECCO '14, Vancouver, BC, Canada, July 12-16, 2014, Companion Material Proceedings* (2014), pp. 37–38.

[6] AHMED, S., ZHANG, M., PENG, L., AND XUE, B. Genetic programming for measuring peptide detectability. In *Simulated Evolution and Learning - 10th International Conference, SEAL 2014, Dunedin, New Zealand, December 15-18, 2014. Proceedings* (2014), pp. 593–604.

[7] AHMED, S., ZHANG, M., PENG, L., AND XUE, B. Multiple feature construction for effective biomarker identification and classification using genetic programming. In *Genetic and Evolutionary Computation Conference, GECCO '14, Vancouver, BC, Canada, July 12-16, 2014* (2014), pp. 249–256.

[8] ALPAYDIN, E. *Introduction to Machine Learning*. 2004.

[9] ATKINSON, A., COLBURN, W., DEGRUTTOLA, V., DEMETS, D., DOWNING, G., HOTH, D., OATES, J., PECK, C., SCHOOLEY, R., AND SPILKER, B. Biomarkers and surrogate endpoints: Preferred definitions and conceptual framework*. *Clin Pharmacol Ther 69* (2001), 89–95.

[10] BAGGERLY, K., MORRIS, J., WANG, J., GOLD, D., XIAO, L., AND COOMBES, K. A comprehensive approach to the analysis of matrix-assisted laser desorption/ionization-time of flight proteomics spectra from serum samples. *Proteomics 3*, 9 (2003), 1667–72.

[11] BANTSCHEFF, M., SCHIRLE, M., SWEETMAN, G., RICK, J., AND KUSTER, B. Quantitative mass spectrometry in proteomics: a critical review. *Analytical and Bioanalytical Chemistry 389*, 4 (2007), 1017–1031.

[12] BEAVIS, R. C., CHAUDHARY, T., AND CHAIT, B. T. ÎÂś-Cyano-4-hydroxycinnamic acid as a matrix for matrixassisted laser desorption mass spectromtry. *Organic Mass Spectrometry 27*, 2 (1992), 156–158.

[13] BELLEW, M., CORAM, M., FITZGIBBON, M., IGRA, M., RANDOLPH, T., WANG, P., MAY, D., ENG, J., FANG, R., LIN, C., CHEN, J., GOODLETT, D., WHITEAKER, J., PAULOVICH, A., AND MCINTOSH, M. A suite of algorithms for the comprehensive analysis of complex protein mixtures using high-resolution LC-MS. *Bioinformatics (Oxford, England) 22*, 15 (2006), 1902–1909.

[14] BHOWAN, U., JOHNSTON, M., AND ZHANG, M. Developing New Fitness Functions in Genetic Programming for Classification With Unbalanced Data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B 42*, 2 (2012), 406–421.

[15] BHOWAN, U., JOHNSTON, M., ZHANG, M., AND YAO, X. Evolving Diverse Ensembles Using Genetic Programming for Classification With Unbalanced Data. *IEEE Trans. Evolutionary Computation 17*, 3 (2013), 368–386.

[16] BHOWAN, U., ZHANG, M., AND JOHNSTON, M. Genetic Programming for Classification with Unbalanced Data. In *13th European Conference on Genetc Programming (EuroGP), Istanbul, Turkey* (2010), pp. 1–13.

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

[18] BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[19] BOEHM, A., PUTZ, S., ALTENHOFER, D., SICKMANN, A., AND FALK, M. Precise protein quantification based on peptide quantification using iTRAQTM. *BMC Bioinformatics 8*, 1 (2007), 214.

[20] BOGGESS, B. Mass Spectrometry Desk Reference (Sparkman, O. David). *Journal of Chemical Education 78*, 2 (2001), 168.

[21] BREIMAN, L. Random Forests. *Machine Learning 45*, 1 (2001), 5–32.

[22] CAI, J., SMITH, D., XIA, X., AND YUEN, K.-Y. MBEToolbox: a Matlab toolbox for sequence data analysis in molecular biology and evolution. *BMC Bioinformatics 6*, 1 (2005), 64.

[23] CHO, C.-K. J., DRABOVICH, A. P., BATRUCH, I., AND DIAMANDIS, E. P. Verification of a biomarker discovery approach for detection of Down syndrome in amniotic fluid via multiplex selected reaction monitoring (SRM) assay. *J Proteomics* (2011), 2052–2059.

[24] C.M., B., AND G., H. *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

[25] DASH, M., AND LIU, H. Feature selection for classification. *Intelligent Data Analysis 1*, 104 (1997), 131–156.

[26] DAVIS, R. A., CHARLTON, A. J., OEHLSCHLAGER, S., AND WILSON, J. C. Novel feature selection method for genetic programming using metabolomic 1H NMR data. *Chemometrics and Intelligent Laboratory Systems 81*, 1 (2006), 50–59.

[27] DEB, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, San Diego, CA, US 6* (2000), 182–197.

[28] DEFOIN PLATEL, M., VÉREL, S., CLERGUE, M., AND CHAMI, M. Density Estimation with Genetic Programming for Inverse Problem Solving. In *Genetic Programming*, vol. 4445. 2007, pp. 45–54.

[29] DEKKER, L. J., BOOGERD, W., STOCKHAMMER, G., DALEBOUT, J. C., SICCAMA, I., ZHENG, P., BONFRER, J. M., VERSCHUUREN, J. J., JENSTER, G., VERBEEK, M. M., LUIDER, T. M., AND SMITT, P. A. S. MALDI-TOF Mass Spectrometry Analysis of Cerebrospinal

Fluid Tryptic Peptide Profiles to Diagnose Leptomeningeal Metastases in Patients with Breast Cancer. *Molecular & Cellular Proteomics 4*, 9 (2005), 1341–1349.

[30] DESIERE, F., DEUTSCH, E. W., KING, N. L., NESVIZHSKII, A. I., MALLICK, P., ENG, J., CHEN, S., EDDES, J., LOEVENICH, S. N., AND AEBERSOLD, R. The PeptideAtlas project. *Nucleic Acids Research 34*, suppl 1 (2006), 655–658.

[31] DESSÌ, N., AND PES, B. Stability in Biomarker Discovery: Does Ensemble Feature Selection Really Help? In *Current Approaches in Applied Artificial Intelligence*, M. Ali, Y. S. Kwon, C.-H. Lee, J. Kim, and Y. Kim, Eds., vol. 9101, (2015). pp. 191–200.

[32] DORIGO, M., AND DI CARO, G. Ant colony optimization: a new meta-heuristic. In *Proceedings of the IEEE Congress on Evolutionary Computation, Washington, DC, USA* (1999), vol. 2, pp. –1477.

[33] DORIGO, M., AND STÜTZLE, T. *Ant Colony Optimization*. A Bradford book. BRADFORD BOOK, 2004.

[34] DOUCETTE, J., AND HEYWOOD, M. GP Classification under Imbalanced Data sets: Active Sub-sampling and AUC Approximation. In *Genetic Programming*, vol. 4971. 2008, pp. 266–277.

[35] DRISCOLL, J. A., WORZEL, B., AND MACLEAN, D. Classification of Gene Expression Data with Genetic Programming. In *Genetic Programming Theory and Practice*. 2003, pp. 25–42.

[36] DUBOIS, F., KNOCHENMUSS, R., ZENOBI, R., BRUNELLE, A., DEPRUN, C., AND LE BEYEC, Y. A comparison between ion-to-photon and microchannel plate detectors. *Rapid Communications in Mass Spectrometry 13*, 9 (1999), 786–791.

[37] DUNN, W. Current trends and future requirements for the mass spectrometric investigation of microbial, mammalian and plant metabolomes. *Physical Biology 5* (2008), 011001. (24pp).

[38] EYERS, C. E., LAWLESS, C., WEDGE, D. C., LAU, K. W., GASKELL, S. J., AND HUBBARD, S. J. CONSeQuence: prediction of reference peptides for absolute quantitative proteomics using consensus machine learning approaches. *Molecular & Cellular Proteomics 10*, 11 (2011), M110–003384.

[39] FIRPI, H., GOODMAN, E., AND ECHAUZ, J. On Prediction of Epileptic Seizures by Computing Multiple Genetic Programming Artificial Features. In *Genetic Programming*, vol. 3447. 2005, pp. 321–330.

[40] FORMAN, G. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *J. Mach. Learn. Res. 3* (2003), 1289–1305.

[41] FRANK, E., AND WITTEN, I. H. Generating Accurate Rule Sets Without Global Optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning, Wisconsin, USA* (1998), pp. 144–151.

[42] FRANKEN, H., LEHMANN, R., HÄRING, H.-U., FRITSCHE, A., STEFAN, N., AND ZELL, A. Wrapper- and Ensemble-Based Feature Subset Selection Methods for Biomarker Discovery in Targeted Metabolomics. In *Pattern Recognition in Bioinformatics*, vol. 7036. 2011, pp. 121–132.

[43] FREUND, Y., AND SCHAPIRE, R. E. Large Margin Classification Using the Perceptron Algorithm. *Mach. Learn. 37*, 3 (1999), 277–296.

[44] FU, W., JOHNSTON, M., AND ZHANG, M. Automatic Construction of Invariant Features Using Genetic Programming for Edge Detec-

tion. In *Proceedings of 25th Australasian Conference on Artificial Intelligence, Sydney, Australia* (2012), pp. 144–155.

[45] FU, W., JOHNSTON, M., AND ZHANG, M. Genetic Programming for Automatic Construction of Variant Features in Edge Detection. In *Proceedings of the 16th European Conference on the Applications of Evolutionary Computation, EvoApplications, Vienna, Austria* (2013), pp. 354–364.

[46] FUSARO VINCENT A, MANI D R, MESIROV JILL P, AND CARR STEVEN A. Prediction of high-responding peptides for targeted protein assays by mass spectrometry. *Nat Biotech 27*, 2 (2009), 190–198.

[47] GARCIA-TORRES, M., ARMANANZAS, R., BIELZA, C., AND LARRANAGA, P. Comparison of metaheuristic strategies for peakbin selection in proteomic mass spectrometry data. *Information Sciences 222*, 0 (2013), 229–246.

[48] GAY, S., BINZ, P.-A., HOCHSTRASSER, D. F., AND APPEL, R. D. Peptide mass fingerprinting peak intensity prediction: Extracting knowledge from spectra. *PROTEOMICS 2*, 10 (2002), 1374–1391.

[49] GE, G., AND WONG, G. W. Classification of premalignant pancreatic cancer mass-spectrometry data using decision tree ensembles. *BMC Bioinformatics 9*, 1 (2008), 275.

[50] GHADERI, S., KULKARNI, P. S., LEDFORD, E. B., WILKINS, C. L., AND GROSS, M. L. Chemical ionization in Fourier transform mass spectrometry. *Analytical Chemistry 53*, 3 (1981), 428–437.

[51] GILBERT, R. J., GOODACRE, R., WOODWARD, A. M., AND KELL, D. B. Genetic Programming: A Novel Method for the Quantitative Analysis of Pyrolysis Mass Spectral Data. *Analytical Chemistry 69*, 21 (1997), 4381–4389.

[52] GILBERT, R. J., JOHNSON, H. E., WINSON, M. K., ROWLAND, J. J., GOODACRE, R., SMITH, A. R., HALL, M. A., AND KELL, D. B. Genetic Programming as an Analytical Tool for Metabolome Data, (1999). 23–33.

[53] GILBERT, R. J., ROWLAND, J. J., AND KELL, D. B. Genomic computing: explanatory modelling for functional genomics. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '00), Las Vegas, Nevada, USA* (2000), pp. 551–557.

[54] GLISH, G., MCLUCKEY, S., RIDLEY, T., AND COOKS, R. A new "hybrid" sector/quadrupole mass spectrometer for mass spectrometry/mass spectrometry. *International Journal of Mass Spectrometry and Ion Physics 41*, 3 (1982), 157–177.

[55] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[56] GORMEZ, Z., SEKER, H., AND SERTBAS, A. Hypertension prediction by multi-objective optimization methods. In *Proceedings of the 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, Turkey* (2014), pp. 882–885.

[57] GRANIZO-MACKENZIE, D., AND MOORE, J. H. Multiple Threshold Spatially Uniform ReliefF for the Genetic Analysis of Complex Human Diseases. In *The European Conference on Evolutionary Computation, Machine Learning and Data mining in computational Biology (EvoBIO), Vienna, Austria* (2013), pp. 1–10.

[58] GROSS, J. Introduction. In *Mass Spectrometry*. Springer Berlin Heidelberg, 2011, pp. 1–20.

[59] GROUP, B. D. W. Biomarkers and surrogate endpoints: Preferred definitions and conceptual framework. *Clinical Pharmacology & Therapeutics 69*, 3 (2001), 89–95.

[60] GUO, H., JACK, L., AND NANDI, A. Automated feature extraction using genetic programming for bearing condition monitoring. In *Proceedings of the 2004 14th IEEE Signal Processing Society Workshop* (2004), pp. 519–528.

[61] GUO, H., AND NANDI, A. Breast Cancer Diagnosis Using Genetic Programming Generated Feature. In *2005 IEEE Workshop on Machine Learning for Signal Processing* (2005), pp. 215–220.

[62] GUO, H., ZHANG, Q., AND NANDI, A. K. Feature extraction and dimensionality reduction by genetic programming based on the Fisher criterion. *Expert Systems 25*, 5 (2008), 444–459.

[63] GUYON, I., GUNN, S., NIKRAVESH, M., AND ZADEH, L. *Feature Extraction: Foundations and Applications*. Studies in Fuzziness and Soft Computing. Springer, 2006.

[64] GUYON, I., GUNN, S. R., BEN-HUR, A., AND DROR, G. Result Analysis of the NIPS 2003 Feature Selection Challenge. In *NIPS* (2004).

[65] GYGI STEVEN P., RIST BEATE, GERBER SCOTT A., TURECEK FRANTISEK, GELB MICHAEL H., AND AEBERSOLD RUEDI. Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nat Biotech 17*, 10 (1999), 994–999.

[66] HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. The WEKA data mining software: an update. *SIGKDD Explorations 11*, 1 (2009), 10–18.

[67] HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. The WEKA data mining software: an update. *SIGKDD Explorer Newsletter* (2009), 10–18.

[68] HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., AND WITTEN, I. H. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl. 11*, 1 (2009), 10–18.

[69] HAN, Y., MA, B., AND ZHANG, K. SPIDER: Software for Protein identification from Sequence Tags with De Novo Sequencing Error. *J Bioinform Comput Biol 3* (2004), 697–716.

[70] HE, H., AND GARCIA, E. A. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering 21*, 9 (2009), 1263–1284.

[71] HECKERMAN, D., GEIGER, D., AND CHICKERING, D. M. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. In *MACHINE LEARNING* (1995), pp. 197–243.

[72] HEIDI VÄHÄMAA, V. R. K. W. H. PolyAlign: A versatile LC-MS data alignment tool for landmark-selected and automated use. *International Journal of Proteomics* (2011), 1–10.

[73] HOEKMAN, B., BREITLING, R., SUITS, F., BISCHOFF, R., AND HORVATOVICH, P. msCompare: a framework for quantitative analysis of label-free LC-MS data for comparative candidate biomarker studies. *Mol Cell Proteomics 11*, 6 (2012), M111.015974.

[74] HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, USA, 1992.

[75] HOLTE, R. Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Machine Learning 11*, 1 (1993), 63–90.

[76] HONG, J.-H., AND CHO, S.-B. Lymphoma Cancer Classification Using Genetic Programming with SNR Features. In *Genetic Programming*, M. Keijzer, U.-M. O.Reilly, S. Lucas, E. Costa, and T. Soule, Eds., vol. 3003 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 78–88.

[77] HU, T., CHEN, Y. P., KIRALIS, J., COLLINS, R. L., WEJSE, C., SIRUGO, G., WILLIAMS, S. M., AND MOORE, J. H. Research and applications: An information-gain approach to detecting three-way epistatic interactions in genetic association studies. *JAMIA 20*, 4 (2013), 630–636.

[78] HUANG, J., CAI, Y., AND XU, X. A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters 28*, 13 (2007), 1825–1844.

[79] HUTTENHAIN, R., MALMSTROM, J., PICOTTI, P., AND AEBERSOLD, R. Perspectives of targeted mass spectrometry for protein biomarker verification. *Curr Opin Chem Biol 13* (2009), 518–525.

[80] KATAJAMAA, M., MIETTINEN, J., AND ORESIC, M. MZmine: Toolbox for processing and visualization of mass spectrometry based molecular profile data. *Bioinformatics (Oxford, England) 22* (2006), 634–636.

[81] KAWASHIMA, S., AND KANEHISA, M. AAindex: Amino Acid index database. *Nucleic Acids Research 28*, 1 (2000), 374.

[82] KELL, D. B. Metabolomics and Machine Learning: Explanatory Analysis of Complex Metabolome Data Using Genetic Programming to Produce Simple, Robust Rules. *Molecular Biology Reports 29*, 1-2 (2002), 237–241.

[83] KENNEDY, J., AND EBERHART, R. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks, Perth, Western Australia* (1995), vol. 4, pp. 1942–1948.

[84] KENNEDY, J., KENNEDY, J., EBERHART, R., AND SHI, Y. *Swarm Intelligence*. Evolutionary Computation Series. Morgan Kaufmann Publishers, 2001.

[85] KINNEAR, K., SPECTOR, L., AND ANGELINE, P. *Advances in Genetic Programming*. No. v. 3. MIT Press, 1999.

[86] KIRCHNER, M., RENARD, B. Y., K THE, U., PAPPIN, D. J., HAMPRECHT, F. A., STEEN, H., AND STEEN, J. A. J. Computational protein profile similarity screening for quantitative mass spectrometry experiments, *Bioinformatics*, (2010). pp. 77–83.

[87] KOENIG, T., MENZE, B. H., KIRCHNER, M., MONIGATTI, F., PARKER, K. C., PATTERSON, T., STEEN, J. J., HAMPRECHT, F. A., AND STEEN, H. Robust Prediction of the MASCOT Score for an Improved Quality Assessment in Mass Spectrometric Proteomics. *Journal of Proteome Research 7*, 9 (2008), 3708–3717.

[88] KOHAVI, R. The Power of Decision Tables. In *Proceedings of the European Conference on Machine Learning, Crete, Greece* (1995), Springer Verlag, pp. 174–189.

[89] KOHAVI, R. Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid. In *Proceedings of the Second International Conference on Knolwedge Discovery and Data Mining, Portland, Oregon* (1996), pp. 202–207.

[90] KOHLBACHER, O., REINERT, K., GROPL, C., LANGE, E., PFEIFER, N., SCHULZ-TRIEGLAFF, O., AND STURM, M. TOPP-the OpenMS proteomics pipeline. *Bioinformatics 23*, 2 (2007), 191–197.

[91] KOURID, A., AND BATOUCHE, M. Biomarker Discovery Based on Large-Scale Feature Selection and MapReduce. In *Computer Science and Its Applications, (2015)*, vol. 456. pp. 81–92.

[92] KOZA, J. *Genetic Programming III: Darwinian Invention and Problem Solving : John R.Koza...[et Al.]*. A Bradford book. Elsevier Science & Tech, 1999.

[93] KRAWIEC, K. Genetic Programming-based Construction of Features for Machine Learning and Knowledge Discovery Tasks. *Genetic Programming and Evolvable Machines 3*, 4 (2002), 329–343.

[94] KRAWIEC, K., AND BHANU, B. Visual learning by coevolutionary feature synthesis. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 35*, 3 (2005), 409–425.

[95] KUESTER, R. K., AND SIPES, I. G. Prediction of metabolic clearance of bisphenol using cryopreserved human hepatocytes. *Drug Metabolism and Disposition 35*, 10 (2007), 1910–1915.

[96] LANGDON, W. B., POLI, R., MCPHEE, N. F., AND KOZA, J. R. Genetic Programming: An Introduction and Tutorial, with a Survey of Techniques and Applications. In *Computational Intelligence: A Compendium*. 2008, pp. 927–1028.

[97] LANGE, E. *Analysis of Mass Spectrometric Data: Peak Picking and Map Alignment*. 2008.

[98] LANGE, E., GRÖPL, C., SCHULZ-TRIEGLAFF, O., LEINENBACH, A., HUBER, C. G., AND REINERT, K. A geometric approach for the alignment of liquid chromatography-mass spectrometry data. *Bioinformatics 23*, 13 (2007), 273–281.

[99] LANGE, E., TAUTENHAHN, R., NEUMANN, S., AND GROPL, C. Critical assessment of alignment procedures for LC-MS proteomics

and metabolomics measurements. *BMC Bioinformatics 9*, 1 (2008), 375–394.

[100] LI, L., TANG, H., WU, Z., GONG, J., GRUIDL, M., ZOU, J., TOCKMAN, M., AND CLARK, R. A. Data mining techniques for cancer detection using serum proteomic profiling. *Artificial Intelligence in Medicine 32*, 2 (2004), 71–83.

[101] LI, X., YI, E., KEMP, C., ZHANG, H., AND AEBERSOLD, R. A software suite for the generation and comparison of peptide arrays from sets of data collected by Liquid Chromatography-Mass Spectrometry. *Molecular & cellular proteomics : MCP 4*, 9 (2005), 1328–1340.

[102] LI, Y., LIU, Y., AND BAI, L. Genetic algorithm based feature selection for mass spectrometry data. In *BioInformatics and BioEngineering, 2008. BIBE 2008. 8th IEEE International Conference on* (2008), pp. 1–6.

[103] LISTGARTEN, J., AND EMILI, A. Statistical and computational methods for comparative proteomic profiling using liquid chromatography-tandem mass spectrometry. *Mol. Cell. Proteomics 4* (2005), 419–434.

[104] LISTGARTEN, J., NEAL, R. M., ROWEIS, S. T., WONG, P., AND EMILI, A. Difference detection in LC-MS data for protein biomarker discovery. *Bioinformatics 23*, 2 (2007), 198–204.

[105] LIU, C., SONG, Y., YAN, B., XU, Y., AND CAI, L. Fast De novo Peptide Sequencing and Spectral Alignment via Tree Decomposition. In *Pacific Symposium on Biocomputing'06* (2006), pp. 255–266.

[106] LIU, H., AND MOTODA, H. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers, 1998.

[107] LIU, Q., QIAO, M., AND SUNG, A. H. Distance metric learning and support vector machines for classification of mass spectrometry proteomics data, (2009). pp. 216–226.

[108] LUKE, S. *Essentials of Metaheuristics*, second ed. Lulu, 2013.

[109] MAIMON, O., AND ROKACH, L. *Data Mining and Knowledge Discovery Handbook*. Series in Solid-State Sciences. Springer, 2010.

[110] MALLEY, J. D., DASGUPTA, A., AND MOORE, J. H. The limits of p-values for biological data mining. *BioData Mining 6* (2013), 10.

[111] MALLICK, P., SCHIRLE, M., CHEN, S., FLORY, M., LEE, H., MARTIN, D., RANISH, J., RAUGHT, B., SCHMITT, R., WERNER, T., KUSTER, B., AND AEBERSOLD, R. Computational Prediction of Proteotypic Peptides for Quantitative Proteomics. *Nat Biotechnol 25*, 1 (2007), 125–31.

[112] MARZILLI, L. A., KOERTJE, C., AND VOUROS, P. Capillary Electrophoresis-Mass Spectrometric Analysis of DNA Adducts, Methods in Molecular Biology, (2000), pp. 395-406.

[113] MAST, S., PENG, L., JORDAN, T. W., FLINT, H., PHILLIPS, L., DONALDSON, L., STRABALA, T. J., AND WAGNER, A. Proteomic analysis of membrane preparations from developing Pinus radiata compression wood. *Tree Physiology 30*, 11 (2010), 1456–1468.

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

[115] MENG, Y. A swarm intelligence based algorithm for proteomic pattern detection of ovarian cancer. In *IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, Toronto, Canada* (2006), pp. 1–7.

[116] MEYDAN, C., KÜÇÜKURAL, A., YÖRÜKOĞLU, D., AND SEZERMAN, O. Discovery of Biomarkers for Hexachlorobenzene Toxicity Using Population Based Methods on Gene Expression Data. In *Pattern Recognition in Bioinformatics*, vol. 5265, (2008), 412-423.

[117] MILLER, A. *Subset Selection in Regression, Second Editon*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Taylor & Francis Group, 2002.

[118] MITCHELL, M. *An Introduction to Genetic Algorithms*. A Bradford book. Bradford Books, 1998.

[119] MITCHELL, T. M. *Machine Learning*. McGraw-Hill, New York, 1997.

[120] MORRIS, J. S., COOMBES, K. R., KOOMEN, J., BAGGERLY, K. A., AND KOBAYASHI, R. Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum. *Bioinformatics 21*, 9 (2005), 1764–1775.

[121] MUKHOPADHYAY, A., AND MANDAL, M. A Hybrid Multiobjective Particle Swarm Optimization Approach for Non-redundant Gene Marker Selection. In *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA),Gwalior, India, (2012)*, vol. 201. pp. 205–216.

[122] NESHATIAN, K., AND ZHANG, M. Unsupervised Elimination of Redundant Features Using Genetic Programming. In *Proceeding of the 22nd Australasian Conference on Artificial Intelligence, Melbourne, Australia* (2009), pp. 432–442.

[123] NESHATIAN, K., ZHANG, M., AND ANDREAE, P. Genetic Programming for Feature Ranking in Classification Problems. In *Proceedings of the 7th International Conference of Simulated Evoltion and Learning (SEAL), Melbourne, Australia* (2008), pp. 544–554.

[124] NESHATIAN, K., ZHANG, M., AND JOHNSTON, M. Feature Construction and Dimension Reduction Using Genetic Programming. In *Proceeding of 20th Australian Conference on Artificial Intelligence, Gold Coast, Australia* (2007), pp. 160–170.

[125] NG, T. A review of research on the protein-bound polysaccharide (polysaccharopeptide PSP) from the mushroom Coriolus versicolor (basidiomycetes: Polyporaceae). *General Pharmacology: The Vascular System 30*, 1 (1998), 1–4.

[126] NGATCHOU, P., ZAREI, A., AND EL-SHARKAWI, M. Pareto Multi Objective Optimization. In *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems* (2005), pp. 84–91.

[127] ONG, S.-E., BLAGOEV, B., KRATCHMAROVA, I., KRISTENSEN, D. B., STEEN, H., PANDEY, A., AND MANN, M. Stable Isotope Labeling by Amino Acids in Cell Culture, SILAC, as a Simple and Accurate Approach to Expression Proteomics. *Molecular and Cellular Proteomics 1*, 5 (2002), 376–386.

[128] OOI, C. H., AND TAN, P. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. *Bioinformatics 19*, 1 (2003), 37–44.

[129] ORRIOLS, A., AND BERNADO-MANSILLA, E. Class imbalance problem in UCS classifier system: fitness adaptation. In *IEEE Congress on Evolutionary Computation, Edinburgh, UK* (2005), vol. 1, pp. 604–611.

[130] PALMBLAD, M., MILLS, D. J., BINDSCHEDLER, L. V., AND CRAMER, R. Chromatographic Alignment of LC-MS and LC-MS/MS Datasets by Genetic Algorithm Feature Extraction. *Journal of the American Society for Mass Spectrometry 18*, 10 (2007), 1835–1843.

[131] PASZKOWICZ, W. Genetic Algorithms, a Nature-Inspired Tool: Survey of Applications in Materials Science and Related Fields. *Materials and Manufacturing Processes 24*, 2 (2009), 174–197.

[132] PENG, Y., WU, Z., AND JIANG, J. A novel feature selection approach for biomedical data classification. *J. of Biomedical Informatics 43*, 1 (2010), 15–23.

[133] PETRICOIN, ARDEKANI, A. M., HITT, B. A., LEVINE, P. J., FUSARO, V. A., STEINBERG, S. M., MILLS, G. B., SIMONE, C., FISHMAN, D. A., KOHN, E. C., AND LIOTTA, L. A. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet 359* (2002), 572–577.

[134] PETRICOIN, E. F., RAJAPASKE, V., HERMAN, E. H., AREKANI, A. M., ROSS, S., JOHANN, D., KNAPTON, A., ZHANG, J., HITT, B. A., CONRADS, T. P., VEENSTRA, T. D., LIOTTA, L. A., AND SISTARE, F. D. Toxicoproteomics: Serum Proteomic Pattern Diagnostics for Early Detection of Drug Induced Cardiac Toxicities and Cardioprotection. *Toxicologic Pathology* (2004), 122–130.

[135] PLUSKAL, T., CASTILLO, S., VILLAR-BRIONES, A., AND ORESIC, M. MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics 11* (2010), 395.

[136] POLI, R., LANGDON, W., AND MCPHEE, N. *A Field Guide to Genetic Programming*. Lulu.com, 2008.

[137] POPOVIC, D., SIFRIM, A., PAVLOPOULOS, G., MOREAU, Y., AND DE MOOR, B. A Simple Genetic Algorithm for Biomarker Mining. In *Pattern Recognition in Bioinformatics*, vol. 7632, (2012). pp. 222–232.

[138] PRINCE, J., CARLSON, M., LU, R., AND MARCOTTE, E. The need for a public proteomics repository. *Nat Biotechnol 22* (2004), 471–472.

[139] PRUITT, K. D., TATUSOVA, T., AND MAGLOTT, D. R. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research 33*, suppl 1 (2005), 501–504.

[140] RESSOM, H., VARGHESE, R. S., ORVISKY, E., DRAKE, S., HORTIN, G., ABDEL-HAMID, M., LOFFREDO, C. A., AND GOLDMAN, R. 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* (2006), pp. 4560–4563.

[141] RESSOM, H., VARGHESE, R. S., SAHA, D., ORVISKY, E., GOLDMAN, L., PETRICOIN, E. F., CONRADS, T. P., VEENSTRA, T. D., ABDEL-HAMID, M., LOFFREDO, C. A., AND GOLDMAN, R. Particle swarm optimization for analysis of mass spectral serum profiles. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO), Washington, DC, USA , (2005), pp.431–438*.

[142] RESSOM, H. W., VARGHESE, R. S., GOLDMAN, L., AN, Y., LOFFREDO, C. A., ABDEL-HAMID, M., KYSELOVA, Z., MECHREF, Y., NOVOTNY, M., DRAKE, S. K., AND GOLDMAN, R. Analysis of MALDI-TOF mass spectrometry data for discovery of peptide and glycan biomarkers of hepatocellular carcinoma. *Journal of Proteome Research 7*, 2 (2008), 603–610.

[143] RODRÍGUEZ-VÁZQUEZ, K., AND OLIVER-MORALES, C. Multi-branches Genetic Programming as a Tool for Function Approximation. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO),Seattle, Washington, USA* (2004), pp. 719–721.

[144] RU, Q. C., ZHU, L. A., SILBERMAN, J., AND SHRIVER, C. D. Label-free Semiquantitative Peptide Feature Profiling of Human Breast Cancer and Breast Disease Sera via Two-dimensional Liquid Chromatography-Mass Spectrometry. *Molecular and Cellular Proteomics 5*, 6 (2006), 1095–1104.

[145] SALMI, J., MOULDER, R., FIL, J.-J., NEVALAINEN, O. S., NYMAN, T. A., LAHESMAA, R., AND AITTOKALLIO, T. BIOINFORMAT-

ICS Quality classification of tandem mass spectrometry data, 2008. pp. 400–406.

[146] SANDERS, W., BRIDGES, S., MCCARTHY, F., NANDURI, B., AND BURGESS, S. Prediction of peptides observable by mass spectrometry applied at the experimental set level. *BMC Bioinformatics 8*, Suppl 7 (2007), S23.

[147] SANDIN, I., ANDRADE, G., VIEGAS, F., MADEIRA, D., DA ROCHA, L. C., SALLES, T., AND GONÇALVES, M. A. Aggressive and effective feature selection using genetic programming. In *IEEE Congress on Evolutionary Computation, Brisbane, Australia* (2012), IEEE, pp. 1–8.

[148] SATTEN, G. A., DATTA, S., MOURA, H., WOOLFITT, A. R., CARVALHO, M. D. G., CARLONE, G. M., DE, B. K., PAVLOPOULOS, A., AND BARR, J. R. Standardization and denoising algorithms for mass spectra to classify whole-organism bacterial specimens. *Bioinformatics 20*, 17 (2004), 3128–3136.

[149] SCHULZ-TRIEGLAFF, O. *Computational Methods for Quantitative Peptide Mass Spectrometry*. 2008.

[150] SEBASTIANI, F., AND RICERCHE, C. N. D. Machine Learning in Automated Text Categorization. *ACM Computing Surveys 34* (2002), 1–47.

[151] SHI, Y., AND EBERHART, R. A modified particle swarm optimizer. In *Evolutionary Computation Proceedings of IEEE World Congress on Computational Intelligence.* (1998), pp. 69–73.

[152] SMITH, C., WANT, E., O'MAILLE, G., ABAGYAN, R., AND SIUZDAK, G. XCMS: Processing mass spectrometry data for metabolite profiling using Nonlinear peak alignment, matching, and identification. *Analytical Chemistry* (2006), 779–787.

[153] SMITH, M., AND BULL, L. Feature Construction and Selection Using Genetic Programming and a Genetic Algorithm. In *Genetic Programming*, vol. 2610. 2003, pp. 229–237.

[154] SOHA AHMED AND MENGJIE ZHANG AND LIFENG PENG. Feature Selection and Classification of High Dimensional Mass Spectrometry Data: A Genetic Programming Approach. In *Proceedings of 11th European Conference on Machine Learning and Data Mining in Bioinformatics, Vienna, Austria* . 2013, pp. 43–55.

[155] SONG, D., HEYWOOD, M., AND ZINCIR-HEYWOOD, A. Training genetic programming on half a million patterns: an example from anomaly detection. *IEEE Transactions on Evolutionary Computation 9*, 3 (2005), 225–239.

[156] SORACE, J., AND ZHAN, M. A data review and re-assessment of ovarian cancer serum proteomic profiling. *BMC Bioinformatics 4*, 1 (2003), 24.

[157] SOYEL, H., TEKGUC, U., AND DEMIREL, H. Application of NSGA-II to feature selection for facial expression recognition. *Computers & Electrical Engineering 37*, 6 (2011), 1232–1240.

[158] STRIMBU KYLE, AND TAVEL JORGE A. What are Biomarkers? *Current opinion in HIV and AIDS 5*, 6 (2010), 463–466.

[159] SUGANTHAN, P. Structural pattern recognition using genetic algorithms. *Pattern Recognition 35*, 9 (2002), 1883–1893.

[160] SUN, Y., WONG, A. K. C., AND KAMEL, M. S. CLASSIFICATION OF IMBALANCED DATA: A REVIEW. *International Journal of Pattern Recognition and Artificial Intelligence 23*, 04 (2009), 687–719.

[161] SUN, Y., AND WU, D. A RELIEF Based Feature Extraction Algorithm. In *SDM* (2008), pp. 188–195.

[162] TAN, N. C., FISHER, W. G., ROSENBLATT, K. P., AND GARNER, H. R. Application of multiple statistical tests to enhance mass spectrometry-based biomarker discovery. *BMC Bioinformatics 10* (2009), 144.

[163] TANAKA, K., WAKI, H., IDO, Y., AKITA, S., YOSHIDA, Y., YOSHIDA, T., AND MATSUO, T. Protein and polymer analyses up to m/z 100 000 by laser ionization time-of-flight mass spectrometry. *Rapid Communications in Mass Spectrometry 2*, 8 (1988), 151–153.

[164] TANG, H., ARNOLD, R. J., ALVES, P., XUN, Z., CLEMMER, D. E., NOVOTNY, M. V., REILLY, J. P., AND RADIVOJAC, P. A computational approach toward label-free protein quantification using predicted peptide detectability. *Bioinformatics 22*, 14 (2006), 481–488.

[165] TANG, N., TORNATORE, P., AND WEINBERGER, S. R. Current developments in SELDI affinity technology. *Mass Spectrometry Reviews 23*, 1 (2004), 34–44.

[166] TAYLOR, J. A., AND JOHNSON, R. S. Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry 11*, 9 (1997), 1067–1075.

[167] TIMM, W. *Peak intensity prediction in mass spectra using machine learning methods*. 2008.

[168] TRIMPIN, S., INUTAN, E., HERATH, T., AND MCEWEN, C. Matrix-assisted laser desorption/ionization mass spectrometry method for selectively producing either singly or multiply charged molecular ions. *Analytical Chemistry 82*, 5 (2010), 11–5.

[169] TULI, L., TSAI, T.-H., VARGHESE, R., XIAO, J. F., CHEEMA, A., AND RESSOM, H. Using a spike-in experiment to evaluate analysis of LC-MS data. *Proteome Science item.volume* (2012), 1–13.

[170] VAIDYANATHAN, S., BROADHURST, D. I., KELL, D. B., AND GOODACRE, R. Explanatory Optimization of Protein Mass Spectrometry via Genetic Search. *Analytical Chemistry 75*, 23 (2003), 6679–6686.

[171] VANDENBOGAERT, M., LI-THIAO-TE, S., KALTENBACH, H., ZHANG, R., AITTOKALLIO, T., AND SCHWIKOWSKI, B. Alignment of LC-MS images, with applications to biomarker discovery and protein identification. *Proteomics 8*, 4 (2008), 650–672.

[172] VAPNIK, V. N. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[173] VILLMANN, T., SCHLEIF, F.-M., KOSTRZEWA, M., WALCH, A., AND HAMMER, B. Classification of mass-spectrometric data in clinical proteomics using learning vector quantization methods. *Briefings in Bioinformatics 9*, 2 (2008), 129–143.

[174] VOSS, B., HANSELMANN, M., RENARD, B., LINDNER, M., KÖTHE, U., KIRCHNER, M., AND HAMPRECHT, F. SIMA: simultaneous multiple alignment of LC/MS peak lists. *Bioinformatics 27*, 7 (2011), 87–93.

[175] WEDGE, D. C., GASKELL, S. J., HUBBARD, S. J., KELL, D. B., LAU, K. W., AND EYERS, C. Peptide detectability following ESI mass spectrometry: prediction using genetic programming. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, London, England* (2007), vol. 2, pp. 2219–2225.

[176] WHITE, D. R. Software review: the ECJ toolkit, , (2012). pp. 65–67.

[177] WILLIAMS, B., CORNETT, S., CRECELIUS, A., CAPRIOLI, R., DAWANT, B., AND BODENHEIMER, B. An algorithm for baseline

correction of MALDI mass spectra, Proceedings of the 43rd Annual Southeast Regional Conference,Kennesaw, GA, USA , 137–142, (2008).

[178] WINKLER, S., AFFENZELLER, M., AND WAGNER, S. Advanced Genetic Programming Based Machine Learning. *Journal of Mathematical Modelling and Algorithms 6, (2007)*, 3, 455–480.

[179] WISHART, D. S., TZUR, D., KNOX, C., EISNER, R., GUO, A. C., YOUNG, N., CHENG, D., JEWELL, K., ARNDT, D., SAWHNEY, S., FUNG, C., NIKOLAI, L., LEWIS, M., COUTOULY, M.-A., FORSYTHE, I., TANG, P., SHRIVASTAVA, S., JERONCIC, K., STOTHARD, P., AMEGBEY, G., BLOCK, D., HAU, D. D., WAGNER, J., MINIACI, J., CLEMENTS, M., GEBREMEDHIN, M., GUO, N., ZHANG, Y., DUGGAN, G. E., MACINNIS, G. D., WELJIE, A. M., DOWLATABADI, R., BAMFORTH, F., CLIVE, D., GREINER, R., LI, L., MARRIE, T., SYKES, B. D., VOGEL, H. J., AND QUERENGESSER, L. HMDB: the human metabolome database. *Nucleic Acids Research 35* (2007), 521–526.

[180] WITTEN, I. H., AND FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[181] WORZEL, W. P., YU, J., ALMAL, A. A., AND CHINNAIYAN, A. M. Applications of genetic programming in cancer research. *The International Journal of Biochemistry & Cell Biology 41*, 2 (2009), 405–413.

[182] WU, B., ABBOTT, T., FISHMAN, D., MCMURRAY, W., MOR, G., STONE, K., WARD, D., WILLIAMS, K., AND ZHAO, H. Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics 19*, 13 (2003), 1636–1643.

[183] XUE, B., CERVANTE, L., SHANG, L., BROWNE, W. N., AND ZHANG, M. Multi-objective evolutionary algorithms for filter based feature selection in classification. *International Journal on Artificial Intelligence Tools 22*, 4,(2013), 31.

[184] XUE, B., CERVANTE, L., SHANG, L., BROWNE, W. N., AND ZHANG, M. Binary PSO and rough set theory for feature selection: a multi-objective filter based approach. *International Journal of Computational Intelligence and Applications 13*, 2 (2014), 34.

[185] YAO, X., FREAS, A., RAMIREZ, J., DEMIREV, P. A., AND FENSELAU, C. Proteolytic 18O Labeling for Comparative Proteomics:Model Studies with Two Serotypes of Adenovirus. *Analytical Chemistry 73*, 13 (2001), 2836–2842.

[186] YOST, R. A., AND ENKE, C. G. Selected ion fragmentation with a tandem quadrupole mass spectrometer. *Journal of the American Chemical Society 100*, 7 (1978), 2274–2275.

[187] YU, JIANJUN, YU, JINDAN, ALMAL, ARPIT, A., DHANASEKARAN, SARAVANA, M., GHOSH, DEBASHIS, WORZEL, WILLIAM, P., CHINNAIYAN, AND ARUL, M. Feature Selection and Molecular Classification of Cancer Using Genetic Programming. *Neoplasia 9*, 4 (2007), 292–303.

[188] ZHANG, J., JEREMIAH, B., LINGYAN, L., SIWEI, W., NAGANA, G. G. A., ZANE, H., AND DANIEL, R. Esophageal Cancer Metabolite Biomarkers Detected by LC-MS and NMR Methods. *PLoS ONE 7*, 1 (01 2012), 30181.

[189] ZHANG, X., ASARA, J., ADAMEC, J., OUZZANI, M., AND ELMAGARMID, A. Data pre-processing in liquid chromatography/mass spectrometry-based proteomics. *Bioinformatics (Oxford, England) 21*, 21 (2005), 4054–4059.

[190] ZHANG, Y., AND ZHANG, M. A Multiple-Output Program Tree Structure in Genetic Programming. In *Proceedings of The Second Asian-Pacific Workshop on Genetic Programming* (Cairns, Australia, 2004), pp. 1–12.

[191] ZHOU, C., BOWLER, L., AND FENG, J. A machine learning approach to explore the spectra intensity pattern of peptides using tandem mass spectrometry data. *BMC Bioinformatics 9* (2008), 1–17.

[192] ZITZLER, E., LAUMANNS, M., AND THIELE, L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Evolutionary Methods for Design, Optimisation, and Control* (2002), CIMNE, Barcelona, Spain, pp. 95–100.

[193] ZITZLER, E., AND THIELE, L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *Trans. Evol. Comp 3*, 4 (1999), 257–271.