Evolutionary Algorithms for Improving *De Novo* Peptide Sequencing

by

Samaneh Azari

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
2020

Abstract

De novo peptide sequencing algorithms have been developed for peptide identification in proteomics from tandem mass spectra (MS/MS), which can be used to identify and discover novel peptides and proteins that do not have a database available.

Despite improvements in MS instrumentation and *de novo* sequencing methods, a significant number of CID MS/MS spectra still remain unassigned with the current algorithms, often leading to low confidence of peptide assignments to the spectra. Moreover, current algorithms often fail to construct the completely matched sequences, and produce partial matches. Therefore, identification of *full-length* peptides remains challenging. Another major challenge is the existence of noise in MS/MS spectra which makes the data highly imbalanced. Also missing peaks, caused by incomplete MS fragmentation makes it more difficult to infer a full-length peptide sequence. In addition, the large search space of all possible amino acid sequences for each spectrum leads to a high false discovery rate.

This thesis focuses on improving the performance of current methods by developing new algorithms corresponding to three steps of preprocessing, sequence optimisation and post-processing using machine learning for more comprehensive interrogation of MS/MS datasets. From the machine learning point of view, the three steps can be addressed by solving different tasks such as classification, optimisation, and symbolic regression. Since *Evolutionary Algorithms* (EAs), as effective global search techniques, have shown promising results in solving these problems, this thesis investigates the capability of EAs in improving the *de novo* peptide sequencing.

In preprocessing step, this thesis proposes an effective GP-based method for classification of signal and noise peaks in highly imbalanced MS/MS spectra with the purpose of having a positive influence on the reliability of the peptide identification. The results show that the proposed algorithm is the most stable classification method across various noise ratios, outperforming six other benchmark classification algorithms. The experimental results show a significant improvement in high confidence peptide assignments to MS/MS spectra when the data is preprocessed by the proposed GP method. Moreover, the first multi-objective GP approach for classification of peaks in MS/MS data, aiming at maximising the accuracy of the minority class (signal peaks) and the accuracy of the majority class (noise peaks) is also proposed in this thesis. The results show that the multi-objective GP method outperforms the single objective GP algorithm and a popular multi-objective approach in terms of retaining more signal peaks and removing more noise peaks. The multi-objective GP approach significantly improved the reliability of peptide identification.

This thesis proposes a GA-based method to solve the complex optimisation task of *de novo* peptide sequencing, aiming at constructing full-length sequences. The proposed GA method benefits the GA capability of searching a large search space of potential amino acid sequences to find the most likely full-length sequence. The experimental results show that the proposed method outperforms the most commonly used *de novo* sequencing method at both amino acid level and peptide level.

This thesis also proposes a novel method for re-scoring and re-ranking the peptide spectrum matches (PSMs) from the result of *de novo* peptide sequencing, aiming at minimising the false discovery rate as a post-processing approach. The proposed GP method evolves the computer programs to perform regression and classification simultaneously in order to generate an effective scoring function for finding the correct PSMs from many incorrect ones. The results show that the new GP-based PSM scoring function significantly improves the identification of full-length peptides when it is used to post-process the *de novo* sequencing results.

Dedication

To my mother who taught me to trust myself and chase my dreams.

Acknowledgments

The four years of my PhD was a journey full of joy, tears & tales and I'd like to say thank you to all those who supported me in this journey.

Firstly, I'd like to express my deep sense of gratitude to my supervisors Prof. Mengjie Zhang, Assoc. Prof. Bing Xue, and Dr. Lifeng Peng for helping me to make my dream a reality. I have been extremely lucky to have three supervisors who cared so much about my progress, my deadlines and who responded to my queries so promptly at any hour. They have dedicated their time and effort to assist me in improving my research skills and constantly provided constructive feedback on my research papers and thesis. Finishing this journey wouldn't have been possible without their insightful advice and encouragement.

I also would like to thank my examiners Dr. Yi Mei, Dr. Michael Mayo, and Dr.Eugene Kapp for the time that they spent reading my thesis and for letting me experience a wonderful oral examination via video conference during the coronavirus lockdown.

I'm so grateful to the Doctoral scholarship of Victoria University of Wellington which provided me financial support for over three years. Special thanks to the staff members, especially Patricia Stein for being such a lovely student adviser and to the ECS staff particularly Diana, Suzan, Monoa, Tony, Roger, and Mark for all your efforts when we moved to Maru, my favorite workplace.

Masood, Andrew, Deepak, Harith, Qi, Yuyu, Bach, and Binh, thank you for inspiration, motivation and sharing your stories about your PhD journey with me. Special thanks to Simonette, Monique, Mazhar, Mahdi, Shima, Ke,

Fangfang, and Anne for making those long nights of submission deadlines more enjoyable with lots of chit-chats, laughters and food.

I also greatly appreciate the support of my employer, Asa Cox, for letting me have flexible working hours and putting my PhD on priority when I needed it.

Heartfelt thanks to my family and friends. To my lovely mum and my brother (Reza) for their unconditional love and encouragement from thousands of miles away.

To my partner's parents, Sally and Patrick, for their love, care and all those delicious dinners every Wednesday night.

To Molly, for being a true friend and constantly reminding me that there is light at the end of the tunnel!

To my best friends in Cyprus, Sahar and Niloofar, for always listening to me, encouraging me and making me laugh.

And finally, to Benedict, for your patience, love and support over the last three years of the most challenging time of my life. Thanks for being so amazing and for being the reason I stay in beautiful Wellington. Now we can enjoy long weekend walks along the harbour without being worried about a PhD!

List of Publications

- Samaneh Azari, Bing Xue, Mengjie Zhang, Lifeng Peng. "GP-PostNovo: A Post-processing Method for Improving the Results of *de novo* Peptide Sequencing Using Genetic Programming". Submitted to IEEE Transactions on Cybernetics. Under review.

- Samaneh Azari, Bing Xue, Mengjie Zhang, Lifeng Peng. "A Decomposition Based Multi-objective Genetic Programming Algorithm for Classification of Highly Imbalanced Tandem Mass Spectrometry." The 5th Asian Conference on Pattern Recognition (ACPR 2019). Auckland, New Zealand. 26-29 November 2019. pp 449-463.

- Samaneh Azari, Bing Xue, Mengjie Zhang, and Lifeng Peng. Improving the results of *de novo* peptide identification via tandem mass spectrometry using a genetic programming-based scoring function for re-ranking peptide-spectrum matches. In Pacific Rim International Conference on Artificial Intelligence (PRICAI 2019), pages 474-487. Springer, 2019.

- Samaneh Azari, Bing Xue, Mengjie Zhang and Lifeng Peng. "Learning to Rank Peptide-Spectrum Matches Using Genetic Programming". Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC 2019). Wellington, New Zealand, 10-13 June, 2019. pp. 3244-3251.

- Samaneh Azari, Bing Xue, Mengjie Zhang, Lifeng Peng. "GA-Novo: De Novo Peptide Sequencing via Tandem Mass Spectrometry using Genetic Algorithm". Proceeding of the 22th European Conference on Applications of Evolutionary Computation (EvoApplications 2019). Lecture Notes in Computer Science. Vol. 11454, Leipzig, Germany. 24-26 April 2019. pp. 72-89.

- Samaneh Azari, Bing Xue, Mengjie Zhang, Lifeng Peng. “Preprocessing Tandem Mass Spectra Using Genetic Programming for Peptide Identification”,. Journal of The American Society for Mass Spectrometry. 2019. pp. 1294-1307, Vol. 30, No. 7, DOI: 10.1007/s13361-019-02196-5.

- Samaneh Azari, Mengjie Zhang, Bing Xue and Lifeng Peng. “Genetic Programming for Preprocessing Tandem Mass Spectra to Improve the Reliability of Peptide Identification.” IEEE World Congress on Computational Intelligence/ IEEE Congress on Evolutionary Computation (WCCI/CEC 2018). Rio de Janeiro, Brazil, 8-13 July, 2018. DOI: 10.1109/CEC.2018.8477810.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter provides an introduction to this thesis, starting with the problem statement, followed by the motivations section which outlines the main limitations of the existing methods and each part explains why *Evolutionary Algorithms* are suitable to deal with these limitations. The rest of this chapter presents the research goals, the major contributions and the organisation of the thesis.

1.1 Problem Statement

Proteomics is the large-scale identification and quantification of proteins in cells [1]. *Proteins* are the main players for cellular functions in the cell. They are micro-molecules made up of amino acids linked together in a linear sequence by peptide bonds. There are 20 common *amino acids* represented by the letters A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, and Y. *Peptides* are generally considered to be short chains of amino acids (from 2 to 50 amino acids). The chains that contain 50 to 2000 amino acid residues are commonly referred to as proteins. A protein can be fragmented into short peptide fragments by proteases.

One of the major challenges of proteomics research is to identify proteins and their post-translational modifications (PTMs) in cells that allows researchers discover possible biomarkers and mechanisms in relation to physiological disease or therapeutic state in an organism [2]. Mass spectrometry (MS) is currently the most commonly used technology in proteomics for identifying proteins in complex biological samples [3]. The common method for MS-based protein identification involves digesting proteins into peptides with enzymes such as trypsin that cleaves the protein at the known sites (trypsin cleaves the protein at 'K' and 'R'), which are then ionised, analysed and detected by mass spectrometers. The mass spectrometer measures the mass-to-charge (m/z) ratios of the precursor peptide ions (called MS spectra) and m/z ratios of the fragment ions of the precursor ions (called tandem mass spectra (MS/MS spectra)) [4]. While an MS spectrum is related to a protein, an MS/MS spectrum corresponds to a precursor peptide. These masses later can be used for identifying the peptides and proteins in the samples.

A key challenge in MS-based proteomics is assigning MS/MS spectra to the peptides, generating peptide-spectrum matches (PSMs). This process is called *peptide identification* [5]. Collision-induced dissociation (CID) is a commonly used technique in many mass spectrometers which fragments the peptides at the peptide bonds, producing b-/y-ions. The amino acid sequence of an CID MS/MS spectrum can be determined by the mass differences between b-ions or between y-ions.

Peptide identification can be performed using *database search* methods [6, 7, 8] where the input experimental MS/MS spectrum is compared with the theoretical spectrum predicted for each peptide sequence in a protein sequence database. However, these methods are only effective when the proteins of interest are already present in the reference protein database. *De novo peptide sequencing* algorithms are particularly appropriate for discovering novel peptides which are not presented in any protein sequence database. They are able to infer the amino acid sequence of a peptide without the assistance of a sequence database [9, 10, 11].

The term "*de novo*" in Latin means "starting from the beginning", and in *de novo* sequencing from MS/MS indicates starting from the beginning

(N-terminus) of the MS/MS spectrum and traversing the whole spectrum to the end (C-terminus). Given an MS/MS spectrum, the *de novo* sequencing algorithm starts with selecting a set of pairs of peaks from the spectrum. If the mass difference between each pair of peaks matches one of the amino acid residue masses, the mass difference will be labelled as the corresponding amino acid letter. The process continues until matching all mass differences with the amino acid residue masses. Then, the labelled amino acids of peak pairs are joined together to generate a candidate peptide. By doing so, the *de novo* sequencing algorithm generates a set of candidate peptides each having a confidence score reflecting the quality of match between the experimental spectrum and the candidate peptide. Normally, to find the best match to the experimental spectrum, the highest-scoring candidate PSM is returned as the results of *de novo* peptide sequencing for the input spectrum [9].

Although there have been many attempts to solve the *de novo* sequencing problem using different approaches, *de novo* sequencing is still not being widely used within the proteomics community compared to the database searching methods [12]. To some extent this shortcoming can be attributed to the limitations of the *de novo* sequencing algorithms. The main issue of *de novo* sequencing is that a large number of MS/MS spectra are unassigned with the existing *de novo* sequencing methods. That means that there is no high confidence peptide assignment to the spectrum. Moreover, the accuracy of *full-length* correct peptide sequencing by many existing *de novo* peptide sequencing methods can only reach 70% [11, 13, 14, 15]. It is crucial to identify the correct full-length peptides to avoid assigning a spectrum to a peptide which is not expressed by a genome [16]. Insufficient peptide level accuracy (i.e., the fraction of fully matched peptides) and lack of high confidence peptides at the amino acid level (i.e., partially matched peptides) result in low protein coverage and false identification. As peptide sequences are assembled to infer proteins in the sample, accurate peptide identification is essential for the correct protein identifications.

Fragmentation incompleteness and missing fragment ions prevent full-length peptide identification [13, 17]. In addition, an experimental MS/MS spectrum with hundreds of peaks normally contains significant background noise resulted from unexpected internal cleavages of the precursor peptide ions leading to non-interpretable peaks. The number of signal peaks is generally small compared with the noise peaks in proteomics data, which makes the MS/MS data highly imbalanced, raising the sensitivity and specificity issues which are the two conflicting objectives. The noise peaks in the spectra reduce the sensitivity of *de novo* peptide sequencing by introducing false identification of peptides.

Advanced computational methods using machine learning can be helpful to improve the performance of *de novo* sequencing [12]. The main focus of this research is improving the *de novo* sequencing results of the existing algorithms by utilising machine learning approaches. In machine learning, the major tasks in developing computational approaches to improve the *de novo* peptide sequencing involve: classification of the signal and noise peaks in MS/MS spectra in order to simplify the spectra; amino acid sequence optimisation in order to search for constructing the most likely amino acid sequence of a peptide; and modelling fragmentation patterns from multiple sources of information in order to discriminate between the true and false matches, controlling the false discovery rate.

Evolutionary computation (EC) is a family of population-based problem solving techniques that employs the principles based on the theory of biological evolution to get involved in many optimisation problems [18]. EC techniques have been successfully applied across a wide range of real-world problems in optimisation, classification, design and modelling [19]. An evolutionary algorithm (EA) as a subset of EC is a search technique which is based mainly on Darwinian principle of natural selection. EA uses a population of individuals to build a model, searching to find a good solution for the problem during the evolutionary process [20]. The goodness of individuals, which determines their potential to survive and represents their ability to solve the problem, is measured by a fitness function. The individuals can be modified by genetic operators to breed new individuals. An EA simulates evolution by employing fitness-based selection where the fitter individuals are expected to have a higher chance to be chosen for producing new individuals.

EAs do not require to make any prior assumption about the underlying fitness landscape. Moreover, although they do not need rich domain knowledge to use, they have the capability to incorporate the domain knowledge. Customisation is allowed in almost any aspect of these algorithms. In addition, they can be extended to solve multi-objective optimisation problems and find solutions to problems with multiple conflicting objectives. EAs are able to approximate a set of all Pareto optimal solutions of a multi-objective optimisation problem in a single run due to their population-based nature. Finding the non-dominated solutions along the trade-off surface allows a decision maker to choose a solution based on his/her preference.

EAs have been shown to be highly successful in many classification [21], regression[22, 23] and optimisation problems [24, 25]. This research focuses on utilising evolutionary algorithms for more comprehensive interrogation of MS/MS datasets in order to assist *de novo* sequencing for accurate peptide identification. An important goal of this research is to understand which EAs are most suited to solve the problems in peptide identification.

1.2 Motivations

1.2.1 Challenges of Preprocessing the MS/MS Spectra

The presence of noise in MS/MS spectra results in low confidence peptide assignment, which leads to low confidence protein identification and a risk of losing true identification. To overcome the problems imposed by the noise, a preprocessing step to denoise the MS/MS spectra in order to increase the overall confidence of peptide identification is required.

Intensity-based thresholding methods have been widely used for denoising the MS/MS spectra, however these methods only consider the intensity information of peaks and neglect the hidden interrelationship between them [8, 26, 27]. Recently, there is a growing trend to apply ML techniques on MS/MS data in order classify the signal and noise peaks [28, 29]. In these methods, the peaks of fragment ions are treated as “signals” and other peaks as noise.

However, working with imbalanced data is difficult as uneven distribution of class examples in the train dataset could leave the learning algorithm with a performance bias, resulting a high majority class accuracy and a poor performance on the minority class [30]. Another consideration is that as the ratio of imbalance varies between MS/MS datasets, the classification method should be robust to different imbalanced ratio. Therefore, a classification strategy which is able to maintain the performance trade-off between the minority (signal) and the majority (noise) class accuracies is required. So, it is worth investigating the classification of peaks in MS/MS spectra via multi-objective optimisation, which has not been found investigated before.

Why Genetic Programming

Inspired by biological evolution, Genetic Programming (GP) is an evolutionary algorithm that uses a variable-length individual representation, traditionally tree-based structures, to evolve computer programs to automatically build a solution for a predefined task. GP has been successfully applied to solve various classification problems [31, 32].

GP has the ability to automatically evolve a model that fits the training data without any prior knowledge or assumption. Moreover, GP using a tree-based representation has the capability for implicit feature selection during the evolutionary process. Therefore, the main advantage of GP in classification problems is its ability to simultaneously evolve a good classification model and implicitly selects only a subset of features during the evolutionary process.

GP has the potential to cope with complex problems and has good learning capability even from imbalanced data [30, 33, 34]. GP can adapt its fitness function to evolve an individual that is capable of dealing with the class imbalanced problem [30]. Unlike other machine learning algorithms, GP has the ability to combine several advantages: GP can integrate various types of data and generate effective models; such models are not black-box models, but instead they are highly interpretable and readable by human. Moreover, GP is able to handle two conflicting objectives, the accuracy of the minority class and the accuracy of the majority class, in imbalanced data using evolutionary multi-objective optimisation (EMO) [35].

There have been successful attempts to use GP and Pareto dominance-based algorithms to solve the class imbalanced problem by maximising two conflicting objectives, the classification accuracy of the minority and majority classes [36]. While Pareto dominance-based algorithms usually produce non-dominated solutions around the centre of the Pareto front, decomposition-based EMO algorithms benefit from having the ability of differently allocating resources to better approximate the Pareto front [37].

Multi-objective evolutionary algorithm based on decomposition (MOEA/D) is an efficient framework for EMO and has been previously applied on several real-world problems such as feature selection for classification problems [37], web service composition [38], and optimal power flow problem [39]. As GP showed to be a promising tool in MS analysis [40, 41], its potential for further improvement in handling two conflicting objectives of majority and minority classes using EMO and particularly MOEA/D has not been investigated in MS/MS analysis.

1.2.2 Limitations of Current *De Novo* Peptide Sequencing Algorithms

As mentioned before, chemical noise, internal cleavages, missing ion types caused by incomplete fragmentation, and sometimes low instrumental accuracy are critical issues for *de novo* sequencing, as they cause confusion during sequence predictions [42, 43].

Graph-theoretical algorithms and dynamic programming have been widely used to solve the complex optimisation task of *de novo* sequencing in the existing methods [9, 10, 44, 45, 46, 47, 48]. Graph theory approaches generate a graph from an MS/MS spectrum where peaks are the vertices and edges are defined as the corresponding amino acids to the mass differences between two vertices. The paths are scored based on a fitness function and dynamic programming is used to traverse through the spectrum graph.

However, a spectrum graph approach has some major difficulties such as having a huge graph due to the noise peaks caused by internal cleavages, contaminants or post-translational modifications. Another problem is the lack of full path due to the missing ion types caused by incomplete fragmentation and low instrument accuracy. Therefore, *de novo* sequencing of full-length peptides remains a challenge.

Being fragile to the missing ions, dynamic programming has an exhaustive enumerating nature when dealing with a big search space due to the presence of noise. For large datasets with a large number of instances and a large set of features, this exhaustive search will not be practical. Even innovative approaches that combine deep learning and dynamic programming to solve the optimisation task of *de novo* sequencing need be further enhanced with more advanced search algorithms [11, 15] to deal with the combinatorial explosion of evaluating all possible sequences.

Why Genetic Algorithms

De novo sequencing can be formulated as an optimisation problem where the objective is to discover the most likely amino acid sequence that can be generated by the input spectrum [49]. Genetic Algorithms (GAs) are suitable to solve the problem of *de novo* sequencing where a GA tries to optimise the amino acid sequence in respect to a scoring function. GA is a population-based problem solving technique which employs techniques inspired of Darwin's theory of evolution such as recombination, mutation, natural selection and survival of the fittest in order to evolve a population of individuals.

With the ability of GAs in exploring a large search space of potential amino acid sequences, GA is able to infer the most likely amino acid sequence directly from the spectrum. GA is a heuristic method and is suitable to solve such NP-hard, large-scale problems. GA can be adapted to represent an individual as variable-length bit string called a chromosome which is appropriate for keeping a peptide sequence containing a series of amino acids.

Unlike exhaustive approaches, a GA does not generate all possible amino acid sequences for a given spectrum. Instead, it can start with a set of initial amino acid sequences as its initial population and then during the evolutionary process manipulate these sequences by appropriate genetic operators until finding the one that best fits to the spectrum in respect to the fitness function. Moreover, unlike spectrum graph based algorithms, the performance of a GA does not deterred by discontinuities in the search space (lack of full path in the graph) due to missing ions. In addition, any component of a GA including the fitness function, genetic operators, evolutionary process can be specifically designed for the domain dependent problem.

1.2.3 Limitations of Current PSM Scoring Functions

As previously mentioned, to find the best match to the experimental spectrum from the results of *de novo* peptide sequencing, normally the highest-scoring candidate PSM is returned as the results of identification. However, the best match does not always indicate the correct (true) match [50].

Many of the existing *de novo* peptide sequencing algorithms suffer from the lack of suitable PSM-scoring functions to measure the goodness of a match between a spectrum and a peptide [51]. Due to the amino acid permutation complexity, often the predictions are similar to each other with even equal confidence scores, which makes it very difficult for the *de novo* sequencing algorithm to distinguish or rank them properly. This results in a high number of false identifications in the result of *de novo* sequencing. Therefore, a quality control strategy to validate the results of *de novo* sequencing in order to increase the accuracy of prediction is required.

A large number of scoring methods are based on the shared peak count approach [52, 53] that may ignore the importance of informative peaks. Some other methods have been developed based on likelihood ratio hypothesis testing [44], Hidden Markov model [54], and decision tree [55, 10]. However, these scoring methods mainly focused on partially correct *de novo* peptide sequencing as their performance evaluation metric is the number of correctly predicted amino acids in each peptide rather than full-length peptide prediction.

PSM scoring is to provide users with accurate and relevant results on the top of the search results [56]. The aim of the scoring function is to come up with an optimal order of the search results [57]. Therefore in the case of MS/MS analysis, a scoring function to re-score and re-rank the order of the results of peptide identification is required. There have been attempts to improve the results of database search algorithms by developing machine learning approaches to learning a model to distinguish between correct and incorrect PSMs [5, 50, 58]. However, *de novo* sequencing algorithms have been given little attention, and most of the methods focus on improving the scoring function implemented in the *de novo* sequencing algorithm [59, 10, 48] rather than developing a post-processing method for *de novo* peptide sequencing. Therefore, for improving the results of *de novo* peptide sequencing, a post-processing step in the form of a PSM scoring function to re-score and re-rank the order of the results of *de novo* peptide sequencing is necessary.

Why GP

Building a PSM scoring function from the possible combinations of different similarity scores (sub-scores) can be considered as a regression problem, where the sub-scores are treated as features. To build such a scoring function, a function identification process is required to identify the hidden relationship between the variables in the dataset and discover the mathematical function models [60]. Symbolic regression is a type of regression analysis that attempts to find the model that best fits a given dataset by discovering both model structure and parameters at the same time. Being a function identification process, symbolic regression does not face the problem of unknown gap in domain knowledge or human bias [61, 62]. Having symbolic nature of solutions and being independent of any prior knowledge, GP [63] is a promising method for symbolic regression problems [64]. Symbolic regression using GP has been successfully applied to many real-world applications such as finance [65, 66], industrial processing [67, 68], and software engineering [69, 70, 71]. Moreover, GP has been used to automatically build effective ranking/scoring functions for information retrieval [72, 73, 74, 75]. Therefore, it is worth investigating how a symbolic regression based approach using GP can find the intrinsic relationship between the sub-scores and improve peptide identification.

Moreover, since distinguishing the true PSM from false PSMs can be treated as a classification problem, while building the PSM scoring function can be considered as a regression problem, developing a method that can simultaneously solve these two problems might be useful for building a powerful discriminative PSM scoring function which can contribute to improving peptide identification. Due to the fact that machine learning algorithms often solve either a classification or a regression problem, not two problems together, it is worth investigating the capability of GP in this regard.

1.3 Research Goals

The overall goal of this thesis is to investigate the capability of evolutionary algorithms particularly GP and GAs in improving the *de novo* peptide sequencing outcome and develop a new evolutionary learning approach to improving *de novo* sequencing and peptide identification using GP/GAs. To achieve this goal, a set of specific research objectives of this work are described in more details as follows.

1. Develop an effective classification method using GP to classify signal and noise peaks for the purpose of preprocessing MS/MS spectra prior to peptide identification. Since the number of signal peaks is very small compared to the number of noise peaks, the classification algorithm needs to handle the problem of imbalanced MS/MS data. As the ratio of imbalance varies between MS/MS datasets, the stability of the GP method across various ratios of signal to noise (S/N) will be investigated and the results will be compared with different types of classification algorithms. Moreover, the effectiveness of the proposed method will be evaluated in terms of the improvement in the reliability of peptide identification with the most commonly used peptide identification tools on a large-scale dataset and the results will be compared to the original un-preprocessed (raw) data and the intensity-based thresholding method in the literature.

2. Develop a *multi-objective* GP (MOGP) approach based on the idea of MOEA/D to solve the class imbalanced problem by evolving a Pareto front of classifiers along the two objectives of maximising the minority and majority class trade-off frontier. It is expected that the proposed algorithm can evolve a set of non-dominated classifiers along the optimal trade-off surface that offers the best compromises between the two conflicting objectives i.e., the majority class and minority class accuracies. The stability of the proposed method with the decrease in S/N ratio in the MS/MS data in terms of convergence to the Pareto front will be investigated and the results will be compared with an MOGP based on non-dominated sorting genetic algorithm II (NSGA-II) [76], a popular elitist method which according to the literature produces good solutions in the centre of the Pareto front [37]. Moreover, the classification performance of the best compromise solutions evolved by the proposed method will be compared with the best solutions evolved by the single objective GP approach and those of the NSGA-II based GP method. The proposed multi-objective GP approach is expected

to achieve better performance than the single objective GP approach, outperforming the NSGA-II based GP method as well.

3. Develop an effective *de novo* sequencing algorithm using GAs to construct the full-length amino acid sequences of MS/MS spectra by proposing a new initialisation method, a new fitness function and new updating mechanisms in GAs. The proposed method is expected to infer the most likely amino acid sequence directly from the spectrum. The fitness function needs to incorporate important spectral features and fragmentation rules in order to enable GAs to discriminate the mismatches. The proposed method will be evaluated in the amino acid level, indicating the ratio of partially correct sequences, and at the peptide level, reflecting the ratio of full-length peptide sequencing. The performance of the proposed method will be compared to PEAKS as the state-of-the-art *de novo* peptide sequencing algorithm which is the most commonly used *de novo* sequencing tool in proteomics community as well [9, 77].

4. Develop an effective GP-based peptide-spectrum match scoring approach to evolve scoring functions to re-score and re-rank the *de novo* peptide sequence predictions, which are the output of a *de novo* sequencing algorithm, in order to find the optimal ranking of those items. A novel GP strategy aiming at improving the rate of *de novo* peptide identification via simultaneously solving a regression and a classification problem will be proposed. An appropriate fitness function that lead GP towards building powerful discriminative PSM scoring functions in order to distinguish between the correct PSMs and incorrect ones will be designed. The effectiveness of the GP evolved PSM scoring function to post-process the results of *de novo* sequencing tools in terms of the full-length peptide identification rate will be evaluated and the results will be compared with other Non-GP methods.

1.4 Major Contributions

The thesis makes the following four major contributions.

1. This thesis has shown how GP can be used for effectively preprocessing and denoising CID MS/MS spectra, and improving the reliability of peptide identification. The proposed method is useful to perform binary classification effectively on highly imbalanced MS/MS spectra, where the two classes are signal and noise, as it is not biased towards the accuracy of the majority class containing the noise peaks. GP takes advantage of using an effective fitness function that accounts for both the minority and the majority class accuracies in the evolved classifiers. The experimental results on a large-scale dataset show that using the proposed GP method prior to peptide identification with either *de novo* sequencing or database searching methods results in improving the high confidence peptide assignments to MS/MS spectra. Therefore, this method clearly shows potential promise in decreasing the number of unassigned MS/MS spectra in the large-scale MS/MS proteomics analysis when is coupled with a peptide identification tool.

   Parts of this contribution have been published in:

   Samaneh Azari, Mengjie Zhang, Bing Xue and Lifeng Peng. “Genetic Programming for Preprocessing Tandem Mass Spectra to Improve the Reliability of Peptide Identification.” IEEE World Congress on Computational Intelligence/ IEEE Congress on Evolutionary Computation (WCCI/CEC 2018). Rio de Janeiro, Brazil, 8-13 July, 2018. DOI: 10.1109/CEC.2018.8477810.

   Samaneh Azari, Bing Xue, Mengjie Zhang, Lifeng Peng. “Preprocessing Tandem Mass Spectra Using Genetic Programming for Peptide Identification”. Journal of The American Society for Mass Spectrometry. 2019. pp. 1294-1307, Vol. 30, No. 7, DOI: 10.1007/s13361-019-02196-5.

2. This thesis proposes the first multi-objective GP approach based on MOEA/D, named MOGP/D, to solve the class imbalanced problem in MS/MS data by maximising the two conflicting objectives, the accuracy of the minority class and the accuracy of the majority class. In comparison with an NSGA-II based MOGP method (NSGP) with decreasing S/N ratio, MOGP/D produces better solutions in the region of interest (centre of the Pareto front) according to the hypervolume indicator on the training sets, showing to be a more stable approach when facing high noise ratios. Moreover, the best compromise solutions of MOGP/D outperformed those of NSGP and the best solutions evolved by the single objective GP (from the first contribution) in terms of sensitivity, specificity and average accuracy on both training and test sets. The results shows that selecting the best compromise solution of MOGP/D to preprocess the MS/MS spectra prior to *de novo* sequencing has a positive influence on the peptide identification reliability.

   A part of this contribution has been accepted by:

   Samaneh Azari, Bing Xue, Mengjie Zhang, Lifeng Peng. "A Decomposition Based Multi-objective Genetic Programming Algorithm for Classification of Highly Imbalanced Tandem Mass Spectrometry." The 5th Asian Conference on Pattern Recognition (ACPR 2019). Auckland, New Zealand. 26-29 November 2019. (Accepted). 14pp.

3. This thesis has shown how GAs can be used to for solving complex optimisation task of *de novo* peptide sequencing and constructing full-length peptide sequences by proposing a genetic algorithm based method, GA-Novo. Given an MS/MS spectrum, GA-Novo optimises the amino acid sequences to best fit the input spectrum. The developments presented in this work are a new domain dependent fitness function, a new initialisation method and two new genetic operators that were particularly designed for the task. The fitness function was able to capture main spectral features and guide the GA to produce the fully matched

peptides. The tag-based initialisation method helped the GA start with a better/fitter initial population. The genetic operators helped GA maintain the diversity in the population and gradually convert partial matches to fully matched sequences. On the testing dataset, GA-Novo outperforms PEAKS at the amino acid level and at the peptide level.

A part of this contribution has been published in:

Samaneh Azari, Bing Xue, Mengjie Zhang, Lifeng Peng. "GA-Novo: De Novo Peptide Sequencing via Tandem Mass Spectrometry using Genetic Algorithm". Proceeding of the 22th European Conference on Applications of Evolutionary Computation (EvoApplications 2019). Lecture Notes in Computer Science. Vol. 11454, Leipzig, Germany. 24-26 April 2019. pp. 72-89.

4. This thesis proposes a novel strategy to generate effective PSM scoring functions to improve the ordering/ranking of the PSMs which are the outputs of a *de novo* sequencing algorithm. This is the first GP approach that evolves computer programs to perform regression and classification tasks simultaneously in order to generate an effective PSM scoring function, which is able to: (1) produce the exact score that each PSM gets via the regression task, and (2) look after of distinguishing the correct PSM from the incorrect PSMs via the classification task. Since machine learning algorithms often solve either a classification or a regression problem, not two problems together, GP shows its capability in handling these two tasks at the same time. Unlike other machine learning algorithms, GP is able to learn from multiple sources of information. From the training set which is suitable for the regression task, GP learns how to assign appropriate scores to the PSMs. Also from the designed training set suitable for classification, GP learns to give the greatest score to the correct PSM in order to bring it ahead of all incorrect PSMs for the same spectrum. The experimental results show that proposed GP method outperforms other GP-based and non-GP

methods in terms of improving the false identification rate.

Parts of this contribution have been published in:

Samaneh Azari, Bing Xue, Mengjie Zhang and Lifeng Peng. "Learning to Rank Peptide-Spectrum Matches Using Genetic Programming". Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC 2019). Wellington, New Zealand, 10-13 June, 2019. pp. 3244-3251.

Samaneh Azari, Bing Xue, Mengjie Zhang, and Lifeng Peng. Improving the results of *de novo* peptide identification via tandem mass spectrometry using a genetic programming-based scoring function for re-ranking peptide-spectrum matches. In Pacific Rim International Conference on Artificial Intelligence (PRICAI 2019), pages 474-487. Springer, 2019.

1.5 Organisation of Thesis

The remainder of this thesis is organised as follows. Chapter 2 provides background on proteomics analysis followed by an overview on machine learning, evolutionary algorithms, GA and GP. Moreover, a review of the previous works on peptide identification by *de novo* sequencing will be presented in this chapter as well. The main contributions of this thesis are presented in Chapters 3-6. Chapter 7 concludes the thesis.

Chapter 2 provides a background on proteomics analysis followed by an overview on machine learning, evolutionary computation particularly GP, GA and evolutionary multi-objective optimisation. Moreover, a review of the previous works on *de novo* sequencing based on the traditional methods and machine learning algorithms including GA and GP has been presented in this chapter as well.

Chapter 3 proposes an effective GP-based preprocessing method for denoising highly imbalanced MS/MS spectra. A set of experiments are conducted to investigate the important ion types for labelling the peaks in the MS/MS datasets, appropriate measures for performance evaluation of the GP method, stability of GP across various ratios of S/N, and the effectiveness of the proposed method in terms of improving the peptide identification reliability. The interpretability of GP is shown by analysing the best GP evolved program and important spectral features selected by GP are revealed.

Chapter 4 proposes a new decomposition-based (MOEA/D) evolutionary multi-objective GP approach to solve the class imbalanced problem in MS/MS spectra by maximising the minority class and the majority class accuracies. It presents a new MOEA/D weight vector initialisation method which allocates the resources more efficiently to approximate the Pareto front. The stability of the proposed method with the decrease in S/N ratio is then investigated in terms of convergence to the Pareto front and the results are compared with a Pareto dominance-based (NSGA-II) multi-objective algorithm. The chapter analyses the classification performance of the best compromise solutions evolved by both MOGP methods and compares them with the best solutions evolved by the single objective GP approach proposed in Chapter 3.

Chapter 5 presents an effective GA-based *de novo* sequencing algorithm for constructing the full-length peptide sequences. The chapter proposes a new fitness function, a new initialisation method, and an effective set of mutation and crossover operators that help GAs construct the full-length amino acid sequences. The proposed algorithm is compared with PEAKS and a GA-based *de novo* sequencing. The chapter also provides further analysis on the effectiveness of each component used in the GA method.

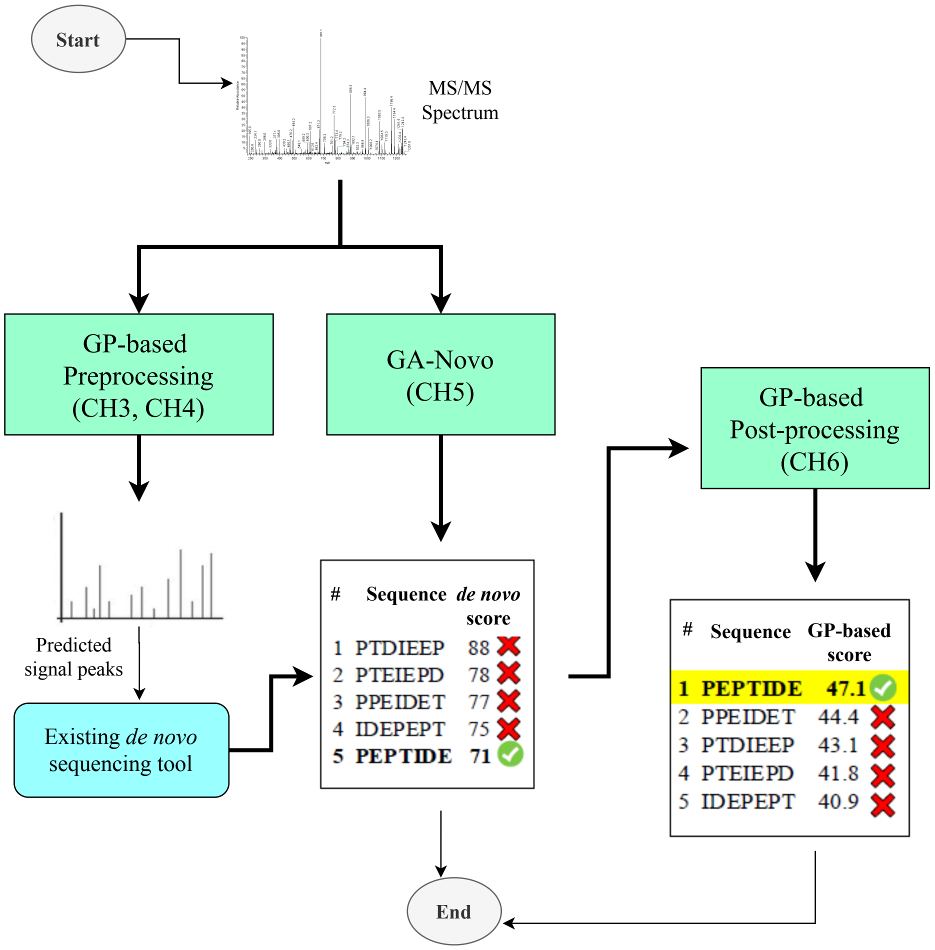
Chapter 6 proposes a new GP method which is able to learn simultaneously from different training sources and solve a regression and a classification problem at the same time. The proposed method is used to generate effective PSM scoring functions to optimise the scores of the *de novo* sequencing results, aiming at identifying the correct PSMs from incorrect ones. The result of the proposed GP method is compared with other GP-based and non-GP based methods in terms of improving the accuracy of *de novo* sequencing at the peptide level. The evolved programs are then analysed to investigate the implicit feature selection ability of the GP approaches and important spectral features are recognised.

Chapter 7 concludes this thesis and summarises the key findings. The research contributions and key points are discussed. The future research opportunities and directions are also discussed in this chapter.

1.6 The Overall Pipeline

The flowchart presented in Figure 1.1 shows the relationship between the proposed methods in this thesis along with the input and output of each method. Given the input spectrum to the preprocessing methods proposed in chapter 3 and chapter 4, the output is the denoised spectrum containing only b-/y-ions. The denoise spectrum is given to the existing *de novo* sequencing tools and their effectiveness is measured based on the improvement in the confidence score of each identified peptide.

However, instead of using the existing *de novo* sequencing tool, Chapter 5 develops a new GA-based *de novo* sequencing algorithm, called GA-Novo. GA-Novo gets the MS/MS spectrum as input and through the evolutionary process generates the full-length amino acid sequence. As can be seen in the flowchart unlike the existing *de novo* sequencing tool, GA-Novo does not get its input from the GP-based preprocessig method. The reason is that GA-Novo is robust enough against the noise in the spectrum due to its special design. Various components in GA-Novo such as an intensity-based denoising, a tag-based initialisation and different genetic operators help GA-Novo handle the problem of large search space due to the presence of noise and missing values. Therefore, we believe that GA-Novo does not need a machine learning based processing method prior to sequencing. At the end of each GA run, the solution with the highest fitness value is selected as the result of the identification. The process of peptide identification is finished here but the identification rate can be further improved by applying the GP-based post processing model. This gives the chance of the correct


Start
MS/MS Spectrum
GP-based Preprocessing (CH3, CH4)
GA-Novo (CH5)
GP-based Post-processing (CH6)
Predicted signal peaks
Existing *de novo* sequencing tool
End

#	Sequence	*de novo* score	
1	PTDIEEP	88	✗
2	PTEIEPD	78	✗
3	PPEIDET	77	✗
4	IDEPEPT	75	✗
5	**PEPTIDE**	**71**	✓

#	Sequence	GP-based score	
1	**PEPTIDE**	**47.1**	✓
2	PPEIDET	44.4	✗
3	PTDIEEP	43.1	✗
4	PTEIEPD	41.8	✗
5	IDEPEPT	40.9	✗

Figure 1.1: The overall structure of the contributions in this thesis.

identification for the input spectrum by bringing the correct peptide on the top of its candidate peptide list.

1.7 Benchmark MS/MS Dataset

The proposed methods in this thesis needs to be trained and tested using a set of MS/MS spectra with known identifications. This is a necessary condition where the correct identification for each spectrum is required to be known. Although a number of proteomics LC-MS datasets have been released, only a relatively small number of them could be considered as benchmark datasets, or were even designed for that purpose. Many of the existing peptide identification tools are not benchmarked on well-annotated MS/MS datasets. Their ground-truth is the results of database searching methods where there is the potential of existence of false positives.

The MS/MS spectra that are used in this thesis are selected from a comprehensive full factorial LC-MS/MS benchmark dataset specially designed for the purpose of benchmarking data analysis methods [78]. The data were obtained from the linear ion trap Fourier-transform (LTQ-FT, Thermo Fisher Scientific) with CID. The dataset contains MS/MS spectra from 50 protein samples extracted from *Escherichia coli* (*E. coli*) K12 and spiked with either 0, 0.125 µg, or 0.5 µg of bovine carbonic anhydrase II and/ or chicken ovalbumin. The spectra in the dataset are high-quality ion trap CID MS/MS spectra in the Mascot generic peak list format. The results of peptide identifications, known as ground-truth, were acquired using Mascot (v2.2) which is searched against a curated Refseq released 33 *E. coli* database with the following parameters: precursor mass tolerance was set to 10 ppm, fragment error was set to 0.8 Dalton (Da), and Mascot peptide identification of 1% false discovery rate (FDR) according to [78]. A total number of 304,321 MS/MS successfully matched with tryptic peptide sequences corresponding to 10,166 unique peptides (resulting 10,166 PSMs), or 968 proteins which are covering 23% of the *E. coli*, were obtained.

The MS/MS acquisition method used to acquire the spectra in full factorial dataset is data-dependent (or "shotgun") acquisition (DDA). In this technique a fixed number of precursor ions (usually the most intense ions) are selected for fragmentation and analysed by tandem mass spectrometry. Co-isolated precursor ions is when DDA often co-isolates more than one " peptide" i.e. the observed fragment ions arise from two or more different peptides (often of different charge states). Due to the fact that unique peptides are more informative and more desirable to protein inference, the experiments conducted in this thesis only focus on unique peptides. Such peptides are usually long with average length of 7 or longer [79] and peptides in full factorial dataset have this characteristics.

As based on the CID fragmentation rules, the charge number of the peptide and its precursor mass influence its fragmentation pattern [16], this study only focuses on doubly charged peptides with no modifications. According to the task and design of the experiments, several different meaningful subsets of sufficient size are possible to be extracted from the dataset. The important point is to make sure that the training sets do not include the peptides in the test sets, and that is the reason that we use unique peptides to run the experiments in this thesis.

Chapter 2

Literature Review

This chapter gives the background on proteomics analysis including a short overview of the components in tandem mass spectrometry and current spectral analysis algorithms. In addition, a brief review of machine learning, evolutionary computation, GP and GA have been provided as well. Finally, this chapter ends with a review of related traditional and machine learning approaches to *de novo* sequencing and peptide identification.

2.1 Background on Proteomics Analysis

Proteomics refers to the large-scale study of proteins. Proteomics analysis is the systematic identification and quantification of proteins, particularly their sequences, structures and functions at a certain point in time. Identification of protein sequences and their modifications is very important in proteomics because it allows researchers discovering possible genetic diseases in an organism [80]. Mass spectrometry has been practically recognised as the primary tool for protein identification. With significant improvements in both accuracy and performance speed, these tools are highly reliable for the high-throughput data analysis of proteomics. The rapid development of MS machines has caused complicated configurations and various data analysis algorithms.

The commonly used technique for proteome analysis involves four main steps consisting of protein separation, protein digestion, mass spectrometry analysis as well as peptide and protein identification [4]. Figure 2.1 illustrates a typical proteomic experiment which involves the aforementioned steps. The circled numbers in this figure are explained in more details as follows.

First Step: Protein Separation

① At first, a protein of interest will be separated from a complex mixture, usually cells, tissues or whole organisms.

Second Step: Protein Digestion

② Then, the protein is digested with trypsin into short peptide fragments (trypsin cleavages the protein at the carboxyl side of amino acids lysine 'K' and arginine 'R' unless they are followed by amino acid proline 'P'). Both the primary structure of the protein (the linear sequence of 20-letter alphabets) and its three-dimensional arrangement are shown in this figure.

Third Step: Mass Spectrometry Analysis

③ The mass spectrometry (MS^1) analyses the ionised peptides and generates an MS spectrum composed of the m/z values of the ions (precursor ions) and their corresponding relative intensities (an m/z value and its relative intensity are denoted as an experimental peak).

Fourth Step: Peptide and Protein Identification

④ An MS spectrum can be identified by matching the measured masses of the spectrum to the corresponding peptide masses of a protein from a protein database. This process is called *protein identification* by *peptide mass fingerprinting (PMF)* method.

⑤ However, more information about the amino acid sequence of a peptide can be acquired using MS/MS (or MS^2) [17, 81]. In MS/MS technique, a selected precursor ion is fragmented into fragment ions whose m/z values

This content is unavailable.
Please consult the print version for access.

Figure 2.1: A typical proteomic experiment, adapted from [4].

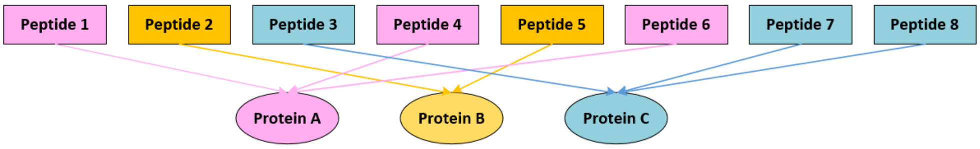


Figure 2.2: Protein inference.

are measured by mass spectrometry to generate an MS/MS spectrum. Therefore, an MS spectrum corresponds to a protein, while an MS/MS spectrum is related to a peptide.

⑥ Peptide identification can be performed by comparing the input experimental MS/MS spectrum with theoretical spectra predicted for each peptide sequence in a protein sequence database (*database search* method). Alternatively, instead of searching the experimental MS/MS spectrum against a database, peptide sequences can be extracted directly from the spectrum with the *de novo* sequencing approach.

Most of today's proteomics analyses are done with MS/MS [17]. Successful proteome analysis requires good experimental design, high quality data and powerful computational tools for protein identification [82]. Protein identification comprises two stages. The first step is peptide identification (Figure 2.1 stage (6)) which translates information embedded in spectra to generate a set of putative modified and unmodified peptides followed by the second step, where the identified peptides are assembled to infer the sequence of a protein (Figure 2.2).

2.1.1 Mass Spectrometer (MS)

The mass spectrometer is a tool that measures the mass-to-charge ratio (m/z) of charged particles [83]. It is composed of five components which are shown in Figure 2.3. The *sample inlet* brings the sample that must be given to the *ion source*. The ionisation chamber ionises the stream of molecules and

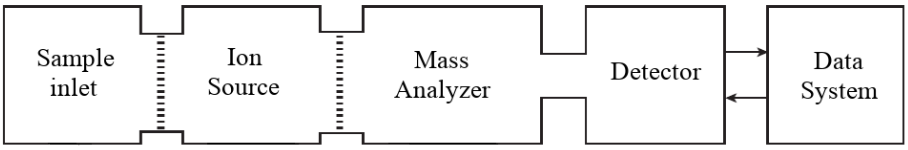


Figure 2.3: A typical mass spectrometry structure.

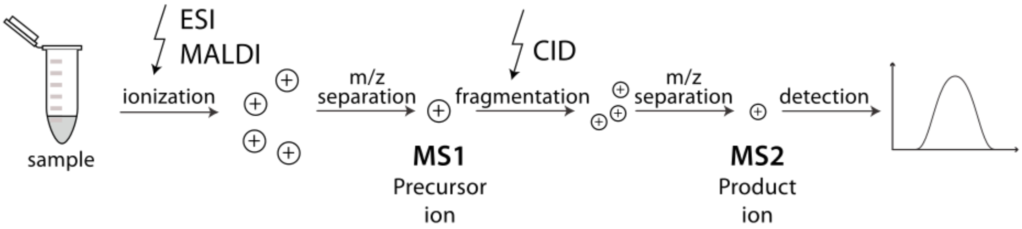


Figure 2.4: Schematic of tandem mass spectrometry.

converts them to charged particles or ions. Then, these ions are accelerated by an electromagnetic field and transferred to *mass analyser* in order to be separated based on their m/z values. Next, a *detector* counts the ions for each m/z and the signal is processed by a *data system* which produces the mass spectrum. A mass spectrum is a stick diagram of the number of ions detected as a function of their m/z ratios.

2.1.2 Tandem Mass Spectrometry (MS/MS)

Tandem mass spectrometry (MS/MS) involves multiple steps of mass spectrometry along with fragmentation occurring in between the steps [84]. In a tandem mass spectrometer, the sample is ionised in the ion source and separated according to m/z ratio by the mass analyser. Then, ions of a particular m/z is selected to be further fragmented into fragment ions whose m/z values are measured by the second mass analyser. Next, the detector counts the resulting ions for each mass and a tandem mass spectrum is generated at the end. Figure 2.4 illustrates the schematic of tandem mass spectrometry.

The following sections describe more details about different components evolved in obtaining MS/MS spectra.

Sample Introduction

A sample introduced to a mass spectrometer may be a gas, a liquid, or a solid. A sufficient amount of the sample is required to be transformed into the vapor state to acquire the stream of molecules that must be given to the ionisation chamber. Therefore, the sample inlet system is often connected to a chromatograph device, which is used for the separation of a mixture by travelling various constituents of the mixture at different speeds. This allows the complex mixture of components to be separated, reducing the complexity of the mass spectrum and increasing the detection coverage [83].

Ionisation Methods

Once the sample has introduced to the ion resource, it needs to be converted to ions. Ionisation techniques include electron ionisation (EI), chemical ionisation (CI), desorption ionisation (DI, including SIMS, FAB, and MALDI) and electrospray ionisation (ESI) [85].

In EI-MS, which is the simplest ionisation method, a beam of high-energy electrons interacts with the stream of molecules that has been admitted from the sample inlet in order to knock off the electrons and produce ions. The hardware required in this technique is low-cost and robust. In addition, fragmentation pattern of a sample is reproducible and there exists many available libraries of EI-MS data. However, since some compounds fragment easily, they cannot be detected by the mass analyser due to their short molecular ions lifetime. In this case, the compound's molecular mass cannot be detected accurately.

In CI-MS, the sample molecules are combined with a stream of ionised reagent gas [86]. CI-MS method is a lower energy process than EI. This leads to obtaining a less complex spectrum because of less or sometimes no fragmentation, resulting insufficient information that can be achieved about the ionated species. Both EI and CI techniques are suitable for low molecular weight samples.

DI techniques allow the analysis of high weight and nonvolatile molecules with minimal fragmentation. In such techniques, the sample is dissolved or dispersed in a matrix substance, which consists of crystallized molecules, and the mixture is placed in the path of a beam of ions in secondary ion mass spectrometry (SIMS), neutral atoms in fast atom bombardment (FAB), or high-intensity photons in matrix-assisted laser desorption ionisation (MALDI). SIMS and FAB ionisation techniques have molecular weight limitation up to about 20,000 g/mol. MALDI technique is quite useful for a wide range of molecular weights. Furthermore, MALDI requires only a few amount of sample. The ionisation mechanism with the MALDI technique is done by the laser pulse fired at the matrix crystals in the dried-droplet spot. At a certain time, the matrix absorbs the laser energy, resulting in being ionised. However, to use MALDI technique, a mass analyser which is compatible with pulsed ionisation techniques is required. MALDI is mostly useful for singly-charged ions, therefore analysis of MS/MS spectra is difficult with this method [83].

ESI is a useful technique for analysis of high molecular weight biomolecules and nonvolatile compounds. As shown in Figure 2.5, the sample is mixed with a liquid. Then, the solution is injected via a metallised needle on which a high potential difference is applied. The sample solution is sprayed out the end of a fine capillary into a heated chamber. The heat and gas flows desolvate the ions of the sample solution. Ions are analysed by a mass analyser while several stages of differential pumping and mass analysing are done. The spray in ESI can progressively produce multiply charged ions, therefore ESI is the best method for analysing multiply charged compounds and compatible with MS/MS spectra [87]. However, it has relatively complex hardware compared to other ion sources.

Mass Analysers

After ionising the sample, the ions are accelerated by an electric field to pass into the mass analyser which separates the ions according to their

This content is unavailable. Please consult the print version for access.

Figure 2.5: Schematic of electrospray ionisation mass spectrometry (ESI MS), adapted from [88].

m/z ratios. Just like there are several ionisation techniques for different applications, there exists also various types of mass analysers [83]. The most common mass analysers include:

- **Quadrupole ion trap mass analyser**: This mass analyser consists of four voltage carrying rods. The ions enter the area between these electrodes. The electrical fields cause the ions of certain m/z values to orbit in the space while passing through the radio frequency quadrupole field. By increasing the radio frequency voltage, heavier mass ion orbits are stabilized and trapped in a two-dimensional electrical field. This causes them to collide with the wall and to be detected by the detector [89].

- **Time of flight mass analyser (TOF)**: In this method, an ion's m/z ratio is determined via a time measurement based on the kinetic

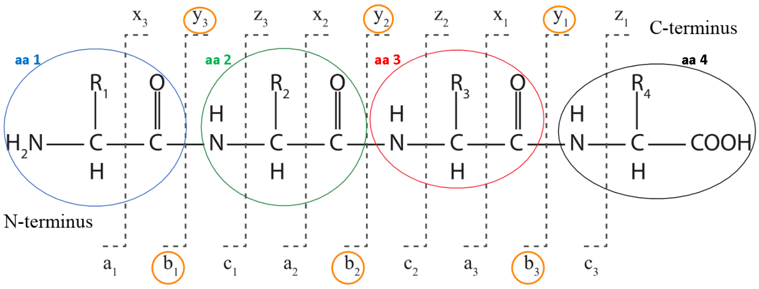


Figure 2.6: Peptide fragmentation in an MS/MS experiment results in 18 fragments which are a,b,c and the x,y,z ion series.

energy and velocity of the ions. A static electric field accelerates the ions and then measures the time they require to reach the detector. The velocity of the ion depends on the m/z ratio, therefore, lighter ions reach detector first.

- **Magnetic/electrostatic sector mass analyser**: Similar to TOF, the ions are accelerated through a flight tube, where the ions are separated by a magnetic/electric field based on their m/z ratios.

- **Ion cyclotron resonance**: This mass analyser uses a magnetic field in order to trap all ions of a particular range into an orbit inside of it. Then electric signals from the trapped ions are generated by applying an external electric field. A Fourier transform is used to differential to the summed signals for different masses to produce the desired results [85].

Fragmentation

Dissociating the precursor ions is called fragmentation. The pattern in the mass spectrum of a fragmented molecule can be used to determine structural information of the molecule. Collision-induced dissociation (CID) is an extensively studied technique which is known to be highly suitable for the identification of peptide sequences [90]. In this technique, fragmentation happens at the peptide bonds, producing b-/y-ions. Each pair is related to the sequence fragments of the precursor peptide. As mentioned in Figure 2.1

at stage (5), in the MS spectrum, each precursor ion, which indicates the m/z value of a peptide, can be selected and fragmented into hundreds of fragment ions that construct an MS/MS spectrum. During fragmentation, different fragment ion types are generated. Figure 2.6 illustrates the pattern of product ions produced by the CID fragmentation technique with four amino acids which are labelled as aa1, aa2, aa3 and aa4. Those fragments that appear to extend from the left side (amino terminus or N-terminus) of the peptide are termed a, b and c ions (prefix ions), while x, y and z ions (suffix ions) starts from the right side or C-terminus of the peptide. The fragment containing only the first amino acid from N-terminus is termed b_1, while the one that contains the first two amino acids is called the b_2 ion, and so forth.

In the CID fragmentation technique, we are only interested in b- and y- ions because the amino acid sequence of an MS/MS spectrum can be determined by the mass differences between b- (or y-) ions. A complete peptide fragmentation gives a contiguous series of ion types which is called *ladder*. For example for an MS/MS spectrum with n amino acids, the ladder constructed from b-ions refers to the peaks corresponding to the prefix ions observed sequentially in the spectrum i.e. $b_1, b_2, b_3, b_4, ..., b_n$. Offsets equal to the mass of an amino acid exist between each ion (the first fragment b_1 is seldom observed in the spectrum of peptides with free N-termini, since there is no carboxyl group to induce cleavage). Likewise, the sequential suffix ions in the MS/MS spectrum construct the y-ion ladder i.e. $y_1, y_2, y_3, ..., y_n$.

Table 2.1 shows an example of a mass fragmentation ladder. It shows a set of possible fragmented ion pairs (b- and y-ions) for the peptide ‘SGFLEEDELK’. It can be seen that each b-ion has a corresponding y-ion. These ions are called complementary ions when the sum of their masses equal to the mass of the pre-fragmented peptide. Therefore, peptide identification softwares employ these fragmentation rules to generate theoretical fragment spectra, which will be matched to the experimental spectra.

Table 2.1: An example of a mass fragmentation ladder.

Mass	ion	b-ions	y-ions	ion	Mass
88	b1	S	GFLEEDELK	y9	1080
145	b2	SG	FLEEDELK	y8	1022
292	b3	SGF	LEEDELK	y7	875
405	b4	SGFL	EEDELK	y6	762
534	b5	SGFLE	EDELK	y5	633
663	b6	SGFLEE	DELK	y4	504
778	b7	SGFLEED	ELK	y3	389
907	b8	SGFLEEDE	LK	y2	260
1020	b9	SGFLEEDEL	K	y1	147

Detectors

The final element of the mass spectrometer is the detector. It consists of a counter that records the number of ions that hit a surface. The detector will produce a mass spectrum, which is a record of ions as a function of m/z. Possible detectors include electron multiplier, Faraday cups and ion-to-photon detectors [91].

2.1.3 Assigning MS/MS Spectra to Peptide Sequences

The computational analysis of peptide sequence assignment to MS/MS spectra is the next step after acquiring a desired amount of raw peak lists of MS/MS data. Conventional methods for peptide identification from MS/MS spectra can be divided into two main categories. Peptide identification can be performed by comparing the input experimental MS/MS spectra with either theoretical spectra predicted for each peptide sequence in a protein sequences database (sequence database search approach). Alternatively, instead of searching acquired MS/MS spectra against a database, peptide sequences can be extracted directly from the spectra with *de novo* sequencing approach.

Sequence Database Search Algorithms

The sequence database search algorithms are known to be the major approach for assigning peptide sequences to MS/MS spectra. There is a number of powerful computational tools using this method [92, 93, 94, 95, 96], where experimental spectra are compared with an *in silico* digested protein database. The two key components in this method are precursor mass and the mass difference between an adjacent pair of peaks. Calculating mass differences reveals particular amino acid residues.

Figure 2.7 illustrates how a basic database search algorithm interprets a noiseless spectrum, which is an ideal case. Normally, MS/MS spectra contain many noisy peaks, resulting in a complicated interpretation process. Considering a pre-defined tolerance, those candidate peptides whose masses are within the tolerated distance to the spectrum's precursor mass will be selected as a reference list from the protein database. The algorithm traverses the spectrum and calculates the mass interval between two neighbouring peaks to inference the corresponding amino acids. While traversing the whole spectrum, the database search algorithm removes those peptides from the reference list if they cannot match to the spectrum's amino acid sequence. The algorithm skips those peak intervals that they cannot match to any amino acid. When all peak intervals are examined, the reference list containing candidate peptides can be used as a guide to fill the possible gaps in the spectrum's sequence and makes the interpretation of the spectrum with missing ion peaks possible. The algorithm scores each peptide-spectrum matches (PSMs) based on the degree of similarity between the experimental spectrum and candidate peptide. Therefore, the output of a database search algorithm is a list of PSMs ranked according to the search score.

There are numerous scoring schemes for match scoring such as simple dot-product [8], cross correlation score [92, 93, 95] or more advanced statistical measures like expectation value [97]. It has been proven that only one score is not enough to select the best match, therefore, various scores have been developed to evaluate the results [92] followed by a post-processing step, This content is unavailable. Please consult the print version for access.

Figure 2.7: An example of database search algorithm to interpret a noiseless MS/MS spectrum, adapted from [98].

which is an statistical data validation process on the results of the database search algorithm.

De Novo Sequencing Algorithms

The determination of the amino acid sequence of a peptide directly from its MS/MS spectrum is called the *de novo* sequencing. Since the presence of a database is not required, this method is highly useful for identification of new proteins and peptides containing Post-translational modifications (PTMs). *De novo* sequencing algorithms reconstruct the peptides by assigning the corresponding amino acids to the mass differences between the peak pairs.

As mentioned in Section 2.1.2, the complete CID peptide fragmentation gives a contiguous series of ion types which is called ladder [99]. The *de novo* sequencing algorithm selects pairs of peaks and labels them if their mass differences are within the tolerance ranges of the amino acid's masses. Therefore, here the distances between ions of the ladder can be used to

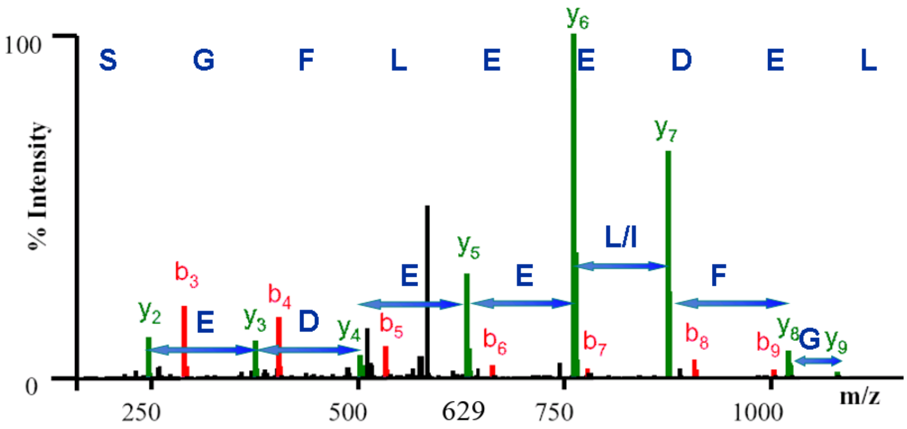


Figure 2.8: *De novo* sequencing on an ideally fragmented MS/MS Spectrum ‘SGFLEEDELK’ with two ladders.

infer the amino acids. Figure 2.8 shows the result of *de novo* sequencing on an ideally fragmented MS/MS spectrum which indicates sequence ‘SGFLEEDELK’. The Y axis indicates the relative abundance to the tallest peak in the spectrum with the tallest peak set to 100% relative intensity. The X axis shows m/z, which is mass divided by charge. The green peaks show y-ions, while red peaks represent b-ions. The blue arrows show the amino acid letters which are labelled based on the mass differences between each pair of b-/y-ions.

As an example, the difference in masses between two consecutive ions y_4 and y_5 is 129 *Da* (unified atomic mass unit or dalton) and this number indicates the mass of Glutamic (E) amino acid. Obtaining the amino acid sequence of an MS/MS spectrum using y-ions gives the reverse order of the sequence i.e. ‘EDEEJFG’ (The masses of amino acids ‘L’ and ‘I’ are not distinguishable, so the letter ‘J’ has been considered instead which indicates either ‘I’ or ‘L’).

Existing *de novo* sequencing tools include PEAKS [9], PepNovo [44], Lutefisk [100], and Novor [10]. Similar to the database search methods, *de novo* sequencing algorithms are adversely affected by spectral noise. The quality of the results of *de novo* sequencing algorithms is highly relevant to the precision of the mass spectrometer instrument and the quality of MS/MS spectra. These algorithms are not influenced by the errors in database search methods, therefore they are quite useful to identify known and unknown proteins. Another advantage of such algorithms is their ability in producing partial sequences which can be submitted to a tag-based database search algorithm to find the complete sequences that may contain PTMs.

2.2 Machine Learning

Machine learning, which is one of the sub-fields of computer science, involves the study of models and algorithms that have ability to automatically learn complex patterns and make predictions on input data [101]. An input example/sample may have a set of features which are individual measurable properties of an object being observed. Depending on the nature of learning, machine learning tasks are categorised into two main groups namely supervised and unsupervised learning [102].

In supervised learning, the training set is presented with example inputs and their desired outputs. Therefore, the machine learning task is to learn from the labelled training data and inferring a function that maps the input data to the desired outputs [103]. In unsupervised learning, no labels are presented in the training dataset and the learning algorithm needs to discover the hidden patterns in its input data by building models that provide prediction of the output to the input data. Clustering, which refers to the categorising a set of objects based on their similarities, is a typical approach to unsupervised learning [104].

2.2.1 Learning Tasks

Depending on the output of the machine learning system, various learning tasks such as classification, regression, clustering, association rules, density estimation, and dimension reduction exist [103, 101]. In this thesis, we mainly focus on classification and regression tasks which are two major applications of supervised learning.

2.2.2 Training and Testing

The learning algorithm used to perform the desire task requires to learn from a *training set* during the training process and make prediction on a *test set* during the testing process [105]. The learning algorithm builds a model that fits the training set consisting of a set of pairs of an input vector and the corresponding output vector (the output is known as target or label as well). The test set contains instances from the same problem domain, but test instances have never been used in training.

A *validation set* can be also used during the training process for model selection. An application of validation set is in early stopping for model selection where the training process stops when the error on the validation set increases. It is also used to improve generalisation and avoid overfitting. While generalisation reflects how well the machine learning model is able to predict the unseen data, overfitting indicates the poor performance of the model on data that the model has never seen [105].

2.2.3 Classification

In supervised learning, classification is a task of classifying a new unseen observation into a set of groups that are already known based on the labelled training datasets [106]. Therefore, classification involves a training and a testing stages. In the former, the classifier is built by learning from the samples accompanied by the class labels in the training dataset, while in the latter the performance of the learned classifier is measured by using unseen examples in the test dataset. Example classification algorithms include Naive Bayes classifier (NB), support vector machines (SVM), decision trees (DT) and artificial neural networks (ANN).

NB belongs to the probabilistic classifiers with a prior assumption of conditionally independent features. NB applies the Bayes theorem, which describes the probability of an event is related to the other known probabilities. Therefore, NB classifier assumes that the relationships between inputs and outputs can be represented as probability distributions [107].

SVMs attempt to construct hyperplanes in a high dimensional space and classify examples by performing linear classification. For each hyperplane, the SVM model aims to maximise the distance between the hyperplane and the nearest data points on each side of it. Then for an given unseen example, the SVM algorithm classify the example based on which side of the hyperplanes it falls on [108].

A DT is a tree-structure classifier that the paths from root to leaves represent the classification rules. The internal nodes represent a test on the attributes/features of the samples, the edges correspond to the outcome of the tests and leaf nodes represent the class labels. The learning method in DT is based on the approximated discrete valued functions. The examples in the training dataset are used for selecting appropriate tests in the DT. Typically learning is a top-down process where the algorithm attempts to choose a variable at each step that best splits the set of items. For example a DT can be built by considering those tests that maximise the information gain about the classification are selected first. A new unseen example is classified by submitting it to the tree, exploring a series of tests that identify the class label of the example when reaches a certain leaf [101].

ANN, which is a computational model inspired of the human brain and nervous system, aims at constructing a network using a number of layers that maps the instances to the target class labels [109].

2.2.4 Regression

Classic Regression

Regression is a process that aims to discover the relationship between inputs and outputs. A regression problem attempts to find a mathematical model that predicts a real value for each input example and measures the error of the prediction in an iteratively way [110]. Given a set of independent variables X and a dependent variable Y, the objective of the regression model is to approximate Y using X along with W as a set of appropriate coefficients.

$$Y = f(X, W) + \epsilon \tag{2.1}$$

where ϵ indicates possible noise.

Classical regressions assume a pre-defined functional form of f such as being a linear regression model [111].

$$f(X, W) = w_0 + w_1x_1 + w_2x_2 + ... + w_mx_m \tag{2.2}$$

Therefore, the set of coefficients may be found by least square which attempt to minimise the sum of the squares of the residuals. Other common classical regression algorithms include Back-Propagation neural networks, support vector regression (SVR) and Multivariate Adaptive Regression Splines (MARS).

The main idea in back-propagation NN is to construct a nonlinear function of input features using a transfer function which can be a sigmoid function. Training the network using known correct outputs in order to update the coefficients (weights in the network) by gradient descent is called back-propagation [112].

SVR, which is a regression version of SVM, maps the input data into a m-dimensional feature space using nonlinear mapping and then constructs a linear model in this feature space [113].

MARS is an extension of linear models that automatically models the interactions between variables. A MARS model is a weighted sum of basis functions $B_i(x)$ with corresponding coefficients c_i.

$$f(x) = \Sigma_{i=1}^{k} c_i B_i(x) \tag{2.3}$$

The basis functions can be either a constant 1 or a hinge function, which is in the form of $max(0, x - c)$ or $max(0, c - x)$, considers two different versions of a feature in the models. The constant c is called a knot. The MARS algorithm aims on creating a mirrored pair of hinge functions with a knot at each observed values of the features [114].

Symbolic Regression

Aiming at finding the model that best fits a given dataset, symbolic regression is an analysis method that discovers both model structure and parameters simultaneously. It is a function identifier which attempts to find the mathematical relationship between the input variables (known as dependent variables) and output variables (known as independent variables). The major difference between symbolic regression and the classical regression techniques is that the former does not put any prior assumption on the model by limiting its identification process to only finding the parameters/ coefficients of a specific predefined model.

No predefined function or a specific model is needed as the start point of the identification process. So, symbolic regression does not face the problem of unknown gap in domain knowledge or human bias. Symbolic regression randomly combines the mathematical building blocks such as primitive functions, independent variables, and constants in order to generate a model with minimum error difference from the target outputs. Therefore, no prior knowledge about the size and shape of the model is required. As the model found by symbolic regression should best fit the given dataset in terms of accuracy and simplicity, during the learning/modelling process irrelevant and redundant input variables are excluded from the search space. Moreover, the number of coefficients and their values are also taken into account. To ensure the accuracy of the model, the fitness function considers error metrics.

Among the existing methods proposed for solving symbolic regression tasks, EC methods, particularly GP-based methods, are still the most popular techniques for symbolic regression [63, 67, 69, 115]. GP is a powerful technique that is able to evolve data-driven models that effectively describe the data, providing human insights about the data-generating system. More details about EC techniques and particularly GP is explained in the following section.

2.3 Evolutionary Computation (EC)

Evolutionary computation is a family of population-based problem solving techniques whose employs principles based on the theory of biological evolution to get involved in many optimisation problems [18]. EC algorithms consists of Evolutionary Algorithms (EAs), Swarm Intelligence (SI) and other EC techniques such as Evolutionary Multi-objective Optimisation (EMO) methods.

EAs employ techniques inspired of Darwin's theory of evolution such as recombination, mutation, natural selection and survival of the fittest in order to evolve a population of individuals. The goodness of individuals, which determines their potential to survive and represents their ability to solve the problem, is measured by using the fitness function. Based on the representation of individuals, EAs are divided into different categories such as Evolutionary Programming (EP) [116], Genetic Algorithm (GA) [117], Evolution Strategies (ES) [118], and Genetic Programming (GP) [63].

EP uses finite-state machines to represent the individuals. Unlike GP, the structure of programs (individuals) are fixed, and only numerical parameters are allowed to evolve. GA, which has been widely used to solve optimisation problems, represents an individual as a fixed-length bit string chromosome. GA aims to decode the chromosomes to get the solution for the problem being faced by employing genetic operators. ES use fixed-length real-valued vectors to represent the individuals. Extending the idea of GAs, GP uses computer programs to represent the individuals. The evolutionary process in these algorithms involves random mutation, reproduction, and survival of the fittest by selection. While mutation is random, selection can be deterministic or stochastic in these algorithms.

SI algorithms are inspired of collective social behaviour of birds, ants and bees or other animal societies as agents. Performing an exploitation search, each agent attempt to discover a better solution. Interaction between the agents the whole group will learn the social behaviour. Two popular algorithms in SI are Particle Swarm Optimisation (PSO) and Ant Colony Optimisation (ACO). PSO, inspired of the movement of organisms in a bird flock or fish school, guides the swarm of candidate solutions in the search space based on their own and the entire swarm best known positions. By repeating the process and updating the position of the swarm, a good solution will be found. Compared with many global optimisation algorithms, PSO is well-known for its fast convergence. ACO, inspired of the foraging behaviour of ants in finding a path between their nest and a source of food, typically performs a model-based search and is suitable to solve discrete optimisation problems.

Since this thesis uses GAs, GP, and EMO, we will describe them in more detail in the next sections. As GP is mostly used in this thesis, first GP and its components are explained, followed by GA and EMO.

2.3.1 Genetic Programming (GP)

This section explains the important aspects of GP, such as evolutionary search process in GP, program representation, genetic operators, and GP fitness function.

Overview of Evolutionary Search Process

GP uses a variable-length individual representation to evolve population of computer programs to automatically build or evolve a model to tackle the problem. Being a stochastic search algorithm, GP generates an initial random population of individuals to search for the solution. During the evolutionary search process, individuals are modified by the set of genetic operators [119]. GP simulates evolution by employing fitness based selection where the fittest program is expected to be chosen. The computer structures

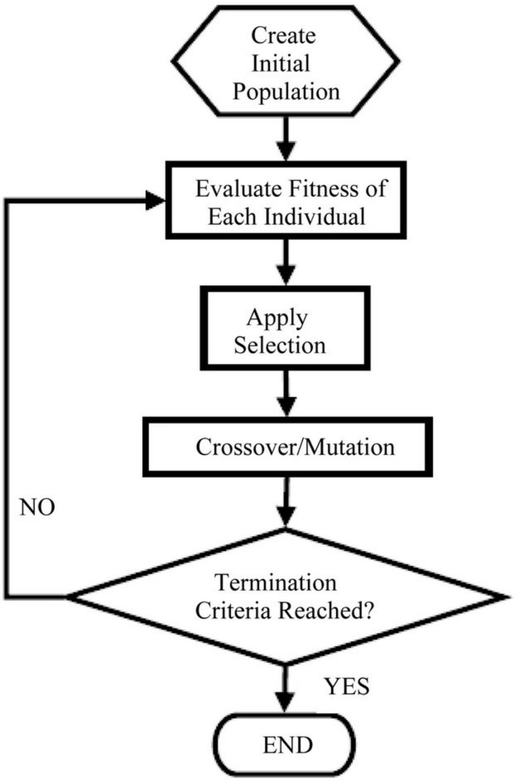

Create Initial Population
Evaluate Fitness of Each Individual
Apply Selection
Crossover/Mutation
NO
Termination Criteria Reached?
YES
END

Figure 2.9: The overall flowchart of a GP algorithm.

used in GP can be in the form of tree, linear and graph-based structure. The most popular GP structure is tree-based which composed of structural units namely terminal and function sets (Figure 2.9). The terminal set, which represents the leaves of the GP tree, provides the inputs to the individuals that may contain variables/features and constants. The function set represents the internal nodes and may consist of arithmetic operators, conditional operations or any user-defined operators. The overall structure of a GP algorithm [63] is illustrated in Figure 2.9 composing of the following steps:

1. *Initial population*: GP employs the function and terminal sets to generate a number of initial/candidate solutions.

2. *Fitness evaluation*: GP executes each individual (program) and evaluates the goodness of the individual using a user-defined fitness function.

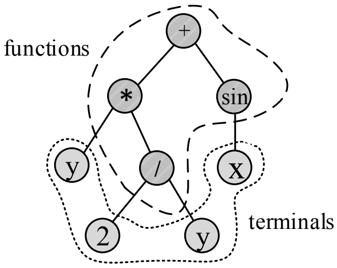


Figure 2.10: An example of a GP tree representing the function and terminal nodes.

3. *Individual selection*: Using a specific selection procedure such as fitness-proportional selection, truncation selection or etc., GP selects the individuals with higher fitness values for the reproduction process.

4. *Genetic operator*s: GP transforms the initial population by applying genetic operators (crossover, mutation and reproduction) to create new individual program(s) which are more likely to contain higher fitness values.

5. *Stopping criteria*: A stopping criterion determines when to stop the evolutionary process. The process can be stopped when an ideal individual with a specified fitness value has been found or when a maximum number of generations has been reached.

Program Representation

The GP individuals can be represented by the following ways:

- *Tree-based* GP. The tree structure is the most popular GP representation where the leaf nodes correspond to terminals, and non-leaf nodes represent functions. This representation is used in the design of the GP models developed in this thesis [63]. Figure 2.10 shows an example of a GP tree representing $\{+, *, /, sin\}$ as functions and $\{x, y, 2\}$ as terminals.

- *Linear* GP. In this representation, GP evolves the computer programs represented as a variable number of sequence of instructions from machine language [120, 121].

- *Graph-based* GP. In this structure, the computer programs in a population are represented as tree-based graphs where apart from function and terminal nodes, the flow of data in the tree is represented by edges of the graph. Cartesian GP uses this encoding scheme, allowing partial sub-trees to be re-used in program execution [122].

- *Grammar-based* GP. Grammar-based GP and grammatical evolution (GE) are the two popular GP approaches that use this representation where the evolved programs are represented as integer strings encoded through the use of a user-specific grammar [123].

As the tree-based representation is used in this thesis, more detail about this structure is given as follows.

Typically two main components of a GP tree are *function* set and *terminal* set where each contains elements to represent a node in th e tree. While a function set contains a set of operators which perform operations on their child(children) node(s), the nodes from the terminal set do not have any child. According to the problem which GP is used to solve, these sets are prepared in advance. The terminal set normally contains either a set of features (variables) or a number of constant values which are randomly selected by GP. The function set can contain a set of functions, including, but not limited to, arithmetic, transcendental, or trigonometric functions.

It is important to satisfy two main properties of *sufficiency* and *closure* when selecting the function set and terminal set [120]. If the two function set and terminal set have enough expressive power to be representative to the solutions of the problem, they will satisfy the sufficiency property. Moreover, when the operators in function set are capable to properly handle all possible inputs, the closure property is met as well.

Initialisation of the Population

The three main ways to initial the population are as follows [63]:

- *Full* method. Before reaching to a maximum allowed tree depth, the fully-formed trees are constructed by selecting the intermediate node only from the function set. Once the maximum depth is reached, the leaf nodes should be only from terminal set.

- *Grow* method. Unlike full method, the nodes can be selected randomly either from terminal or function sets. The growth of a sub-tree is terminated once a terminal node is selected. This method allows for constructing initial tree-based individuals with different depths and irregular shapes.

- *Ramped half-and-half* method. Utilising from both methods above to create each half of the population, ramped half-and-half allows for enhancing the diversity in the initial population.

Genetic Operators

GP has three main genetic operators, *Crossover*, *Mutation*, and *Reproduction* to create the new offsprings. The first two operators modify the genetic information in the parent individuals, whereas elitism create the new offspring without altering the original parent. Generally speaking, these operators are necessary to avoid having the same set of individuals in the population of the next generation. The rate or the probability assigned to each of these operators determine their importance during the evolutionary process.

Crossover Being the most predominant operator used in GP, the crossover operator (also called recombination) selects two parents based on the selection mechanism. In the tree-based GP, the sub-trees of the two parents at

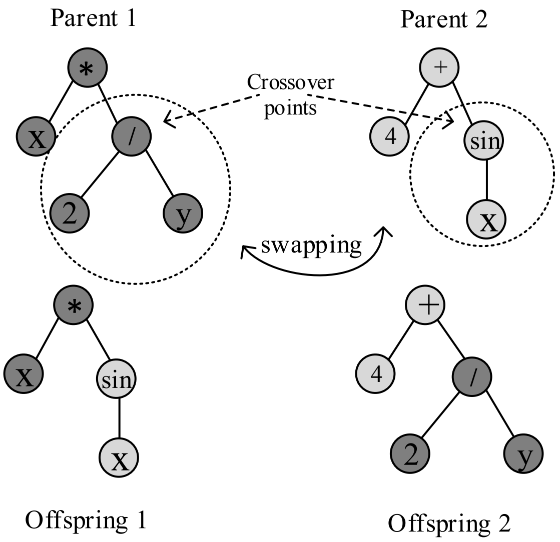


Figure 2.11: An example of a crossover operator in GP.

randomly selected points are swapped, generating two new children as shown in Figure 2.11.

Crossover implies the fact that fitter individuals tend to be selected more frequently than weaker ones and this is the major concept of “survival of the fittest” originated from Darwinian evolutionary theory where fittest individuals leave more copies of themselves in the next generations.

Mutation Mutation operator only select one parent (individual) to create the new child. In the case of the tree-based GP, as shown in Figure 2.12 mutation operator randomly selects a sub-tree and replaces it with another sub-tree generated by any of the initialisation methods such as full, or grow. Introducing new genetic makeup into the population, mutation operator is applied to maintain the diversity from one generation of the population to the next [63].

Reproduction Reproduction operator selects an individual based on the selection mechanism and simply copies that into the next generation. Al-

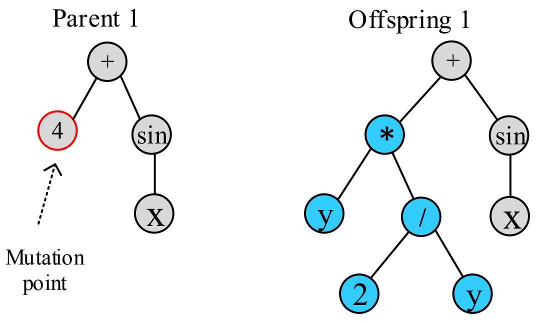


Figure 2.12: An example of a mutation operator in GP.

though there are some differences (due to the selection mechanism), reproduction is also known as *elitism* which protects the best individual(s) for the next generation[1]. As reproduction operator does not alter the parent individual, it ensures that the new offspring generated for the next generation does not have fitness worse than its original parent from the current generation. Normally the fittest individuals are selected for reproduction. Having this operator among the other genetic operators in the design of the GP method ensures that individuals of the population in the next generation are at least as fit as those from the current generation.

Evaluation

Evaluation of an individual or a candidate solution is done through the fitness function and that reflects the goodness of the individual in solving the problem [63]. With respect to the problem or the task, the fitness function is designed accordingly. For example, when GP is applied to solve a classification problem, *classification accuracy* or *classification error* rate can be used to measure the performance of the candidate solutions.

However, if GP is applied for regression problems, suitable performance measures in regression such as *absolute error* or *squared error* can be used [120]. More details about various performance metrics appropriate for classification

[1] This thesis does not distinguish elitism and reproduction, similar to [124].

and regression is explained in Section 2.4.1. The fitness of an individual determines how likely it is selected to be used by the genetic operators [125].

Selection

The selection operator provides the opportunity for the individuals to get passed to the mating pool for applying the genetic operators. Although normally better fitted solutions have more opportunities to get selected, it does not mean that less fitted individuals are excluded from the mating pool. Even the worst individuals still have the chance, but at a lower probability. The two most commonly used selection methods in GP algorithms are as follows:

- *Fitness-proportional selection.* Known as roulette-wheel selection, this operator potentially selects fitter solutions for recombination as the probability of selection is proportional to the fitness values of the individuals. Still each individual has the chance to compete with the rest of the individuals of population [125].

- *Tournament selection.* Regarding to a *tournament size*, a set of individuals are randomly selected to pass to the tournament. Then the fittest individual is picked up from the tournament. Here only the individuals in the tournament compete with each other. The larger tournament size, the more pressure on the selection which means less fitted individuals have less opportunity to be selected [124].

2.3.2 Genetic Algorithm (GA)

Inspired of Darwin's theory of evolution, Genetic Algorithms (GAs) were first introduced in 1960 by John Holland [117] and got extended afterwards in 1989 [126] by one of his students. GA starts its evolutionary process to find the candidate solutions by evolving a population of *chromosomes*. Typically in GA, chromosomes (or individuals) are represented as an array of bits. As

S	G	F	L	E	E	D	E	L	K

Figure 2.13: An example of an encoded GA chromosome (individual) to represent a sequence of amino acids.

this thesis uses GA for *de novo* sequencing, a relevant example of an encoded bit strings chromosome which is used to represent a sequence of amino acids is shown in Figure 2.13. Each element in this string is called a *gene* which stores an amino acid. GA evolves the population of individuals to search for the optimal solution. During the evolutionary search at each generation, selection operator, the main driving operator of GA, is applied on the population to breed a new generation. This operator serves for modelling the survival of the fittest (as measured by a fitness function). By applying other genetic operators namely crossover and mutation, the solutions of the next generation through those selected individuals are generated. The main concepts in GA such as genetic operators and selection are quite similar to those of GP which in previous sections are explained. Unlike traditional gradient search methods for optimisation, GAs are less likely to get stuck in local minimum. They, however, are time taken for convergence and need a decent sized population and sufficient generations to find the optimal solution.

2.3.3 Evolutionary Multi-objective Optimisation

The traditional evolutionary search process can be further extended to adapt the learning process to handle the conflicting objectives and this is called evolutionary multi-objective optimisation (EMO). Optimising multiple (usually conflicting) goals (or objective functions), multi-objective optimisation problems (MOPs) aim at generating a set of solutions capturing the best possible trade-offs (compromises) among objective functions.

Generally speaking, in an MOP, N_{obj} conflicting objectives are required to be optimised simultaneously, while a set of inequality and equality constraint functions are satisfied. Therefore, a multi-objective minimisation problem can be formulated based on Equation (2.4).

$$\text{minimise F(x)} = (f_1(x), f_2(x), .., f_{N_{obj}}(x)) \tag{2.4}$$

$$\begin{aligned} \text{subject to } g_i(x) &\leq 0, \quad i = 1, 2, ..., k \\ h_i(x) &= 0, \quad i = 1, 2, ..., l \end{aligned}$$

where $F(x)$ is an objective vector representing N_{obj} objectives. $f_i(x)$ is the i-th objective of $F(x)$ and x is the decision vector. $g_i(x)$ and $h_i(x)$ represent the inequality and equality constraint functions of the optimisation problem.

In single objective optimisations, the optimal solution is usually unique, while the optimal solution in MOP is often a set of non-dominated solutions due to the conflict between different objective functions. Since the quality of each solution is based on the compromise between objectives, knowing the concepts of dominance is necessary. A solution y dominates z (denoted by y $\prec$ z) if and only if:

1. $f_i(y) \leq f_i(z)$ for all f_i functions in F, and

2. there is **at least one** j such that $f_j(y) < f_j(z)$

The solution y is called Pareto optimal if it is not dominated by any other feasible solutions. The set of all Pareto optimals is called the Pareto front, representing the trade-off surface in the objective space. An EMO algorithm is expected to evolve a set of non-dominated solutions to approximate the Pareto front.

EAs are highly appropriate to approximate the Pareto front as they are capable of producing multiple Pareto-optimal solutions in a single run. Unlike traditional single objective EAs, EMOs often produce a set of optimal solutions rather than only a single solution. A number of EMO methods are proposed in the literature. The two main categories of EMO methods are *Pareto dominance-based* and *decomposition-based* methods [127].

In Pareto dominance-based algorithm, the performance of a solution is determined through non-dominated sorting which means that the algorithm assigns a rank (based on its Pareto dominance) to a solution on all objectives relative to all other solutions in the population [128]. At each generation all non-dominated solutions, which positively guide the search to convergence, are selected to form the population of the next generation. Three are three main categorises of Pareto dominance measures, dominance rank, dominance count and dominance depth. Dominance rank counts the number of individuals which dominate the current individual, while dominance count represents the number of individuals which are dominated by the current individual. Individuals can be sorted into fronts according depth by dominance depth. Pareto dominance-based algorithms also benefit from diversity promotion techniques, for example, crowding distance to maintain the diversity of the population [129].

Among Pareto dominance-based, Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [76] and Strength Pareto Evolutionary Algorithm 2 (SPEA-2) [130] are known to be standard approaches to solve MOP, although some other approaches are proposed in literature where more details can be found in [131]. NSGA-II is an extension of GAs for solving MOP which uses dominance rank to measure the goodness of individuals. The solution in the Pareto front evolved by NSGA-II tend to have the best fitness values (dominance rank) of zero which means they are not dominated by any other solutions in the population. SPEA-2 use both dominance rank and dominance count to evaluate the performance of each individual. First dominance count (strength) is calculated for each individual. Then, the dominance rank of each solution is counted as the summation of the strengths of all individuals that dominate the current solution. Therefore, the individuals in the Pareto front have the best fitness of 0.

Being a basic technique in traditional MOP, decomposition has recently attracted attentions in EMO. Decomposition-based methods decompose an MOP through decomposition method (e.g. weighted sum or Tchebycheff) into a set of scalar sub-problems and simultaneously optimise them by an optimisation algorithm [127]. Multi-objective evolutionary algorithm based on decomposition (MOEA/D) [132] is a popular decomposition based MOEA method [133]. For better convergence, MOEA/D uses evolutionary operators for combining good solutions of neighbouring problems. Using information from the neighbourhood makes MOEA/D having lower computational complexity than Pareto dominance-based algorithms such as NSGA-II [37].

MOEA/D

In order to find the set of non-dominated solutions for Pareto front approximation, MOEA/D decomposes an MOP into a set of N (equal to the population size) scalar objective optimisation sub-problems, each with the objective of the aggregation of all objective functions. MOEA/D attempts to optimise these N scalar optimisation sub-problems simultaneously instead of solving MOP directly in a single run. Tchebycheff is one of the most widely used decomposition approaches [132, 134]. In this approach, the fitness function of each single objective sub-problem is defined by a weight vector λ. This approach represents the j-th scalar optimisation sub-problem in the following form:

$$\text{minimise } g^{te}(x|\lambda^j, z^*) = \max_{1 \leq i \leq N_{obj}} \{\lambda_i^j |f_i(x) - z_i^*|\} \tag{2.5}$$

where $\lambda^j = \left(\lambda_1^j, ..., \lambda_{N_{obj}}^j\right)^T$, z^* is the reference point, and z_i^* in a minimisation MOP is the minimum value of each objective function. The major motivation behind MOEA/D is the concept of neighbourhood.

In this approach the neighbourhood of λ^j is defined as a set of the T closest weight vectors in $\{\lambda^1, \lambda^2, ..., \lambda^N\}$ and the Euclidean distance between these weight vectors defines the neighbourhood relation. It is expected that any information from the neighbouring sub-problems should be helpful for optimising the current sub-problem. In summary, each Pareto optimal point, x^*, with a weight vector of λ, which is the optimal solution of Equation (2.5), is a Pareto optimal solution of Equation (2.4) and belongs to Pareto front.

2.4 Performance Evaluation

In this section, the metrics for performance evaluation in classification, regression, and peptide and PTM identification problems are presented.

2.4.1 Classification and Regression Problems

The key component to evolutionary algorithms is the performance measure of candidate solutions. The evaluation process determines the goodness of an individual (or an evolved program, in GP) through the fitness function. Therefore, the evolutionary process is guided towards finding better solutions [63].

The widely used performance measures in **classification** problems include:

- *Classification accuracy* is the number of correctly classified instances as a percentage of the total number of instances.

- *Classification error rate* is defined as the ratio between the number of error predictions and the total number of predictions.

- *Confusion matrix* is a table which presents the number of correct and incorrect predictions made by the classification model compared to the target value in ground-truth data. It reports TP (true positive), TN (true negative), FP (false positive), and FN (false negative). More detailed analysis can be done using the information of this table. *Sensitivity* (also known as true positive rate) and *Specificity* (also known as true negative rate), which both used in this thesis, categorise the type of error made by a classifier and are calculated using Confusion matrix. While sensitivity calculates the proportion of actual positives that are correctly identified, specificity measures the rate of actual negative instances that are correctly identified.

- *Receiver operating characteristic (ROC)* plots the classification ability of a binary classifier system across various thresholds. The true positive rate is plotted in function of the false positive rate for different cut-off points of a parameter. Ideally, if the curve rises quickly toward the top-left, the model has correctly predicted the cases [135].

- *Area under ROC curve (AUC)* measures the quality of a classifier. The accuracy is measured by the area under the ROC curve. In practice, most of the classification models have an AUC between 0.5 and 1 (ideal).

The common performance measures in **regression** problems are:

- *Absolute error* sums the absolute values of differences between the inferred values and the desired values [136].

- *Squared error*, similar to the absolute error, calculates the sum of the squares of the differences.

- *Scaled error* can either amplify or damp smaller deviations from the desired output values.

- *Relative sum of squared error* compares the performance of the evolved model against the mean of the target values as the baseline model. This measure is used in Chapter 6 and is presented by Equation (6.3).

2.4.2 Sequence Comparison Metrics

There are mainly three common measures introduced in literature to evaluate the performance of the *de novo* sequencing algorithms. The metrics are recall at the amino acid level, precision at the amino acid level, and recall at the peptide level [11]. The three aforementioned metrics are also named as single residues recall, sequence tag recall, and full-sequence recall [43].

To evaluate the accuracy of the *de novo* sequencing methods, normally the predicted amino acid sequence (also known as *de novo* peptide sequence) is compared with the real peptide sequence (which is known to be a true peptide sequence). Therefore, the three metrics are defined as follows:

The recall at amino acid level metric is calculated based on the ratio of the total number of correctly predicted amino acids over the total number of amino acids in the real peptide sequence. This metric determines the average coverage of correctly predicted single amino acids.

The precision metric at the amino acid level is defined as the ratio of the total number of correctly predicted amino acids over the total number of amino acids in the predicted peptide sequence. This metric provides an overview of the average predicted peptide length.

The rigorous recall at the peptide level measure is a harsh metric and reflects the actual *de novo* sequencing errors, such as amino acid permutations. Being a full-sequence-based sensitivity criterion, this metric calculates the number of fully correctly predicted peptides divided by the number of real peptides. These three metrics are used in Chapter 5 and represented by Equation 5.9, Equation 5.10, and Equation 5.11,

2.5 Related Work

This section reviews methods related to the prediction of fragmentation patterns and *de novo* peptide sequencing. Some of them will be used in this thesis to compare with our proposed methods.

2.5.1 Preprocessing MS/MS Spectra

To overcome the problem of incomplete fragmentation and noisy data, a preprocessing step to denoise the MS/MS spectra and find signal peaks for more reliable peptide identification is usually performed. Generally, there are three types of methods to denoise MS/MS spectra: simple intensity-based thresholding [92, 8, 137, 138, 26, 27], peak detection methods inspired of digital signal processing [139, 140, 141], and machine learning algorithms [142, 29, 143].

Intensity-based Thresholding

Threshold methods are normally used in database search engines prior to searching in order to simplify the spectra by discarding peaks with intensity below an specific threshold. However, an optimal threshold value is hardly determined and varies from dataset to dataset. Moreover, these methods by only considering the intensity information of peaks and assuming independence of peaks, neglect the hidden interrelationship between them. In an MS/MS spectrum, signal peaks are related to each other, for example, the mass difference between two consecutive signal peaks may be equal to the mass of one of the 20 amino acids. Another example is the relation between complementary peaks which is mentioned in more details in Section 2.1.2.

Peak Detection

Peak detection methods such as Fourier analysis and wavelet analysis usually rely on the shape of the signals and assume stationary signals which are not the characteristic of signals in mass spectrometry spectra. For low quality MS/MS data where the peaks are not well-defined shape, these methods are considerably less effective [26, 144, 29].

Machine Learning Methods

Recently developing intensity-based models for peptide identification has become an attractive prospect and many efforts have been made by different groups. Zhou et al. [142] developed an intensity-based model using a Bayesian neural network approach. A set of 35 peptide fragmentation features were used to analyse the ion intensity pattern present in 13878 different tandem spectra. Based on the selected features, the intensity-based model was built in order to predict the intensity patterns for the given tandem mass spectra. The work done by Zhou was capable of identifying more contribution of different amino acids to peptide fragmentation during CID. However, both approaches did not model the peak intensities directly. They modelled the probability of observing a certain fragment ion intensity.

Cleveland et al. [28] proposed an approach which used a two-staged neural network (SNN) to model ion fragmentation patterns. The model estimated the posterior probability of each ion type. The main objective was to identify informative peaks particularly b- and y-ions. A total of 482,604 spectra for doubly charged peptides ranging from 8 to 20 residues was chosen as the dataset. The initial preprocessing step was expected to eliminate almost 50% of all noise peaks in the spectrum. So it removed all peaks with intensities less that 50. In the first NN, for each peak in the training dataset, a feature vector containing 14 features and a target vector denoting three different ion types were fed to the input layer of the NN for training and classification. For computing the classification error, the cross entropy error function was used as appropriate objective error function. In the second NN, the same 14 features along with the results of the first NN were given to the NN in order to reinforce positive evidence that the current peak is a signal peak. The SNN algorithm outperformed PepNovo [44] and pNovo [145] in terms of correctly identified signal peaks. However, ANNs cannot be interpreted and effectiveness of the SNN model by itself without applying the threshold method was not investigated.

Tiwary et al. [143] also developed two different regression strategies to model peak intensities using deep learning, DeepMass:Prism and wiNNer and applied them to the analysis of both data-dependent and data-independent acquisition datasets. Gessulat et al. [146] also trained a deep neural network, termed Prosit to predict the fragment ion intensities. Although both methods reported significant improvements in the prediction of fragment ion spectra, more work is required to clarify best integration of these models into the proteomics pipelines [147] and to investigate their impact on improving the results of particularly *de novo* sequencing algorithms.

Overall, a number of methods has been developed to predict the intensities of different fragment ions by training on a large dataset of experimental fragment ion spectra. However, predicting the relative intensities of these patterns are difficult as fragmentation is a stochastic process [147]. Moreover, many methods consider prior assumptions about the fragmentation model [142, 148], for example, a random match is more likely to have a low intensity peak [149], whereas even an informative b-/y-ion can be low abundant. Although using millions of MS/MS spectra to train deep neural networks has made these models to be more accurate than other methods in predicting ion fragmentations, they require very large amount of MS/MS spectra with known identification while in the case of unknown proteins and peptides in proteomics, such data might not be always available. Another main drawback of these models is their limited interpretability as deep learning models are known to be 'black boxes' [147].

2.5.2 *De Novo* Sequencing

A large numbers of *de novo* sequencing algorithms have been developed. They are mainly divided into the following groups: naïve approaches, spectrum graph models, dynamic programming, hidden Markov models, and machine learning based algorithms [150, 12]. Each group is explained in more details as follows.

Naïve Approach

Generating all possible sequences matching the measured precursor mass for the input spectrum is called brute force, or naïve, approach [151, 152]. The sequences are then scored against the spectrum and the sequence with the highest score is accepted as the correct identification. However, this method is only feasible for very short peptides, because the time complexity grows exponentially in terms of peptide length. PAAS [151], the first *de novo* sequencing method, generated more than 21 million possible peptides for an MS/MS spectrum shown in Figure 2.14 with precursor mass of 775. Also

Figure 2.14: CID mass spectrum of peptide “VYLHPF”, adapted from [151].

for a large amount of spectra, this approach is not reasonable in terms of time and space complexity. Many researches have been conducted on prefix pruning to speed up the search, but since the prefixes are poorly represented in the spectrum, such methods are not always successful in discovering the correct sequences [52, 51, 153]

Spectrum Graph Theory Models

Another approach to *de novo* sequencing is to generate a graph from an MS/MS spectrum and then finding paths in the spectrum graph that represent peptide sequences possibly giving rise to the spectrum [154, 155, 45]. A spectrum is represented as a graph, where peaks (m/z values) are the vertices and edges are defined as the corresponding amino acids to the mass differences between two vertices. The *de novo* sequencing algorithm in this approach is transformed to simply trace a full pathway through the directed acyclic graph to obtain the spectrum’s sequence [49, 156, 157]. A scoring function then will be used to score each vertex in this spectrum graph according to its previous supporting peaks. The identification process involves parsing the graph in order to find the highest scored path, which can be interpreted as the corresponding peptide sequence for the given MS/MS spectrum. Popitam [158] uses a non deterministic heuristic approach as an ant colony

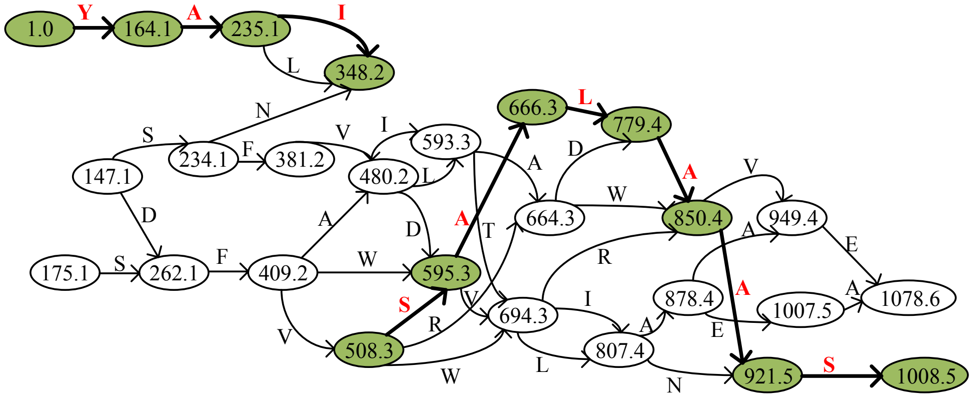


Figure 2.15: An example of an spectral graph reconstructed for *de novo* sequencing.

optimisation algorithm (ACO) for exploring the graph. Many other methods use dynamic programming to parse the graph. However, a spectrum graph approach has some major difficulties such as having a huge graph due to the noise peaks caused by internal cleavages, contaminants or PTMs. Another problem is the lack of full path due to the missing ion types caused by incomplete fragmentation and low instrument accuracy which makes the *de novo* sequencing of full-length peptides challenging (see Figure 2.15).

Dynamic Programming

Dynamic programming can solve a complicated problem by breaking it down into simpler sub-problems and solve them in a recursive manner [159]. Several tools such as PEAKS [9], PepNovo [44], Lutefisk [100], Sherenga [160], AUDENS [161], Eigenms [162], pNovo+ [163], MSNovo [46], UniNovo [47], Open-pNovo [48], MRUniNovo [164] used dynamic programming to traverse through the spectrum graph for finding optimal paths, although faster heuristics exist. MSNovo uses dynamic programming to find the best peptide among all possible peptides which are encoded by a mass array data structure [46]. Although dynamic programming can guarantee to find the optimal sequence, this result might not always represent the correct sequence.

Hidden Markov Models

Hidden Markov models are also used in NovoHMM to evolve a model for solving *de novo* sequencing by estimating the Bayesian posterior probabilities for each amino acid [54]. The model is constructed based on the observed mass peaks as the observable random variables and the unknown peptide sequence corresponded to the hidden variables. The inference with the model estimates posterior probabilities for the inferred sequence of amino acids rather than scores for single amino acid in the sequence. However, the model is highly dependent on the completeness of the fragment ladder. Incomplete or multiple fragmentations and random noises severely diminish the efficacy of the algorithm, leading to false positive predictions [46]. UVnovo used a random forest algorithm to learn interpreting ultraviolet photodissociation (UVPD) spectra and passed the results to a hidden Markov model for presiding and scoring the sequences. However, UVnovo is a special-purpose program and not generally applicable [43, 165].

Machine Learning Methods

Recently, machine learning based algorithms are used for automated *de novo* sequencing. Decision trees have been used to model the spectra intensity patterns of given peptides. An intensity-based scoring model using two probabilistic decision trees to model the fragment ion intensities, learning from a library of MS/MS spectra was proposed in [55]. Similar to this work, two new scoring functions based on two large decision trees prior to *de novo* sequencing were used in Novor [10]. The decision trees were learnt from a training data composed of more than 300,000 spectra. However, producing huge decision trees with 7,000 and 14,000 nodes have potential to misclassify the unseen new data.

Recently, deep learning has been introduced in this area to enhance both accuracy and speed of de novo sequencing. DeepNovo algorithms [11, 15], based on deep neural networks and local dynamic programming, show significant improvement in full-length *de novo* sequencing over their competitors PEAKS [9], PepNovo [44], and Novor [10]. Although the two algorithms use a local version of dynamic programming that filters inappropriate amino acid sequences and does not perform backtracking, the methods still can be further enhanced with more advanced search algorithms as the authors of DeepNovo have mentioned.

Comparisons of Multiple State-of-the-art De Novo Sequencing Algorithms

Muth et al. [43] selected three dominant *de novo* sequencing methods, Novor [10], PepNovo [44] and commercial PEAKS software [77] for the purpose of providing a detailed evaluation on the performance of algorithms on two HCD and two CID datasets. Their experimental results showed that the overall performance of Novor, PEAKS and PepNovo on peptide level were 32.25%, 29.25%, and 18.25%, respectively. Clearly, Novor outperformed PepNovo, but not very far from PEAKS. Moreover, Novor and PEAKS showed better performance for HCD comparing with CID datasets. This indicates that both algorithms could take advantage of the high resolution of HCD fragmentation.

Tran et al. [11] compared the performance of DeepNovo [11] with PEAKS [77], PepNovo, and Novor on five CID spectra on peptide level. The overall performance of DeepNovo on the five datasets showed a significant improvement at 25.81% in comparison with the other methods. PEAKS and Novor almost achieved the same overall accuracy at 16%, both outperforming PepNovo by 6%. Moreover, on 9 HCD datasets, the results provided by Tran et al. showed overall accuracies of 37.6%, 15.4%, 32.22%, and 2.8% for DeepNovo, Novor, PEAKS and, PepNovo on peptide level, respectively. The results showed that DeepNovo and PEAKS were two main competitors with each other on HCD data.

Qiao et al. in the design of DeepNovoV2 [15] used a new representation for the input spectrum along with a new architecture of the deep model.

They only evaluated the method on the same 9 HCD datasets and compared the results with PEAKS [9] and DeepNovo [11]. The experimental results showed that DeepNovoV2 achieved an average accuracy of 43.93% across all nine HCD datasets, outperforming DeepNovo and PEAKS by average of 6.3% and 28.4% at the peptide level, respectively.

However, among all these *de novo* sequencing algorithms, PEAKS as a commercial software is the most commonly used tool in the proteomics community. Therefore, in this thesis we use PEAKS for further analysis and cross comparison.

2.5.3 PSMs Post-processing

Given an MS/MS spectrum, typically peptide identification methods generate a set of candidate sequences with each having a match score indicating the confidence of the match. Normally, the highest score sequence is considered as the best match. However, the best match does not always indicate a correct match. As high confidence peptide identification increase the confidence of protein inference, normally database searching methods perform a post-processing step to evaluate the correctness of PSMs. The input to post-processing algorithm is a list of high-scored PSMs and the output is the high-confidence PSMs.

Keller et al. [166] used 4 scores computed by the SEQUEST database search algorithm [92] as input features to a linear discriminant analysis classifier in PeptideProphet. The classifier was trained on correct and incorrect PSMs dataset derived from a purified sample of known proteins. The main disadvantage of PeptideProphet is the strong assumptions that were made on the model including a score function learnt from a small dataset and the Gaussian distributions presumption of correct and incorrect matches. Anderson et al. [167] applied similar approach using various features and used SVM as the classification algorithm.

Käll et al. [168] developed Percolator that utilized a richer feature representation for PSMs by maximising the use of the information provided by the database search. They employed an SVM-based two-step strategy in an iterative manner. First, the algorithm selected an initial set of PSMs, which were generated based on the SEQUEST cross-correlation score, as positive samples. Also a negative set was generated based on the concept of decoy PSMs. The first SVM classifier was developed to learn from this initial labelled set. Then in an iterative manner, the algorithm re-ranked the entire PSMs using the first classifier to select a new set of positives and then trained the second SVM in the new set. Being largely heuristic is the main disadvantage of Percolator. Moreover, what exact loss function Percolator optimises is not clearly provided.

However, there have been only a few attempts to develop post-processing algorithms for refining the results of *de novo* sequencing. Many *de novo* sequencing methods focus on improving the scoring function inside the sequencing algorithm [59, 10, 48] rather than developing a separate post-processing algorithm for further improving the results.

Yang et al. developed pSite [13] to solve amino acid confidence evaluation and modification site localisation on the results of *de novo* peptide sequencing. SVM was used to learn how to discriminate correct amino acids from random one with and without modification. Then, a Bayesian model was used to evaluate the false amino-acid rate (FAR) at any given threshold. Their experimental results on three test data sets showed that pSite recalled 86% more amino acids on average than PEAKS at the FAR of 5%. However, their method focuses on evaluation the confidence of each amino acid which means partially correct *de novo* peptide sequencing rather than full-length *de novo* peptide sequencing.

In pNovo3 [169], along with a *de novo* sequencing method to produce top-10 candidates (pNovo) and a deep learning model (pDeep) to predict the theoretical spectra, a new learning-to-rank framework was also developed to discriminate similar peptide candidates for each spectrum. Six features, extracted from a set of PSMs, were used as the input to SVM-rank [170] to build a model for re-ranking top-ranked peptide candidates. Then, a further step,spectrum merging, was applied to refine the top-1 results. On seven HCD datasets, pNovo3 outperformed pNovo, PEAKS, and Novor with average accuracy of 56.3% at the peptide level. PEAKS, pNovo, and Novor achieved the average accuracy of 36.63%, 35.23%, and 14.17%, respectively. Although pNovo3 showed significant improvement over the other methods, the effectiveness of its learning to rank framework by itself is not evaluated. Moreover, the motivation of the authors is discovering important features which are most useful for *de novo* sequencing to distinguish between correct and incorrect peptides, but the new scoring model build by SVM-rank does not reveal the relationship between the features fed into SVM-rank, and still it is a black-box model.

2.5.4 GP for MS/MS Analysis

GP has shown a great potential to deal effectively with the challenges in MS data to solve the problems such as biomarker detection [171], peptide detection for biomarker verification [172, 173], quantitative analysis [40], feature selection and classification [174, 175]. Moreover, GP was successfully used in motif discovery and cleavage site prediction from amino acid sequence of proteins [176, 177, 178]. However, the capability of GP in the analysis of MS/MS data, particularly for the peptide identification problem has not been fully investigated.

Dorfer et al. [179] proposed a white box modelling based on symbolic regression using GP for target-decoy classification on the results of their database search engine, MS Amanda [180], to calculate updated scores of PSMs. Pearson's correlation coefficient was used as GP fitness function with a terminal set containing a set of spectral and peptide specific features, and a function set of arithmetic and logical operators. Their experimental results on a low resolution dataset showed that the GP-based scoring function outperformed random forests (RFs) scoring function and significantly improved the results of peptide identification by 14%, while RFs increased the performance by up to 4%. However, using Pearson's correlation imposes a prior assumption of a linear pre-defined model structure. Moreover, as the motivation of this work was using a white box method for modelling the separation of correct and false identification, no detailed analysis of the generated models was provided.

2.5.5 GA for Sequence Optimisation

De novo sequencing can be formulated as an optimisation problem where the objective is to discover the most likely amino acid sequence that can be generated by the input spectrum [49]. *De novo* sequencing was performed via stochastic optimisation using genetic algorithms [181]. Unlike the naïve approach, this method does not need to generate all possible sequences, instead a small set of amino acid sequences is sufficient to start the process, optimising them to best fit the input spectrum. However, the algorithm failed as the simple fitness function used in GA was not discriminative enough. Later on another GA-based method with better fitness functions that got advantage of shared peak count was developed [182]. Given an MS/MS spectrum, a small initial population of amino acid sequences as individuals was generated randomly. Each individual was represented by a vector of integers between 1 and 18 (considering 20 amino acids with two identical pairs I/L and K/Q). In this algorithm, GA created 350 individuals in each of 150 generations. A fitness function, which measured the similarity between the experimental spectrum and the theoretical one produced by a candidate individual, was adapted for GAs. The candidate individuals were manipulated using mechanisms of recombination, selection and mutation until some pre-defined criterion of convergence such as certain number of generations or the maximum fitness has been met. The performance of the proposed method in situation when there are missing peaks were evaluated and the results were compared with those obtained by Lutefisk [100]. The results showed that Lutefisk was not able to find the correct peptide in majority of cases and the GA outperformed it. The GA-based *de novo* sequencing algorithm could potentially overcome the problem of missing ion peaks, one of the major problems in real MS/MS spectra, so this methodology can be a promising approach in the proteomics field. The work was extended by Malard et al. [183, 184] as *de novo* sequencing via constrained multi-objective optimisation using a Pareto-driven GA with two scores namely H-score and T-score, which were considered as the conflicting objectives to evolve a population of putative peptide sequences. H-score counted the number of matches between the peak locations in two experimental and theoretical spectra, whereas T-score measured the probability of a match being incorrect. However, further analysis on the quality and biological relevance of the evolved peptide sequences were not provided by the authors. PepyGen [185] was developed as a new post-processing algorithm for optimising the results of existing *de novo* sequencing algorithms. The partial sequences, tags, from Lutefisk were used as the input to PepyGen to reconstruct the full-length sequences and only two MS/MS spectra were used to check the performance of the algorithm. Overall, these GA algorithms showed promising results, but they only focused on improving the fitness functions of GAs while the impact of designing domain-dependent genetic operators was totally neglected in these methods.

2.5.6 Multi-objective Optimisation in MS/MS

There have been a very limited number of studies that used MOP in MS analysis for biomarker detection [186]. 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.

In MS/MS analysis MOP has not been fully investigated yet. Popitam proposed by Hernandez et al. [158] used parallel multi-objective GP (MOGP) to generate scoring functions for optimising the scores of the PSMs from the results of a Full Path algorithm which used ant colony optimisation to find the full paths in the spectrum graphs. A set of 12 scenario subscores each calculated based on a particular criterion such as sequence coverage, node pertinence, and peak intensity was extracted and fed to GP as its terminal set along with a random coefficient. The function set of GP is compromised of six mathematical operators addition, subtraction, multiplication, protected division, power, and a conditional statement if-less. The two fitness scores rankFitness and discFitness were considered as the function objectives for MOGP to optimise. RankFitness measures the ability of the GP evolved solution to give the highest score to the correct peptide and compromises between 0 and 1 (the optimum value), while discFitness evaluates how capable is the solution in distinguishing the correct PSM from other incorrect ones and varies between 0 (the optimum value) and 1. discFitness corresponds to mean of p-values computed based on score distributions of negative or random peptides. They reported that the results were promising as the new scoring function was able to distinguish the correct PSMs, resulting in improving the performance of Popitam. However, it is not clear that the two objectives considered in MOGP are in conflict with each other. Moreover, apart from the pre-assumption of Gaussian distribution by discFitness score, it was observed by the authors that quite often many random matches (incorrect peptides) receive p-values near 0, which is the optimum value, resulting in contamination of the distribution. As the motivation of the proposed GP method was building a scoring function which is able to give a good rank to the correct peptide while separating the correct peptide from other negative peptide scores, new criteria to separate correct assignments from incorrect ones is required.

2.6 Summary

This chapter presents basic notions about mass spectrometry-based Proteomics analysis. A brief background in machine learning with particular emphasis on classification and regression problems is also provided. As this thesis takes advantage of evolutionary computation (EC) techniques, some detailed background on genetic programming (GP), genetic algorithm (GA), and evolutionary multi-objective optimisation (EMO) are also covered in this chapter. The recent related works in MS/MS preprocessing, *de novo* sequencing and PSM scoring including traditional statistical and machine learning based method are reviewed as well. There are a few open issues that have not been explored yet:

- Despite the recent improvements in mass spectrometers and the reliability of peptide and protein identification tools, a number of studies revealed that state-of-the-art only assigned less than 40% of spectra to peptides, and still a significant number of MS/MS spectra remained unassigned [5, 187].

- Existing denoising models due to their black box nature have not fully investigated the important spectral features in MS/MS that can be used to learn fragmentation patterns [44, 145, 28]. Other methods need large amount of data to learn the fragmentation patterns [143, 146]. While over-fitting can be caused by performing classification on the small datasets, there are also limitations in proteomics on generating large benchmark MS/MS datasets with known identification, therefore the classification algorithm should be able to learn from a reasonable size of MS/MS dataset. Some methods mainly focused on the probability of observing a particular ion intensity which is a challenging task due to the stochastic nature of the fragmentation, rather than predicting the ion types. Moreover, as the number of signal peaks is very small compared to the noise peaks, the imbalanced property of MS/MS spectra and its impact on the accuracy of the machine learning model used to denoise/classify the peaks in order to learn the fragmentation patterns are not systematically investigated. As the imbalanced ratio varies between the MS/MS dataset, a stable learning method to various S/N ratios is required. Since learning from an imbalanced dataset is a very challenging task, multi-objective optimisation is a great technique to deal with the accuracies of the minority and majority classes separately. To the best of our knowledge there has not been any multi-objective algorithm to learn the fragmentation patterns from the imbalanced MS/MS data.

- *De novo* sequencing of *full-length* peptides is still a challenging task due to the missing ion types and presence of noise in the data. Many existing methods use either graph-theory or dynamic programming to find the full-length paths which in either cases the performance of the *de novo* sequencing algorithm deteriorates when facing missing values or noise. Also other methods mainly focus on improving the scoring functions used to score the candidate sequences, and neglect the search mechanism to find the correct candidates. GA was used to solve the optimisation task of *de novo* sequencing due to its ability to explore a large search space. However, current GA-based *de novo* sequencing methods neglect the flexibility of GA in adapting with domain-dependent knowledge and did not consider domain-specific genetic operators to enhance the quality of the evolutionary search process. It is worth investigating how designing appropriate GA components such as genetic operators, fitness function, and selection operator can help GAs construct full-length peptides.

- Developing PSM post-processing algorithms to refine the results of *de novo* sequencing is relatively a new research area and recently has attracted more attention. White/grey box models such as GP are more preferred as the researchers are more interested in discovering the relationship between the features that account for separation of correct and false identification. The current GP methods either solve a classification problem (a learning to rank framework) or a regression problem to generate the new PSM scoring function to optimise the scores of PSMs. However, these methods can be further improved by designing a strategy that enables GP to solve a classification and a regression problem simultaneously in order to generate a powerful discriminative scoring function that gives the highest score to the correct PSM among the other false identifications, resulting in improving the performance of the *de novo* sequencing method at the peptide level.

This thesis aims at exploring apparent limitations and suggests applying GP and GA to conventional *de novo* peptide sequencing methods in order to achieve the best possible identification accuracy at the peptide level.

Chapter 3

Preprocessing MS/MS Using GP for Improving the Reliability of Peptide Identification

3.1 Introduction

In MS/MS data, the background noise does not necessarily mean noise from low instrument accuracy, but anything which makes the peptide identification tool having a big search space should be removed from data. Therefore, a preprocessing step in a binary classification manner to remove background noise prior to *de novo* peptide sequencing in order to simplify the highly imbalanced MS/MS spectra could be useful for more high-confidence identifications.

As there is no gold standard for highly imbalanced MS/MS data containing already labelled signal and noise peaks, which can be used by the supervised classification methods, it is worth investigating which ion types should be considered as signals in the training set of the gold standard dataset.

Since the MS/MS data is highly imbalanced, developing a suitable classification algorithm that can handle this problem and also finding appropriate evaluation matrices for measuring the performance of the classifier are of great importance. One of the main advantages of GP in classification tasks is its flexible representation. GP has the potential to cope with complex problems and has good learning capability even from imbalanced data [35]. GP can adapt its fitness function to evolve a model that is capable of dealing with the class imbalanced problem. Moreover, various spectral features and fragmentation rules are introduced in the literature and their effectiveness in improving the peptide identification is not systematically investigated. Since GP has the capability of implicit feature selection, analysis of a GP model can reveal the important spectral features that have better discrimination ability.

GP has been successfully applied on MS data and proved to be a promising tool in MS analysis [188, 175, 40]. However, its potential for peptide identification from MS/MS spectra has not been systematically investigated.

3.1.1 Chapter Goals

The overall goal of this chapter is to develop a GP approach to preprocessing MS/MS spectra in order to reduce the noise peaks and to retain the signal peaks for the purpose of improving the reliability of peptide identification. Specifically, the following research objectives will be investigated:

- Investigating appropriate measures for performance evaluation of the preprocessing method to classify imbalanced MS/MS data;

- Developing an effective fitness function that accounts for both the minority and the majority class accuracies in the evolved GP classifiers;

- Investigating important ion types in peptide identification in order to create a suitable gold standard MS/MS dataset that can be used by GP for the training purposes;

- Investigating the stability of the proposed GP method across various imbalance ratios;
- Investigating the effectiveness of incorporating various spectral features and fragmentation rules as features into the GP method and analysing the capability of GP in implicit feature selection; and
- Analysing the effectiveness of the proposed GP method in terms of the improvement in the reliability of peptide identification with existing peptide identification tools.

3.1.2 Chapter Organisation

The remainder of the chapter is organised as follows. The proposed GP-based preprocessing method is explained in Section 3.2. Section 3.3 explains the experiment design where MS/MS datasets, benchmark algorithms considered for comparisons, and a set of experiments are described. Section 3.4 presents the results of the experiments and provides further analysis on the best evolved GP program. The chapter ends with Section 3.5 summarising the findings.

3.2 The Proposed GP Method

Figure 3.1 illustrates the MS/MS analysis workflow designed for preprocessing MS/MS spectra with GP followed by an evaluation step. The workflow starts with an MS/MS spectra (in the Mascot generic peak list format) dataset containing noise and signal peaks. The preprocessing method is composed of three steps including feature extraction, labelling peaks and classification using GP. In feature extraction, a set of intensity-based spectral features are extracted for each peak in the spectrum. The next step, labelling peaks, determines the class label of each peak as either signal or noise. The data then is divided into a training and a test set. GP uses the training

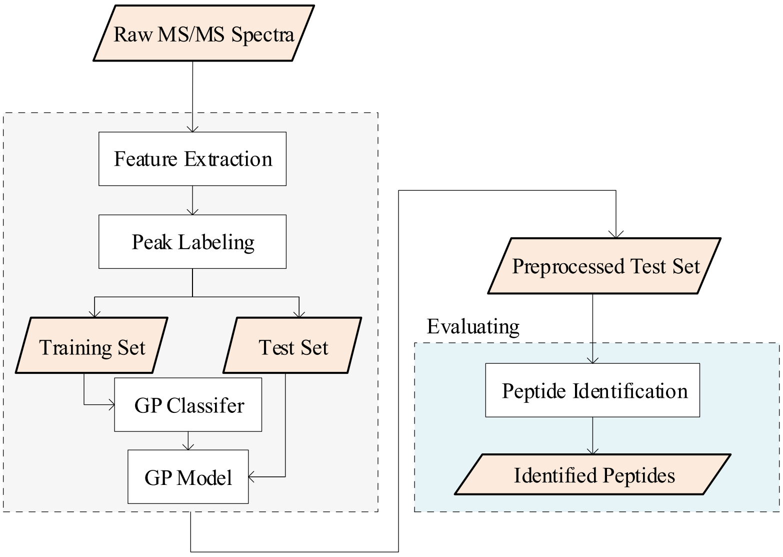


Figure 3.1: The MS/MS analysis workflow composed of the proposed GP method for preprocessing the spectra and an evaluating step.

set to build the model for binary classification then applies the model for classifying the peaks in the test set into signal peaks and noise peaks.

The preprocessed test set by GP is then submitted to PEAKS to evaluate the effectiveness of the GP method. The result of PEAKS is a set of identified peptides with different confidence scores. It is worth mentioning that, in peptide identification with PEAKS, an average local confidence score (ALC) indicates the reliability of the results. An ALC score reflects the average correct ratio of the predicted amino acids in a peptide sequence. The higher the confidence score, the more reliable the peptide identification. In PEAKS, ALC scores range from 0% to 99% and a score at 55% or above, as suggested in the PEAKS website [189], is considered as a confident match. The entire sequence of a peptide is not necessarily to be mapped due to the incomplete fragmentation and the difficulty in detecting the signal peaks of the fragments from the beginning and the end of the peptide sequences in MS/MS. The results of peptide identification are grouped into five intervals which are {[55, 60), [60, 70), [70, 80), [80, 90), [90, 99]}. For each interval, the number of peptides identified by PEAKS are counted.

Table 3.1: List of spectral features. N denote a normalised value; D specifies a discretised value; B denote a binary value;

Group	features	Feature	Value
(1)	{f1}	Normalised m/z	N
(2)	{f2}	Normalised and Discretised Intensity	N,D
(3)	{f3,...,f15}	Is Top X in Win ± Z	B
(4)	{f16,...,f28}	Local Rank in Win ± Z	N
(5)	{f29}	Global Rank	N
(6)	{f30}	Complementary Ion	B
(7)	{f31,...,f40}	Sister Ions	B

PEAKS is also used to perform *de novo* sequencing on un-preprocessed (raw) data and on the spectra preprocessed by an intensity-based threshold method. The results of peptide identification preprocessed by GP are then compared to those of un-preprocessed data and the intensity-based threshold method.

3.2.1 Feature Extraction

The intensity value of each peak in an MS/MS spectrum can be used to extract a set of spectral features that explain the CID fragmentation properties of peptides. Table 3.1 presents a total number of 7 groups of spectral features extracted from the MS/MS data. These spectral features can be good discriminators between the signal and noise peaks. All groups include only one feature except group 3, 4 and 7, which contain parametric features where changing the parameter values result in new features.

Given spectrum s with n peaks and precursor mass of m_{prec}, let $s = (mz_{(1)}, mz_{(2)}, mz_{(3)}, ..., mz_{(n)})$ denotes a spectrum with an intensity vector of $I = (I_1, I_2, I_3, ..., I_n)$. The i-th peak in the spectrum corresponds to the mass-to-charge value of $mz(i)$ with $intensity(i)$. More details about how to extract the features of each group from the spectrum are explained as follows:

Group (1): “Normalised m/z” feature [29] normalises the m/z value of each peak to an integer value between 0 and 100 by measuring the relative location of the current m/z value in the whole spectrum (see Equation (3.1)).

$$f_{norm_{mz}}(mz_{(i)}) = \left\lfloor \frac{mz(i) \times 100}{m_{prec}} \right\rfloor \tag{3.1}$$

Group (2): The "Normalised and Discretised Intensity" feature [28] divides the intensity value of the current peak to the highest intensity value in the whole spectrum. The intensity values within the whole spectrum normally are very fluctuated from a very small value of less than 100.00 to a large value of 10,000.00. Therefore, to have a better scaled values, discretisation is applied on the normalised values in order to map them into m discrete bins. For example for $m = 5$ in Equation (3.2), the normalised intensities are rounded up to 0.05, 0.10, 0.20, 0.40, 0.80, or 1.00 as discrete values [28].

$$f_{norm_{intensity}}(I_i) = \left\lfloor m(\frac{I_i}{I_{max}}) \right\rfloor / m \tag{3.2}$$

where I_{max} is the most abundant peak in the spectrum, I_i is the intensity of the current peak and m indicates the number of discrete intervals.

Group (3): The "Top X in Win ± Z" features [27] rank all peaks within the windows size of ± Z" around the current peak. If the current peak is amongst the top X most intense peaks in the window, then its corresponding feature value is considered to be 1, otherwise 0. The value of X and Z can be determined empirically or based on the literature. It is worth mentioning that overlapping windows are allowed in all window based features.

Group (4): The "Local Rank in Win ± Z" features [27] rank the number of peaks that are the same or are more abundant than the current peak within a local window of ± Z. Normalised ranks are computed by dividing the rank of each peak by the number of peaks available within the window ± Z.

Group (5): The "Global Rank" feature, presented by Equation (3.3), ranks the intensity of the current peak compared to all of the peaks in the whole spectrum and then normalises the rank by dividing the rank of the current peak to the total number of n peaks in the spectrum.

$$f_{norm_{rank}}(I_i) = \frac{rank(I_i)}{n} \tag{3.3}$$

Group (6): The "Complementary Ion" feature investigates if the complementary ion of the current peak exists in the whole spectrum. In CID fragmentation, a complete peptide fragmentation gives a contiguous series of ions. However, sometimes due to the low ion fragmentation efficiency of the mass spectrometer, some ions are not available in the spectrum. By finding the complementary ion peaks, undetected ions can be added to the spectrum. As shown in Equation (3.4), the sum of the two complementary ions' masses should be equal to the precursor mass of the spectrum. Therefore, each peak in the spectrum is checked for the existence of its complementary peak. Based on the CID fragmentation parameters of the dataset, a mass tolerance is considered to estimate the existence of the complementary ion of the current peak.

$$f_c(mz_{(i)}) = \begin{cases} 1, & \text{if } mz(i) + mz(j) + \delta \simeq m_{prec} \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

where $1 \leq j \leq n$ and n is the total number of peaks in the spectrum. δ is the mass tolerance of the mass spectrometry device used to ionise the spectra in the dataset and varies from one dataset to another one.

Group (7): The "Sister Ion" features check the existence of the sister ions of the current peak. A sister ion is a peak that can be found at the fixed Δ m/z value away from the current m/z value. Based on the literature [28], a list of 10 common sister ions including Δ values of $\Delta = \{-2, -1, 1, 2, 17, 18, 28, 34, 35, 36\}$ are considered in this study.

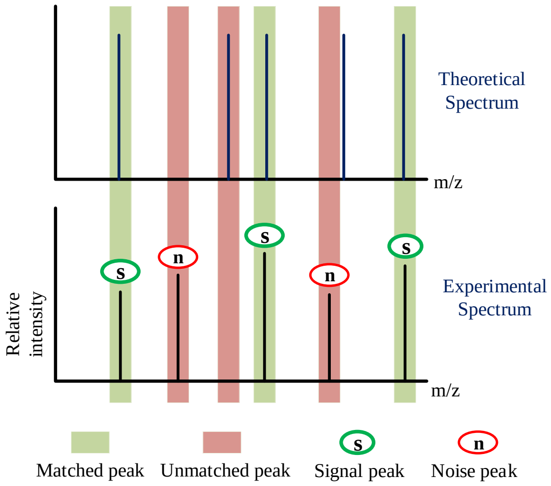


Figure 3.2: An example of the labelled peaks in the experimental spectrum by matching against the theoretical spectrum.

These numbers are related to the loss mass of H2O, NH3, H2O-H2O, H2O-NH3, and isotopic ions. This set can be extended to a larger range of all possible Δ values from -2 to 143 (145 sister ions).

3.2.2 Labelling Peaks/Instances

The so-called ground-truth is the spectrum peptide matches. The theoretical spectra of the known peptides based on the CID fragmentation rules of doubly charged peptides [190] are constructed. The theoretical spectra include only signal peaks with no noise peaks. More details about how to construct the theoretical spectra and which fragment ions should be considered are experimentally investigated in Experiment I of Section 3.3.3 (see Page 89) and the results are explained in Section 3.4 (see Page 95).

After constructing the theoretical spectrum, each peak in the experimental spectrum with those in the theoretical one is matched. Figure 3.2 illustrates the process of matching peaks in both experimental and theoretical spectra. A peak in the experimental spectrum within 0.8 Da of the signal peaks in theoretical spectrum is considered to be matched so it can be manually labelled as a signal peak. However, those unmatched peaks in the experimental spectrum are labelled as noise peaks. The mass tolerance of 0.8 Da is the mass error tolerance of the mass spectrometry used for producing the benchmark dataset used in this thesis [78].

3.2.3 Creation of the Training Set and Test Set

After applying the labelling process, the dataset consists of instances (peaks, m/z values), a set of extracted features followed by class labels which are presented by numerical values of 0 (for noise peaks) and 1 (signal peaks). Then, the data is divided into two sets: training set and test set. The training set is used during the GP learning process to build the model and the test set is used to evaluate the GP model.

3.2.4 GP Program Representation and Classification Strategy

As mentioned before, for preprocessing the MS/MS spectra, a binary classification using GP for classifying signal peaks and noise peaks is performed. A tree-based GP representation which is the most popular GP structure [63] is used. The terminal set of the GP method comprises of the extracted features and randomly generated floating point numbers, which are known as constants. The function set consists of the four arithmetic operators $+, -, \times$, and protected / (indicates usual division except that a division by zero gives a result of one) and the trigonometric *sin* function. The *sin* function is considered due to the periodic waveform of MS/MS spectra which might be turned into a sum of different amplitude sine waves. The first four operators take two arguments, whereas the last operator, *sin*, takes one argument. All five operators return one argument. The output of the GP program is a single floating point. It is worth mentioning that the function set considered in the design of GP is not obtained based on many trial and errors, therefore it is

Table 3.2: Genetic programming parameters

Parameter	Value
Function Set	$\{+, -, \times, /, sin\}$
Terminal Set	{Features from dataset, Random Constant}
Initial Population	Ramped Half-and-Half
Population Size	1024
Generations	50
Mutation Rate	0.19
Elitism Rate	0.01
Crossover Rate	0.8
Selection	Tournament, Size = 7

not the optimal function set for GP.

A threshold output value of zero is chosen to distinguish between the two classes. Based on the class distribution of an imbalanced MS/MS dataset, the signal class is considered to be the *minority class*, while the noise class indicates the *majority class*. A positive GP-output (including zero) value of the GP-tree indicates that the instance belongs to the signal class (minority class) and a negative output value points to the noise class (majority class). Table 3.2 displays the genetic programming parameters used in this work. The Evolutionary Computation Java-based (ECJ) package [191] is used for implementing the GP method.

3.2.5 An Effective Fitness Function for Imbalanced MS/MS Data

The key component to an evolutionary algorithms is the performance measure of candidate solutions. The evaluation process determines the goodness of an individual (or an evolved program in GP) through the fitness function. Therefore, the evolutionary process is guided towards finding better solutions. The number of correctly classified instances as a percentage of the total number of instances, i.e., the overall classification accuracy, is a widely used performance measure in classification problems. However, in the case of an imbalanced dataset, using such a measure as the fitness function leads the evolved classifiers to be biased towards the majority class [192]. Therefore, to avoid this issue as the MS/MS spectra dataset is highly imbalanced, a weighted fitness function including the true positive rate and true negative rate with coefficient of α is used to evaluate the evolved GP programs.

$$\begin{aligned} A\text{-}acc &= \alpha \times (\frac{TP}{TP+FN}) + (1-\alpha) \times (\frac{TN}{TN+FP}) \\ &= \alpha \times SE + (1-\alpha) \times SP \end{aligned} \tag{3.5}$$

where *A-acc* is the abbreviation of average accuracy. The signal peaks are considered as positive instances and noise peaks as negative examples. Therefore, TP is the number of correctly classified signal peaks, whereas FN indicates the number of incorrectly classified signal peaks. Also, TN is used to count the number of correctly classified noise peaks, and FP presents the number of incorrectly classified noise peaks. *A-acc* represents a weighted sum function composed of accuracy of the minority class (*sensitivity*, *SE*) and accuracy of the majority class (*specificity*, *SP*). α is a coefficient that needs to be determined empirically. This is addressed in Experiment III of Section 3.3.3 and the results are presented in Section 3.4.1 (see Page 101). Considering a weighted accuracy rather than an average accuracy prevents the bias issue when there is a prevalence relationship between the positive class and the negative class.

3.3 Experiment Design

3.3.1 Dataset

Table 3.3 presents the datasets used to run the experiments in this chapter. Each dataset contains different number of MS/MS spectra which are

Table 3.3: Datasets details

Datasets		No. of spectra	No. of peaks
Synthetic dataset	train	10	9,958
	test	5	4,475
Original dataset	train	2,630	1,730,190
	test	1,674	1,228,529
Evaluation set		253,732	185,224,471

selected from the original benchmark dataset [78]. More details about the datasets are as follows:

- *Synthetic dataset*: including 10 MS/MS spectra in the training set and 5 spectra in the test set. This dataset is used for the purpose of finding the most stable classification algorithm across various imbalance ratios. Various training and test sets having different S/N ratios are created from this dataset. Also the dataset is used for tuning the GP fitness function and finding appropriate feature parameters.

- *Original dataset*: including 2,630 MS/MS spectra in the training set along with a test set of 1,674 MS/MS spectra. This dataset is used as the gold standard dataset. The dataset is used with the purpose of comparing the threshold-based preprocessing method with the classification-based method in terms of improvement in the reliability of peptide identification.

- *Evaluation set*: containing 253,732 MS/MS spectra corresponding to doubly charged peptides. This is the large-scale dataset which is used for evaluating the GP method in terms of improving the reliability of peptide identification with *de novo* sequencing and database searching methods.

3.3.2 Benchmark Algorithms

In order to have a comprehensive investigation of the effectiveness of the GP method to handle the imbalanced MS/MS data, different types of classification algorithms including a distanced-based classifier (k-Nearest Neighbour, K-NN), a kernel-based classifier (Support Vector Machine, SVM), a probabilistic classifier (Naïve Bayes, NB), a ruled based classifier (Decision Tree, DT), an ensemble-based classifier (Random Forest, RF), a network-based classifier (Multilayer Perceptron, MLP) are used to compare with GP. The implementation of these learning algorithms are taken from the Waikato Environment for Knowledge Analysis (WEKA) package [193]. A brief description of these algorithms can be seen as follows.

1. K-NN tries to assign a class label resulted from the majority vote of its k nearest neighbours.

2. SVMs attempt to construct hyperplanes in a high dimensional space and classify examples. For each hyperplane, the SVM model aims to maximise the distance between the hyperplane and the nearest data points on each side of it.

3. NB applies the Bayes theorem, which works on conditional probability. NB predicts membership probabilities for each class and then selects the class with the highest probability as the most likely class for an unseen example.

4. DT is a tree-structure classification algorithm, where the paths from root to leaves represent the classification rules. The internal nodes contain splits representing a test on the attributes/features of the samples, the edges correspond to the outcome of the tests and the leaf nodes represent the class labels.

5. RF is an ensemble learning method in which multiple DTs are constructed together at the training time and the output is the most common class among the individual DTs.

6. MLP is a computational model inspired of the human brain and nervous system, aiming at constructing a network using a number of layers that maps the instances to the target class labels.

3.3.3 Experiments

This section conducts three sets of experiments which are addressing the research goals in this chapter.

(1) Experiment Set I (on GenGP method)

In this set of experiments, GP evolves a general GP-based preprocessing model, called GenGP, using four commonly used spectral features in the literature [28]. The goal of designing this GP method is assessing the capability of GP in handling the classification of imbalanced MS/MS data across various imbalanced ratios. The experiments in Experiment set I are briefly as follows:

- *Experiment I*: Investigating important ion types in peptide identification.

- *Experiment II*: Investigating appropriate evaluation metrics for performance evaluation of the classification algorithms using MS/MS spectra in the test set of the gold standard dataset.

- *Experiment III*: Investigating appropriate α coefficient for the proposed GP method on the synthetic dataset.

- *Experiment IV*: Comparing the performance of the proposed GP method with six different classification algorithms across various ratios of S/N using MS/MS spectra from the synthetic dataset.

- Experiment V: Investigating the performance of the proposed GP method on the gold standard dataset.

- *Experiment VI*: Evaluating the effectiveness of the proposed method in terms of improvement in reliability of peptide identification on the gold standard dataset.

In the rest of this section, the experiments above are explained in more details.

Experiment I: Investigating important ion types in peptide identification.

In an MS spectrum, each precursor ion, which indicates the m/z value of a peptide, can be selected and fragmented into hundreds of fragment ions that construct an MS/MS spectrum. During fragmentation by CID, different fragment ion types are generated. In the CID fragmentation technique, we are only interested in b-/y-ions because the amino acid sequence of an MS/MS spectrum can be determined by the mass differences between b-/y-ions. However, during the fragmentation, different ion types such as isotopologues, neutral losses, and doubly charged ions are produced. The presence of different types of ions along with the background noise can produce a large and complex search space for the peptide identification tool to explore, leading to a high false discovery rate. Therefore, prior to peptide identification, it is worth investigating which ion types should be considered as signals and noise peaks. This investigation helps make a clear definition of background noise in the data. Based on the results of this experiment, the peaks in the MS/MS datasets from Table 3.3 (see Page 86) are labelled as either signal or noise to create a gold standard dataset which later will be used by the proposed GP method and other classification algorithms.

Figure 3.3 illustrates the workflow of investigating the important ion types in peptide identification. The workflow starts with an experimental MS/MS spectrum with known peptide. The spectrum is submitted to the Mascot [93], a popular database search engine, for labelling each peak in the spectrum. Different peaks/ions are extracted from the spectrum to create different scenarios from 1 to 7. Table 3.4 shows the Mascot parameter settings. Each sce-

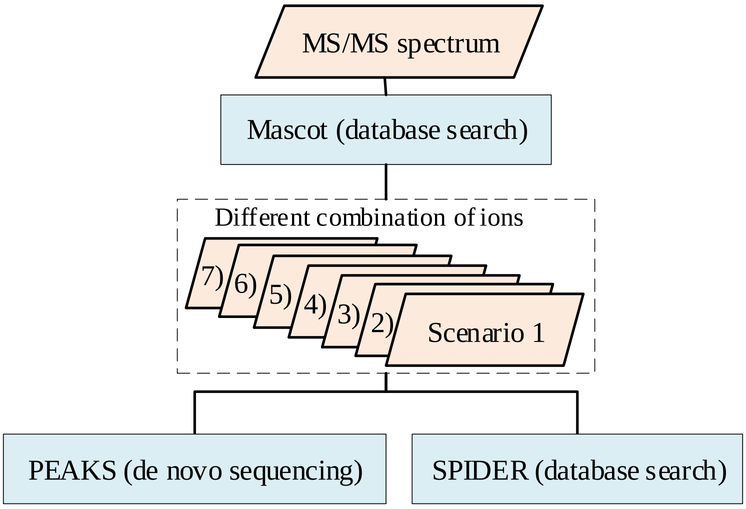


Figure 3.3: The workflow of investigating important ion types for effective peak labelling.

nario containing a spectrum with different ions is submitted to both PEAKS as a *de novo* sequencing software [9], and SPIDER, a benchmark database search tool [194], to re-identify the spectrum.

The single experimental spectrum is chosen from the ground-truth provided by LC-MS/MS benchmark dataset [78]. The spectrum corresponds to doubly charged peptide sequence “SEQGMSLLQPGK”. This spectrum is submitted to Mascot database search tool to be searched against the *Escherichia coli* K12 [195] protein database with Fragment match tolerance of 0.8 Da. Table 3.4 shows more details of the database search parameter setting.

The Mascot search result returns peptide “SEQGMSLLQPGK”, which is the same as the ground-truth. The result of the Mascot database search is exported as an annotated spectrum where each peak is labelled as: y(1+), b(1+), y(2+), b(2+), b(1+)-H2O, b(1+)-NH3, y(1+)-H2O, y(1+)-NH3 and no label which is considered as noise. To find out which ion types should be labelled in the gold standard dataset, different scenarios are provided to test the peptide identification rate using different combinations of the ions. The scenarios are as follows:

Table 3.4: Mascot database search parameter setting

Protein database		**Spectrum**	
database name	Escherichia coli (strain K12)	min. precursor mass	350 Da
enzyme name	Trypsin	max. precursor mass	5000 Da
max. missed cleavage	1	min number of peaks	1
ms1 tolerance	10 ppm	**Fragment match options** (export search results)	
ms2 tolerance	0.8 Da	charge details	+1 and/or +2
Percolator setting		matched tolerance	0.8 Da
validation based on	q-value	selected ion series	b, y, -H2O, -NH3, Immonium
cutoff q-value	0.01		

1. Raw spectrum containing all peaks which include all ion types (as signal peaks) along with noise peaks.
2. Labelling only matched doubly charges *b-/y*-ions as signal peaks. This includes {b(2+), y(2+)}.
3. Labelling only matched singly-charged *b-/y*-ions as signal peaks. This includes {b(1+), y(1+)}.
4. Labelling both matched singly and doubly charges *b-/y*-ions as signal peaks. This includes {b(1+), y(1+), b(2+), y(2+)}.
5. Labelling matched singly-charged and neutral losses as signal peaks. This includes {b(1+), y(1+), b(1+)-H2O, y(1+)-H2O, b(1+)-NH3, y (1+)-NH3}.
6. Labelling CID simulated fragments singly-charged ions as signal peaks.
7. Labelling CID simulated fragments singly-charged ions and neutral losses as signal peaks.

In each scenario, only those peaks which are mentioned in the description of each scenario are submitted to PEAKS and SPIDER, while other peaks are removed from the spectrum. The results of these experiments will indicate which scenario results in a higher score (more confident) identified peptide with a larger number of matched amino acids. Two different commonly used peptide identification tools, PEAKS and SPIDER, are used to make a stronger conclusion. Based on the results of this experiment, the peaks in the spectra from the datasets in Table 3.3 are labelled to create a gold standard dataset.

Experiment II: Investigating appropriate evaluation metrics for performance evaluation of classification algorithms.

This experiment investigates three metrics *SE*, *SP*, *A-acc*, which are introduced in Equation (3.5), to determine the efficient metrics in noise reduction and signal retention of MS/MS spectra when a preprocessing method is applied on the data. *SE* is calculated as the number of correctly classified positive instances of peaks divided by the total number of positives, whereas *SP* is the number of correctly classified negative instances divided by the total number of negatives. The selected metrics later are used to evaluate the performance of the proposed GP method and other classification algorithms.

Moreover, since the proposed GP preprocessing method is compared with the current threshold-based method [92, 8] in terms of improving peptide identification on the test set of the gold standard dataset, by conducting this experiment, the best threshold value for the dataset used in this study is achieved to have a fair comparison. The idea is finding the best threshold value of the gold standard dataset in terms of the number of identified peptides. The test set contains 1,674 MS/MS spectra that correspond to 1,674 peptides (ground-truth).

To find the best threshold value, different thresholds ranging from 0 to 25,000 are considered. Then based on the flowchart in Figure 3.1, the preprocessed data by each threshold value is submitted to PEAKS to identify the peptides. The threshold value with the highest number of identified peptides is chosen as the best threshold value for the dataset.

After finding the best threshold value, *SE*, *SP* and *A-acc* are calculated for all thresholds. The aim is to find a relationship between the three aforementioned metrics across different thresholds. This relationship (pattern) can give us a rough idea of how to evaluate the proposed GP-based preprocessing method and other classification algorithms in Experiment IV and Experiment V. The results are explained in Section 3.4.

Experiment III: Investigating appropriate α coefficient for the proposed GP method.

In binary classification of imbalanced datasets, it is highly important to identify instances belonging to the minority class correctly. Therefore, as previously shown in Equation (3.5) on Page 85), a weighted average is used to evaluate the evolved GP classifiers. In this section, different α coefficients are experimentally checked to find a suitable α value. The synthetic dataset containing 10 spectra in the training set and 5 spectra in the test set from Table 3.3 is used to conduct this experiment.

Experiment IV: Comparing the performance of the GP method with six different classification algorithms across various ratios of S/N peaks in the synthetic dataset.

Since MS/MS spectra are highly imbalanced and due to the fact that the ratio of imbalance varies between datasets, the aim of this experiment is to investigate the stability of GP along with 6 learning algorithms on the synthetic dataset across different imbalance ratios. Various training and test sets having different S/N ratios including {1:1, 1:2, 1:4, 1:6, 1:13, 1:20} from the synthetic dataset, where 1:1 indicates a balanced dataset whereas 1:20 implies 20 times more noise peaks than signal peaks are created. The performance of each classifier is measured through the suitable metrics identified in Experiment II.

Experiment V: Investigating the performance of the proposed GP method on the gold standard dataset.

In this experiment, GP and other six classifiers are allowed to build the model on the training set and then test it on the test set of the gold standard dataset. The aim of this experiment is to investigate if the results on the gold standard dataset is consistent with those of the synthetic dataset (in Experiment IV).

Experiment VI: Evaluating the effectiveness of the proposed method in terms of improvement in reliability of peptide identification.

In this experiment, the impact of the GP method on peptide identification on the test set of the gold standard dataset is investigated. Three different scenarios are provided to investigate the effectiveness of the preprocessing method. The three scenarios are as follow:

1. Using the original data (un-preprocessed dataset).
2. Preprocessing the data using the best threshold value identified in experiment II.
3. Preprocessing the data produced by GenGP.

The spectra in the test set of the gold standard dataset preprocessed by each scenario are submitted to the PEAKS software to identify the peptides and the peptide reliability of each scenario is measured through the 5 different ALC ranges.

(2) Experiment Set II

The second set of the experiments investigates whether the performance of the proposed GP method improves by incorporating additional fragmentation rules from CID spectra (considering all feature groups from Table 3.1 on Page 79). The spectral features might provide more evidence to the classifier for distinguishing the noise peaks from signal peaks. Therefore, a set of experiments including tuning “Top X in Win ± Z” features using different X and Z parameters, and investigating the effectiveness of each group of features in improving the classification performance are conducted. As adding more features increase the search space of the classification problem, the experiments also investigate the feature selection capability of GP. So the feature selection capability of GP is explored by combining all features from Table 3.1. The method is called CID-GP. The implicit feature selection and interpretability of GP might reveal important spectral features that have positive influence in classification of peaks.

(3) Experiment Set III (Performance Evaluation on Peptide Identification Reliability)

Finally, the performance of GenGP and CID-GP are compared with the best intensity-based thresholding method in terms of improving the reliability of peptide identification with PEAKS as a *de novo* sequencing method and SEQUEST [92] as a benchmark database search engine. The evaluation set from Table 3.3 is used to run this experiment.

3.4 Results and Discussions

This section presents the results of all experiments in the previous section and discusses important findings.

3.4.1 Results of Experiment Set I

Results of Experiment I, important ion types

This section presents the results of Experiment I where important ion types in peptide identification for effective peak labelling is investigated. As previously mentioned, seven different scenarios (see Page 91) each using different combinations of fragment ions are considered. Table 3.5 presents the results of peptide identification on each scenario using PEAKS *de novo*

Table 3.5: Comparisons of various scenarios containing different set of ion types submitted to PEAKS and Spider for peptide identification. The ground-truth sequence is "SEQGMSLLQPGK". The scenario with high scores and more sequence coverage for both identification tool indicates containing the most important ion types. Scenario 6 gives the best results.

	PEAKS		SPIDER	
Scenario no.	Identified Sequence	score (%)	Identified Sequence	score (%)
(1)	WNKVELASAE**K**	34	DLGFLPGDLAE**K**	17.42
(2)	FLLLKEYGY**K**	10	FLLLDEPTRGL	19.34
(3)	LCK**GMSLLQPGK**	61	**SEQGMSLLQPGK**	42.02
(4)	CLK**GMSLLQPGK**	60	**SEQGMSLLQPGK**	42.02
(5)	YHQLLSMT**PGK**	52	LTTLLLSQGTPM	21.63
(6)	LCK**GMSLLQPGK**	70	**SEQGMSLLQPGK**	42.02
(7)	CLLV**LL**SMT**PGK**	72	GQDQLLSLAGGDT	25.02

sequencing and SPIDER database search. As previously mentioned, a single experimental spectrum is chosen from the ground-truth and its corresponding doubly charged peptide sequence is "SEQGMSLLQPGK".

Any letter of the identified sequence which is in bold indicates an exact match against the corresponding letter of the ground-truth sequence. In this table, the scenario that obtains a high PEAKS and SPIDER confidence score with higher sequence coverage (a larger number of bold AA letters) to the ground-truth is selected as the best case and determines the most important ion types to peptide identification. This indicates the best decision on the ion types to be selected for labelling the MS/MS datasets which are used by the machine learning methods. The following sections analyse the results of each tool separately.

The Scenarios Submitted to PEAKS and the Peptide Identification Results Table 3.5 presents the results of each scenario submitted to PEAKS for *de novo* sequencing of the spectrum. The results in Table 3.5

show that submitting the raw spectrum (scenario 1), which contains all ion types and noise peaks, results in a sequence with a low ALC score of 34. Moreover, if only matched doubly charged ions (scenario 2) are selected out of all peaks in the spectrum and submitted to the *de novo* sequencing tool, a worse result, ALC = 10, is obtained. However, the matched singly charged of b-/y-ions (scenario 3) gives a better result of ALC = 61. In the next scenario, both singly and doubly charged ions are combined (scenario 4), but the ALC score decreases to 60. The next scenario combines singly-charged and neutral losses ions (scenario 5) to investigate whether the presence of neutral ions can improve the identification rate. However, the results deteriorate (ALC = 52) due to the fact that the presence of neutral losses ions and doubly charged ions makes the ladder complicated and increases the potential false positive sequences. Therefore, so far, only using the matched singly-charged ions (scenario 3) are the best choice (ALC = 61). However, only using the matched ions may make the ladder incomplete and deteriorate the performance of peptide identification. In the next scenario, the ions are constructed virtually based on the known CID fragmentation rules of doubly charged peptides [190] using only b-/y-(1+) (scenario 6). This results in a complete ladder and presents a higher ALC score and closer to the exact match compared to the previous scenarios. The last experiment combines CID b-/y-(1+) and neutral losses (scenario 7). Although the ALC score is the highest among all experiments, the sequence is far from the exact match. Therefore, it can be seen that scenario 6 where only CID ions are used, gives the best results. Furthermore, another set of experiments using the database search tool, SPIDER, is conducted in the following to make a stronger conclusion.

The Scenarios Submitted to SPIDER and the Peptide Identification Results Here, the same set of scenarios are submitted to SPIDER. The purpose of running this experiment is to see how a database search tool interact with different sets of ions/peaks in the spectrum and to check

Table 3.6: Labelled Datasets

Datasets		No. of spectra	No. of signal peaks	No. of noise peaks
Synthetic dataset	train	10	278	9,680
	test	5	115	4,360
Golden standard dataset	train	2,630	42,960	1,687,230
	test	1,674	38,707	1,189,822
Evaluation set		253,732	4,095,873	181,128,598

whether the previous results from the de novo sequencing is consistent with the results from a database search tool.

From Table 3.5, it can be seen that similar to the PEAKS results, using doubly charged ions and neutral losses ions (scenario 2) does not help the peptide identification. Since there is a protein database to be searched against, it can be seen that in three scenarios, 3, 4, and 6, the results are the same as with each other. It can be seen that, both tools on scenario 6 show good results. Therefore, only extracting CID singly-charged b-/y-ions can “guarantee” to obtain a reasonable peptide identification rate. Shao et al. [29] has also reported that complementary signal peaks are more likely to be found at a charge state of +1 than at other charge states.

So based on the results in this section, all datasets introduced in Table 3.3 are labelled by considering only CID singly charges ions as signal peaks and the rest of the peaks as noise peaks. Table 3.6 presents more details about the number of signal and noise peaks in all datasets. It can be seen that all datasets are highly imbalanced.

Results of Experiment II, appropriate evaluation metrics

This section provides the results of Experiment II where appropriate evaluation metrics for performance evaluation of classification algorithms is investigated. Figure 3.4 presents the result of peptide identification done by

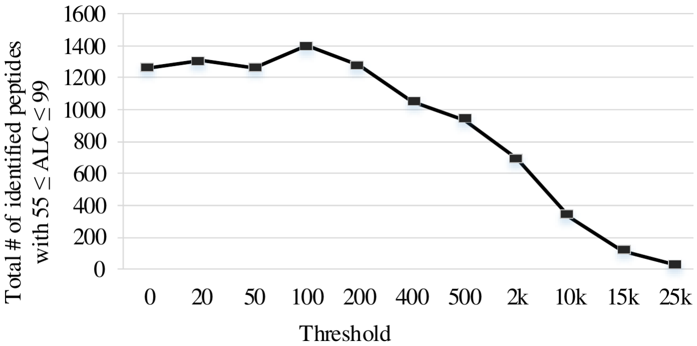


Figure 3.4: The result of peptide identification using PEAKS on the test set of the gold standard dataset containing 1,674 MS/MS spectra preprocessed by different intensity-based threshold values.

PEAKS on the test set of the gold standard dataset containing 1,674 MS/MS spectra preprocessed by 50 experimentally determined threshold values in the range $0 - 25,000$. The results represent the number of identified peptides with ALC scores between 55 and 99. For more convenient presentation of the overall trend, the x-axis of the plot shows only a few threshold values out of 50. It can be seen that by increasing the threshold from 0, identification rate slightly increased and reached a peak of about 1400 by using threshold 100. After this, there was a sharp decline in the number of identified peptides, decreasing to less than 50 using threshold 25k. Therefore, the best threshold value for the gold standard dataset is at 100.

In order to further investigate how different threshold values including the best threshold classify signal and noise peaks, for each threshold, *SE*, *SP* and *A-acc* have been calculated and plotted in Figure 3.5. Again similar to Figure 3.4, only the results of the a few number of threshold values in the x-axis have been plotted. Overall, it can be seen that *SP* is far higher than *SE* throughout the whole threshold increment. This indicates that selecting high threshold values to filter the signal and noise peaks in MS/MS spectra results in removing a significant number of signal peaks due to their low intensity values which are below the thresholds. Therefore, a trade-off between *SE* and *SP* is required. This indicates that classification of noise and signal peaks in

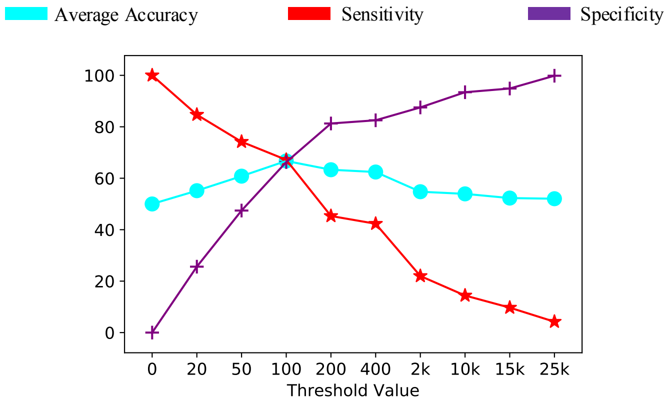


Figure 3.5: The total number of identified peptides by PEAKS with confidence scores between 55 and 99 on the original data and the preprocessed data by GP method and an intensity-based thresholding method. The y-axis shows the value of each measure for its corresponding graph.

MS/MS data is a multi-objective problem as increasing the accuracy of one class results in decreasing the accuracy of another class. The multi-objective approach will be investigated in the next chapter.

The *A-acc* graph shows a steady downward trend in overall, which seemed set to continue to reduce. As the best threshold value of 100 got the highest rate of *A-acc* among other thresholds, it seems that *A-acc*, a trade-off value between *SE* and *SP*, is an appropriate metric to evaluate the classification algorithms in the next experiments.

In summary, the results show that to have high peptide identification reliability, the preprocessing method is required to have a high *A-acc* value. Since identifying signal peaks can highly improve the peptide identification rate, as the second criterion to evaluate the classification performance, *SE* value is considered as well. Therefore, in the next two experiments, the effectiveness of the preprocessing methods in terms of two metrics, *A-acc* and *SE* is measured.

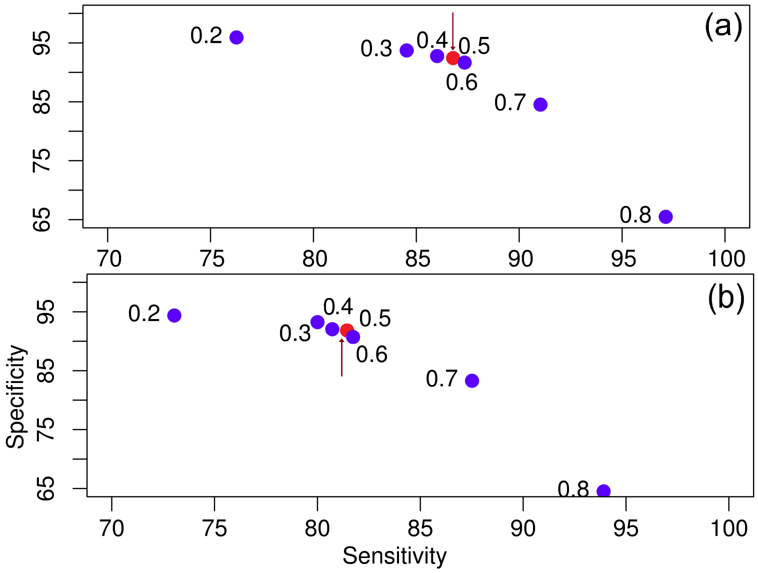


Figure 3.6: The classification results of GP on the labelled synthetic dataset for different coefficients of sensitivity and specificity in the GP fitness function: $\alpha \times SE + (1-\alpha) \times SP$, where $\alpha = \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. (a) The classification results on the training set. (b) The classification results on the test set.

Results of Experiment III, investigating appropriate α coefficient for GP

This section presents the results of Experiment III which investigates a suitable α value used in the fitness function (Equation (3.5) on Page 85) of the proposed GP method.

Figure 3.6 shows the classification results of GP using different coefficients for *SE* and *SP* on training and test sets of the labelled synthetic dataset from Table 3.6, respectively. On one hand, it can be seen that by increasing the coefficient of *SE*, the specificity in both the training and test sets drops.

On the other hand, giving a high coefficient to *SP* can decrease the sensitivity and this is not desired. Therefore, it seems that the sets of coefficients ($0.5 \times SE + 0.5 \times SP$) or ($0.6 \times SE + 0.4 \times SP$) work better compared to other sets. However, ($0.6 \times SE + 0.4 \times SP$) results in higher sensitivity value, but decreases the precision due to the increase in false positives.

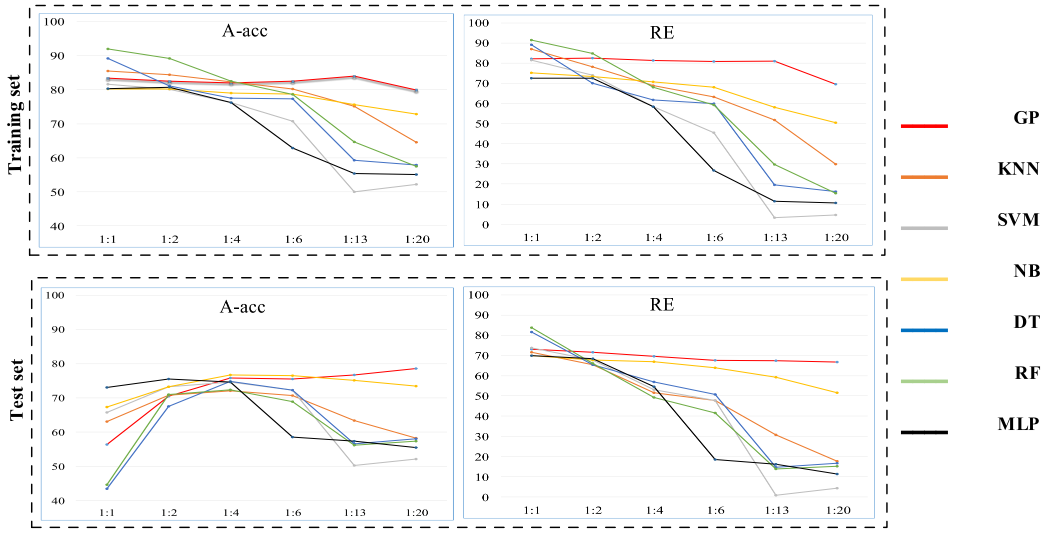


Figure 3.7: Classification results of the GP method and six classification algorithms on the synthetic dataset containing 10 MS/MS spectra in the training set and 5 MS/MS spectra in the test set on different S/N ratios.

Therefore, the rest of the experiments are done by using $\alpha = 0.5$, i.e., $(0.5 \times SE + 0.5 \times SP)$ as the fitness function of GP.

Results of Experiment IV, investigating the stability of GP across various ratios of S/N.

This experiment provides the results of comparing the performance of the proposed GP method with six different classification algorithms across various S/N ratios of the synthetic dataset. For the GP system, the experiments are repeated for 30 independent runs with 30 different random seeds.

Figure 3.7 shows the classification results of the GP method and other classification algorithms on the synthetic dataset. It can be seen that GP has the best classification results in both training and test sets in terms of *A-acc* and *SE* where noise ratio is high. All classifiers show decreasing *A-acc* with the decrease in the S/N ratio of the MS/MS data in the training set except GP that shows relatively “constant” accuracy at >80%. On the test set most of the classifiers show top accuracy at 1:4 S/N ratio that dropped

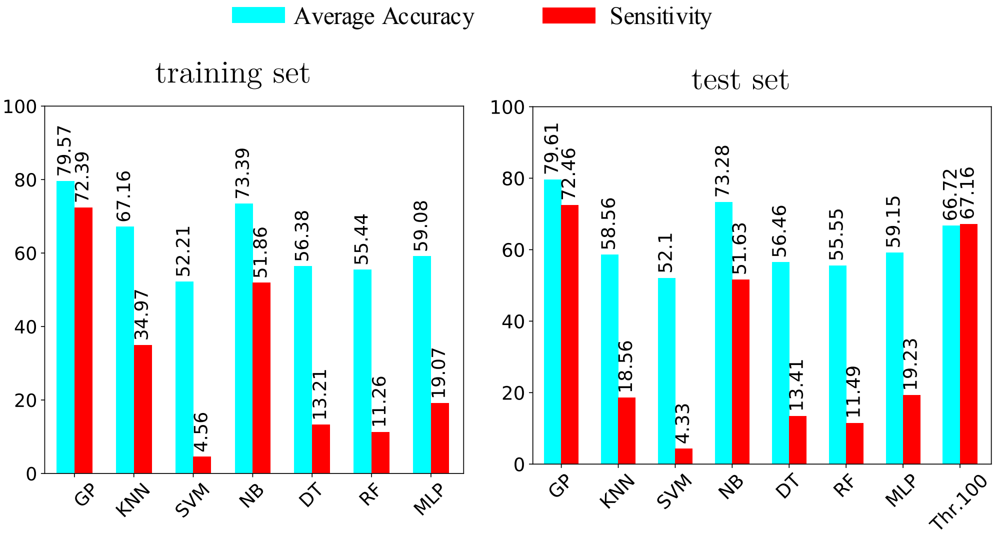


Figure 3.8: Classification results of the GP method and six classification algorithms on the gold standard dataset containing 2,630 MS/MS spectra in the training set and 1,674 MS/MS spectra in the test set. The classification results of the best intensity-based thresholding method (Thr.100) is also presented on the test set for further comparison.

significantly with the increase of the noise ratio except GP that shows steady increase in accuracy with the increase of the S/N ratio.

Overall, GP showed the best performance in terms of *A-acc* among other classification algorithms. In terms of *SE*, GP also showed the best and most stable behaviour in both training and test sets. The GP algorithm developed here focuses on both the majority and the minority classes in finding signal peaks (*A-acc* and *SE*), which is desirable. Two possible reasons for the good performance of GP could be: (1) other methods did not use the average accuracy as the cost function in the training process, and (2) GP is capable of automatically applying features selection while solving a binary classification problem.

Results of Experiment V, investigating the performance of the proposed GP method on the gold standard dataset.

Figure 3.8 presents the results of *A-acc* and *SE* that were achieved by all classifiers on both training set and test set of the gold standard dataset. As can be seen in Figure 3.8, the results of all classification algorithms on the gold standard dataset are consistent with those in Figure 3.7 at 1:20 S/N ratio. The S/N ratio in both training set and test set of the gold standard dataset is almost 1:30. This experiment demonstrates the performance of GP when facing an MS/MS spectra dataset with a realistic imbalance ratio. The results also show that GP has the best classification results in both training and test sets in terms of *A-acc* and *SE*. The results of the best threshold value, Thr.100, which was previously presented in Figure 3.5 has been illustrated in bar chart of test set, to make a comparison between the accuracy of the best classification method and best threshold based method. It can be seen that, the results of the GP method in terms of *A-acc* and *SE* is better than those of the best threshold value. It is worth mentioning that in threshold based method, using only one feature value, intensity, could not achieve a trade-off between classification of signal and noise peaks. Whereas with the proposed GP method a high value of *A-acc* and *SE* can be achieved, and this leads to discriminating more signal peaks from noise ones.

In summary, the GP algorithm has been demonstrated to be promising in handling the highly imbalanced MS/MS data in retaining the signal peaks that are minority instances and removing the noise peaks that are the majority ones. GP achieved the best average accuracy and sensitivity results compared to the other algorithms examined. Next, the preprocessed MS/MS data by GP is submitted to PEAKS in order to identify the peptides and evaluating the effectiveness of the prepossessing GP method.

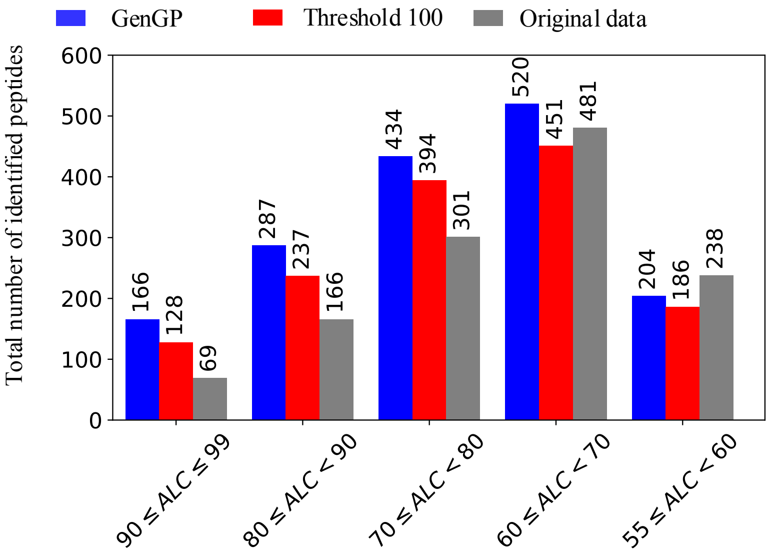


Figure 3.9: The results of peptide identification using PEAKS on the test set of the gold standard dataset. The original un-preprocessed data and the preprocessed data by the initial proposed GP method and an intensity-based thresholding method are given to PEAKS and the results are shown in different ALC ranges.

Results of Experiment VI, evaluating the effectiveness of GenGP in terms of improvement in reliability of peptide identification.

Based on the results of Experiment II, threshold 100 received the highest peptide identification rate among the other thresholds. The results of the experiments IV and V showed that the GP method is the most promising preprocessing method to classify signal and noise peaks. Therefore, the results of GP with the best threshold value is compared here. Figure 3.9 shows the result of peptide identification performed by PEAKS using preprocessed and un-preprocessed data referring to the three scenarios mentioned before (see Page 94). The experiments are run on 1,674 MS/MS spectra in the test set of the gold standard dataset. In Figure 3.9, the original data refers to the un-preprocessed spectra. The results are presented in five different ranges of ALC scores. For each ALC range, the number of identified peptides by PEAKS has been counted.

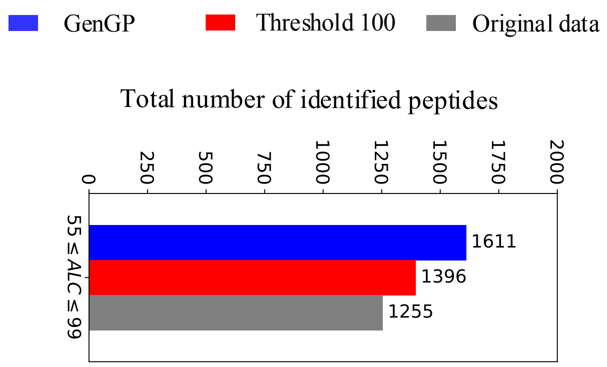


Figure 3.10: The total number of identified peptides by PEAKS with confidence scores between 55 and 99 using original data and the preprocessed data by the initial proposed GP method and an intensity-based thresholding method.

Overall, GP could identify more high ALC scored peptides compared to the other scenarios. It can be seen that there is a significant difference between the results of GP for peptides with ALC higher than 90% compared to un-preprocessed data. For $60 \leq ALC \leq 99$ in every range of ten, preprocessing data with GP achieved the highest number of identified peptides by PEAKS. However, the identification rate slightly dropped in the range of $55 \leq ALC < 60$ for both GP method and threshold 100 compared to the un-preprocessed data as most of the peptide are already identified with higher confidence score and are shifted to the other ranges.

Figure 3.10 presents the results of total number of identified peptides by PEAKS with ALC scores between 55 and 99. These results are the summation of number of identified peptides for each ALC range presented in Figure 3.9. For $55 \leq ALC \leq 99$, the results shows that GP could help PEAKS in finding more high confidence peptides rather than the other methods. GP could improve the reliability of peptide identification by 28.3% $(= (\frac{1611-1255}{1255} \times 100))$ compared to un-preprocessed data, whereas the best threshold at 100, had the improvident of 11.2% $(= (\frac{1396-1255}{1255} \times 100))$. Comparing both GP method and threshold method, GP had 15.4% $(= (\frac{1611-1396}{1396} \times 100))$ improvement over threshold method.

In summary, the influence of the proposed GP method on PEAKS in terms of finding more high confident identified peptides is more than the threshold-based preprocessing method. The reason is that the threshold method ignores those peaks with intensities less than the threshold, resulting in loosing many low intensity signal peaks and keeping a number of high intensity noise peaks. That is one of the disadvantage of threshold method as it ignores the hidden relationship between the peaks and only filters them based on only the intensity feature. Figure 3.8 shows that GP achieved 72.46% of *A-acc* and 86.77% of *SE* on the test set of the gold standard dataset, which means GP could keep reasonable amount of signal peaks, while removing a significant number of noise peaks and this allows GP to improve the results of PEAKS.

3.4.2 Results of Experiment Set II

This section presents the results of the experiments conducted to tune the parameters and features. The classification results of each group of features along with combining all features are also provided.

Tuning the Parameters in "X Intensities in Win ± Z" Feature In this section, a set of experiments is conducted where GP is used to perform binary classification on labelled synthetic dataset using one single feature of "Top X Intensities in Win ± Z". A range of 1-10 is considered for X, while Z ranges from 27 to 100 with an increment of 10. The value 27 is suggested by [196] where a top 1 in Win 27 approach is applied on the MS/MS spectrum to remove the potential noise peaks. Here, the aim of running these experiments is finding appropriate parameter values for X and Z that keep the classification performance reasonably high. The average accuracy graphs in Figure 3.11 show that for X values from 3 to 8, and for window size Z values less than 60 (the first four graphs starting from top left), the results of classification in terms of average accuracy is reasonably high (more than 85%). By increasing the window size Z to more than 70, the average accuracy graphs keep increasing for all X values in range the 1-10. It

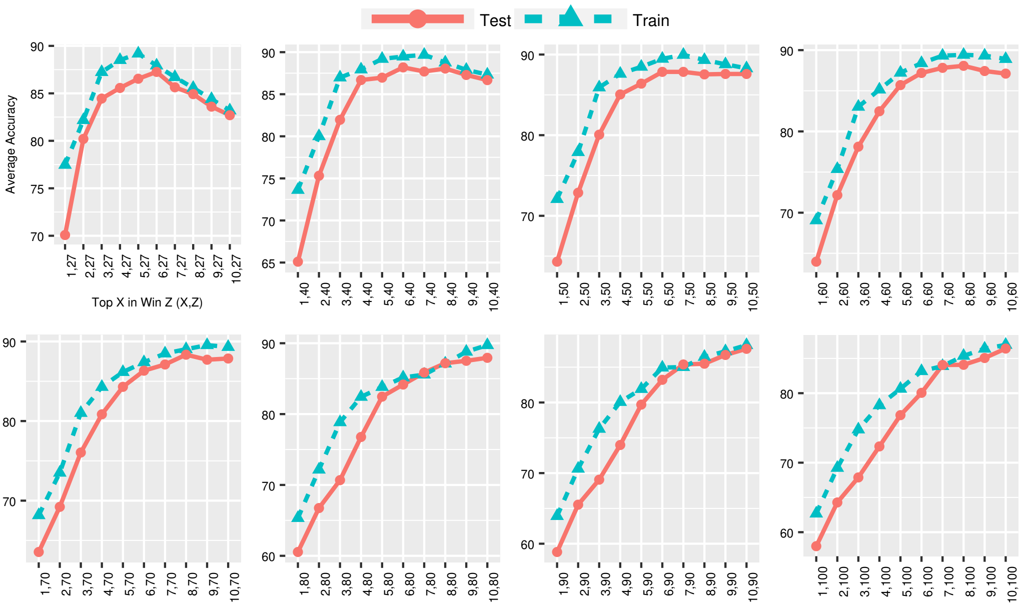


Figure 3.11: Classification results of the GP method on the synthetic dataset using one single feature of “Top X Intensities in Win ± Z”.

means that for window sizes more than 70, the X range of 1-10 is not sufficient to give a downward trend to the average accuracy graphs. The reason is that for a big window size Z, increasing the value of X gives more chances to keep the potential signal peaks. Because of retention of more signal peaks, the classification accuracy increases. As Z indicates a neighbourhood around each peak, we are not interested in big neighbourhoods which can turn the local “top X intensities in Win ± Z” feature to a global feature. Therefore, one solution could be limiting the range of Z to the mass of the smallest amino acid which is 57 Da. This is almost consistent with [27] where only 13 sets of X,Z where X,Z = (1,27), (3,56), (4,40), (4,50), (4,60), (6,25), (6,30), (6,40), (6,50), (6,60), (8,40), (8,50), (8,60) were used for the purpose of noise thresholding of MS/MS spectra. Also, another alternative would be considering all X and Z values in the graphs of Figure 3.12 where the average accuracy is more than 85% in both train and test sets. As there are 45 cases with that condition, the next experiments investigate the classification results of using 45 sets of X,Z as features. Moreover, as the 13 sets of X,Z

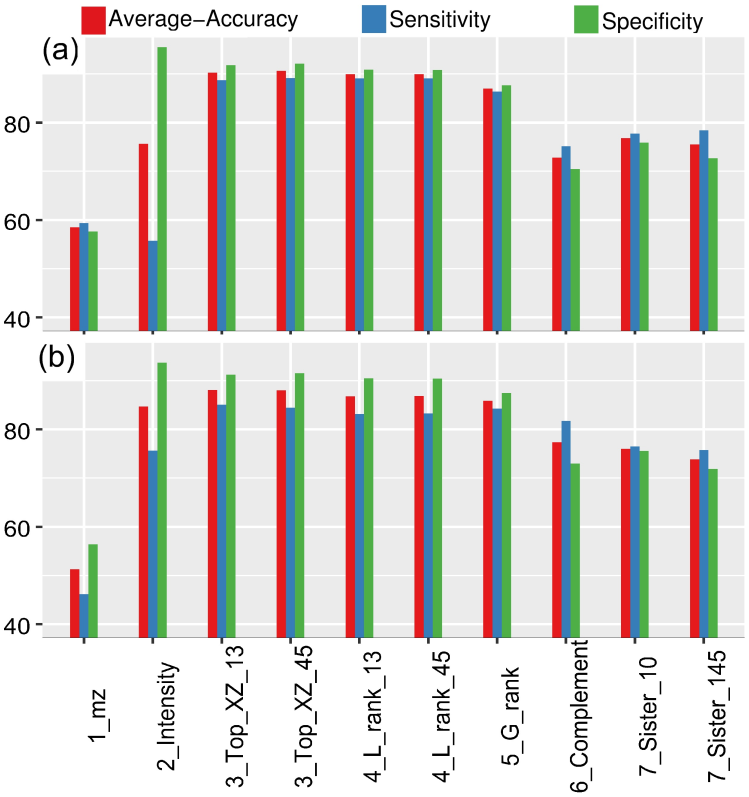


Figure 3.12: The classification results of GP on labelled synthetic dataset. (a) Using each group of features individually on the training set. (b) The results on the test set.

reported in [27] were not used as the features for the machine learning-based preprocessing method, in the next experiment, these sets of features are used with the GP method and the results will be compared with other groups of features including the top 45 sets of X,Z values.

Classification Results of each Group of Features This experiment investigates the effectiveness of each group of features in improving the classification performance of the GP system. For "Top X Intensities in Win ± Z" and "Sister Ions" features, various parameter values are considered to investigate all possibilities. Figure 3.12 (a) and (b) show the classification results of each group of features from Table 3.1 (on Page 79) on the labelled synthetic dataset.

The results of Figure 3.12 (a) and (b) show that among the seven groups of features from Table 3.1, “Top X Intensities in Win ± Z” and “Local Rank in Win ± Z” groups (group 3 and 4) achieve the highest classification results on both the training and test sets compared to other feature groups. The reason is that these two groups try to identify possible noise peaks within a local window around the current peak and keep signal peaks. The second best group of features is “Global Rank” (group 5), where each peak is compared to all of the peaks in the spectrum.

“Complementary Ion” and “Sister Ions” groups are the third best sets of features. These features are based on the CID fragmentation rules and try to find the hidden relationship between the peaks in the whole spectrum without considering the intensity of each peak. The last two best features are “Normalised m/z” and “Normalised Intensity” groups.

As mentioned before, for the two groups of features, “Top X Intensities in Win ± Z” and “Sister Ions”, two sets of experiments including different parameter values are conducted to investigate appropriate parameters for these groups. For the group “Top X Intensities in Win ± Z”, two sets of 13 and 45 features are used. The bar charts (Figure 3.12 (a) and (b)) show that the classification results in terms of average accuracy, sensitivity, and specificity on both the training and test sets are relatively the same for both sets of 13 and 45 features. As using more features during the learning process requires more processing time, for the group “Top X Intensities in Win ± Z”, 13 features are considered afterwards. Also, the results of the bar charts show that for “Sister Ions” group, considering only 10 common features will be sufficient to get a reasonable classification result on both train and test sets compared to having all 145 possible features.

So far, the results of each individual groups of features are obtained. Now it is worth investigating the classification results of combining all features together.

Classification Results of Combining All Features GenGP, explained in Experiment Set I , only considers 4 features. In this experiment, a different set of features are extracted from the MS/MS data. Based on the results of the classification of individual groups, a total number of 40 features including 1, 1, 13, 13, 1, 1, and 10 features from group (1) to (7) are extracted, respectively. For more investigation, 145 possible sister ions from group (7) are used together with another 30 features from group (1) to (6). Therefore, a total number of 175 features will be compared to 4 and 40 features. Shao et al. [29] show that a total number of 20 delta values including common neutral losses with Δ = 17, 18, 28, 34, 35, 36, 44, 45, 46, 64 and isotopic ions with Δ = -1, -2, +1, +2 and delta values separated by masses of amino acids including Δ = 57, 63, 71, 87, 97, 99 are meaningful delta values and contribute to better signal and noise peak discrimination. Therefore, the 20 features above from group (7) along with 30 features from other groups (totally 50 features) will be also compared to 4, 40, and 175 features to find out the best set of features aiming at increasing the classification performance on the gold standard dataset (Table 3.6, which is a large dataset containing thousands of spectra.

Figure 3.13 shows the classification results of GP using different number of features on the training and the test sets of the gold standard dataset. It can be seen that there is a huge difference between using only 4 features and using more than 4 on both train and test sets. This is a good indication to motivate using more features. Among the other cases when using 40, 50, and 175 features, it can be seen that using 40 features gives higher classification result on the training and test sets compared to using 50 and 175 features. Also, training process takes shorter time when using 40 features compared to 50 and 175 features. In summary, the results show that choosing 40 spectral features are good discriminators to help GP identify signal and noise peaks. As the main purpose of having a preprocessing method is improving the peptide identification reliability, the next section evaluates the effectiveness of the new GP-based preprocessing method using 40 features (CID-GP) and

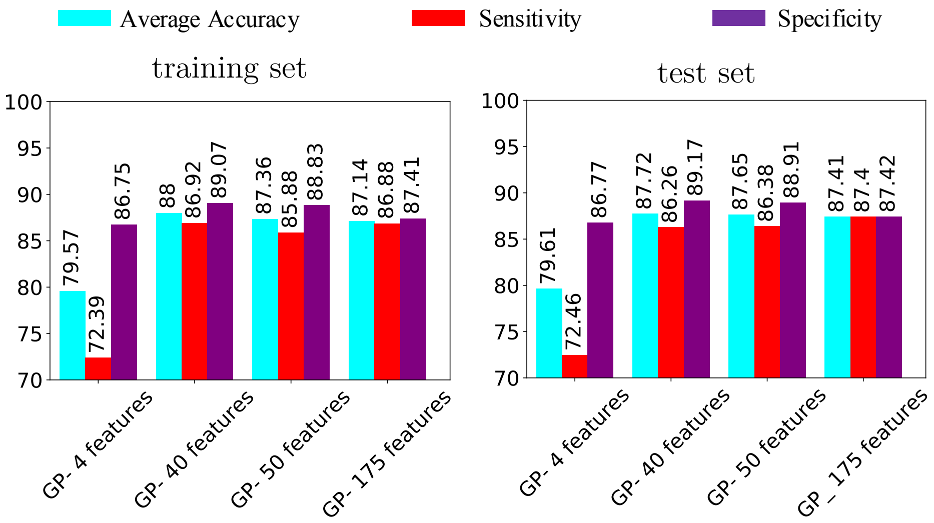


Figure 3.13: The classification results of GP on the gold standard dataset using different set of features.

compares the results with GenGP using 4 features, the best threshold-based method for the gold standard dataset, and the un-preprocessed data.

3.4.3 Results of Experiment Set III

Evaluating the Effectiveness of CID-GP on a large-scale dataset using PEAKS In this section a large-scale peptide identification using the evaluation set which contains 253,732 MS/MS spectra is performed. The evaluation set preprocessed by CID-GP, GenGP and the intensity-based thresholding method are submitted to PEAKS for peptide identification. Also, the evaluation set without applying any preprocessing method is submitted to PEAKS to be a baseline of all comparisons.

Figure 3.14 shows the results of peptide identification done by PEAKS using different methods to preprocess the evaluation set. Overall, CID-GP achieved the highest number of identified peptides by PEAKS compared to the other methods. The CID-GP helped PEAKS identify more highly confident peptides with scores $70 \leq ALC \leq 99$. Since the method has

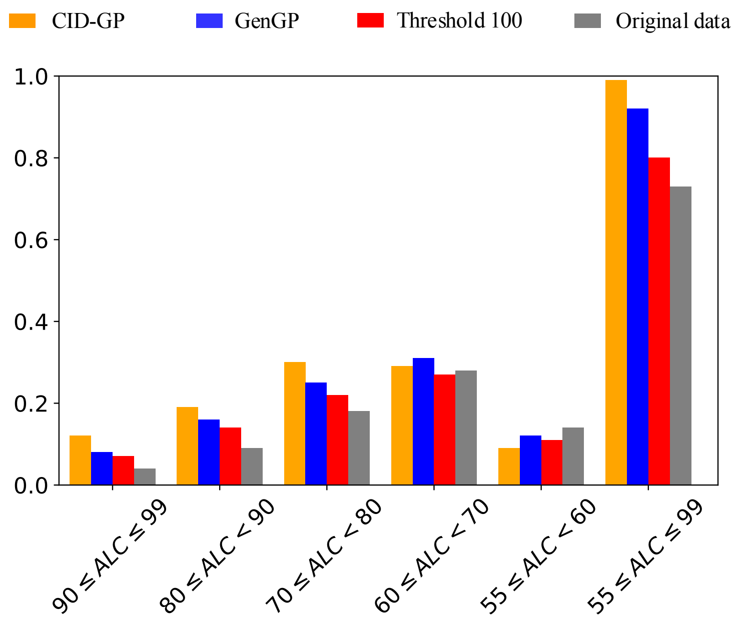


Figure 3.14: The results of peptide identification using PEAKS on the evaluation set. The original un-preprocessed data and the preprocessed data by CID-GP, GenGP and the intensity-based thresholding method are given to PEAKS and the results are shown in different ALC ranges including the whole range of $55 \leq ALC \leq 99$.

already identified a large number of peptides in range $70 \leq ALC \leq 99$, there are fewer peptides to be identified with low confidence scores in range 55 $\leq ALC < 70$.

For $55 \leq ALC \leq 99$ in Figure 3.14, which contains the results of the summation of identification rate for each ALC range, the results show that CID-GP could help PEAKS find more highly confident peptides rather than the other methods. This method could improve the reliability of peptide identification by 26.6% (= (0.995−0.728)×100) compared to un-preprocessed data. Comparing CID-GP with the threshold method, CID-GP has 19.3% (= (0.995 − 0.802) × 100) improvement over the threshold method. Also, CID-GP has 7.2% (= (0.995 − 0.923) × 100) improvement compared to the GenGP.

In terms of the identification rate, the CID-GP could help the peptide

Table 3.7: The statistical results of peptide identification by SEQUEST using CID-GP, GenGP, an intensity-based thresholding method, and gold standard data (un-preprocessed/original data). The average and standard deviation of xcorr-scores for each method is shown.

	CID-GP	GenGP	Threshold 100	Original data
Xcorr-score	**3.07 ± 0.74**	2.93 ± 0.78	2.91 ± 0.77	2.54 ± 0.63

identification tool identify 99.5% of the highly confident peptides, whereas the threshold method only achieved an identification rate of 80.2%.

In summary, CID-GP helps PEAKS find more highly confident identified peptides than threshold-based preprocessing method. The reason is that the threshold method ignores those peaks with intensities less than the threshold, resulting in loosing many low intensity signal peaks and keeping a number of high intensity noise peaks. That is one of the disadvantage of threshold method as it ignores the hidden relationship between the peaks and only filters them based on only the intensity feature. CID-GP achieved *A-acc* of 88% and *SE* of 86.92% on the test set of the gold standard dataset (see the 40 features bar charts in Figure 3.13. This means that GP could keep a reasonable amount of signal peaks, while removing a significant number of noise peaks and this allows GP to improve the results of PEAKS.

Evaluating the Effectiveness of CID-GP Using a Database Search Tool for Peptide Identification

The same experiment explained in previous section is conducted using a database search engine, SEQUEST [92], to check the effectiveness of the GP-based preprocessing method. SEQUEST is a dominant benchmark database search tool and reports a confidence score for each peptide spectrum match. A cross-correlation (Xcorr) as a confidence score measures the goodness of fit of experimental spectra to theoretical spectra created from the sequence b- and y-ions. For each spectrum, the peptide candidate with the highest Xcorr-score is known to be a better match.

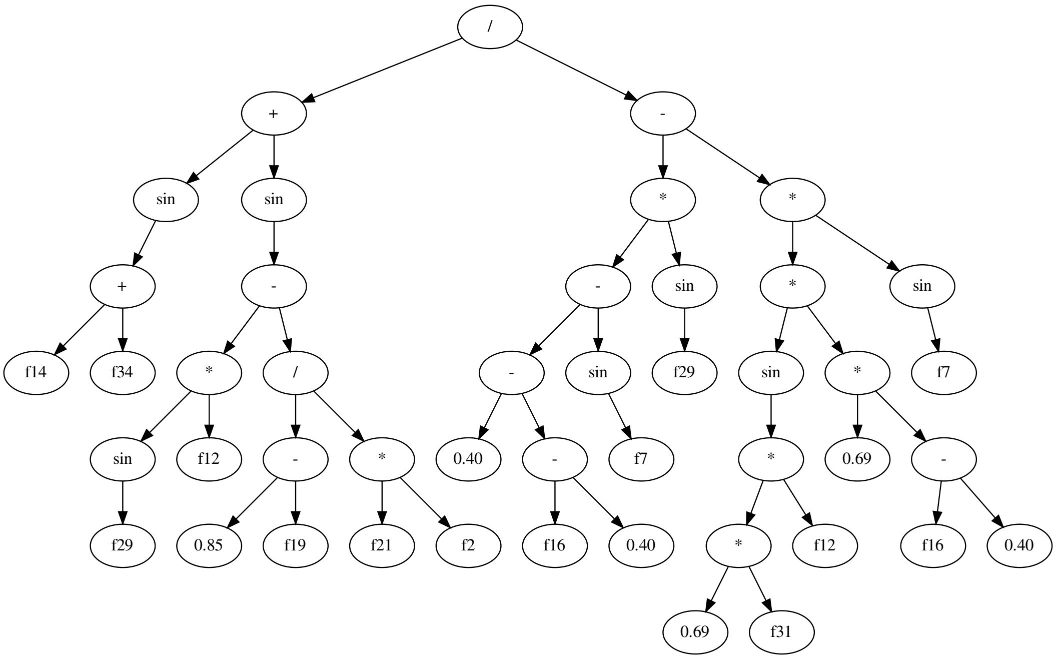


Figure 3.15: The best GP evolved program using 40 features.

To compare the results of CID-GP with other methods, a statistical unpaired t-test with 95% confidence interval is used. Table 3.7 shows that the result of CID-GP is statistically significantly better (shown in italics in the table) than the other methods. CID-GP outperformed GenGP and increased the mean of the Xcorr score by 0.53 and 0.16 compared to the best threshold value and the un-preprocessed data, respectively.

In summary, CID-GP was also helpful for increasing the reliability of peptide identification done by SEQUEST as a database search engine. By filtering more noise peaks and retaining sufficient signal peaks, it increased the average of confidence scores of identified peptides and reduced the standard deviation of these scores.

3.4.4 Analysis of the Evolved GP Program

Figure 3.15 shows the best GP-evolved program using 40 features (CID-GP). It can be seen that the GP tree uses features f2, f7, f7, f12, f12, f14, f16, f16, f19, f21, f29, f29, f31, f34 which correspond to the groups (2), (3), (4), (5), and (7) of Table 1. Analysis of the GP tree reveals that the features "Normalised Intensity", "Top X Intensities in Win ± Z", "Local Intensity Rank in Win ± Z", "Global Rank", and "Sister Ions" are good discriminators which help GP distinguish signal peaks from noise peaks. This is the evidence of why CID-GP gets better results compared with GenGP which only uses 2 groups of the features above (normalised intensity and global rank). Also appearing the "Sister Ions" features in the evolved GP program confirms the result of experiment I (investigating the important ion types) where it was expected that ion types such as neutral losses can help GP identify the signal peaks from noise peaks. So it can be seen that sister ion features have been found by GP and help GP distinguish the signal peaks from noise peaks.

3.5 Chapter Summary

The goal of this chapter was to develop an effective preprocessing method to filter noise peaks and identify the signal peaks for improving the reliability of peptide identification using highly noisy CID spectra. The goal has been successfully achieved by proposing a classification-based preprocessing method using GP to classify peaks to signal or noise peaks. As the MS/MS data is highly imbalanced, average accuracy of true positive rate and true negative rate was used as the fitness function of GP, and this helped GP not be biased towards the accuracy of the majority class containing noise peaks. A suitable gold standard MS/MS dataset containing thousands of MS/MS spectra was created and used as the training set of the GP system. Meanwhile, a set of suitable spectral features based on the CID fragmentation rules was extracted from the dataset. With the tree-based representation of GP, feature selection was implicitly applied during the evolutionary process and the analysis of a GP model revealed the important spectral features that have better discrimination ability. The experiments showed that the GP-based preprocessing method improved the reliability of peptide identification and increased the identification rate of PEAKS by 26.6% compared to the un-preprocessed data and 19.3% over the threshold-based method. Moreover, the results of SEQUEST, the database search tool, using the data preprocessed by GP were statistically significantly better than those with the un-preprocessed data and the best threshold-based method.

The single objective GP approach showed promising results in handling the highly imbalanced MS/MS data. From the results of Figure 3.5 on Page 100, it is found that the two metrics *SP* and *SN* are conflicting with each other. Although the effective weighted sum fitness function in the single objective GP is able to look after the accuracies of both majority and minority classes, it is worth investigating how multi-objective GP is able to handle this problem. This investigation is addressed in the next chapter.

Chapter 4

Multi-objective GP for Classification of Highly Imbalanced Tandem Mass Spectrometry

4.1 Introduction

In Chapter 3, GP was successfully used in solving imbalanced classification problems aiming at improving the reliability of peptide identification. Being a population-based problem solving technique, GP has been proved to be more stable compared to the other six investigated classification algorithms including decision tree (DT), k-nearest neighbour (K-NN), multilayer perceptron (MLP), naive Bayes (NB), random forest (RF), and support vector machines (SVMs) when the S/N in the MS/MS data decreases.

However, working with imbalanced data is difficult as uneven distribution of class examples in the train dataset could leave the learning algorithm with a performance bias, resulting a high majority class accuracy and a poor performance on the minority class [30]. As it was seen in the previous chapter (Experiment II) that *SP*, which is the accuracy of the majority class (noise), is in conflict with *SN*, which is the accuracy of the minority class (signal). This means that increasing one results in decreasing the other one. Therefore, classification of imbalanced MS/MS spectra can be served as a multi-objective classification problem.

Moreover, as the objective preference information from the decision maker is usually *a priori* built into the learning algorithm, in practice it is very difficult to obtain sufficient preference information and accurately represent the decision maker's preferences. Subsequently after finding the best satisfying solution, any change to the decision maker's preference requires to start the search process with the new preference information again.

Evolutionary multi-objective optimisation (EMO) as *a posteriori* method provides a set of Pareto optimal solutions prior to the decision maker's preference. So during a single run, a set of non-dominated solutions along the trade-off surface is found and then the decision maker can choose one of them based on his/her preference. EMO techniques have been effectively used to solve many real-world problems [186, 197].

A number of different EMO methods have been proposed [198]. There have been successful attempts to use GP and Pareto dominance-based algorithms to solve the class imbalanced problem by maximising two conflicting objectives, the classification accuracy of the minority and majority classes [36].While Pareto dominance-based algorithms usually produce non-dominated solutions around the centre of the Pareto front, decomposition-based EMO algorithms benefit from having the ability of differently allocating resources to better approximate the Pareto front [37].

Decomposing a multi-objective optimisation problem (MOP) into a set of scalar sub-problems and simultaneous optimising them, MOEA/D (multi-objective evolutionary algorithm based on decomposition) is an efficient framework for EMO. Neighbourhood is an essential property of MOEA/D. It uses evolutionary operators to combine good solutions of neighbouring problems for better convergence.

As GP proved to be a promising tool in MS/MS analysis, its potential for further improvement in handling two conflicting objectives of majority and minority classes using MOEA/D has not been investigated. Therefore, this chapter aims to extend the previous single objective GP approach by developing a multi-objective GP (MOGP) approach using the MOEA/D framework to evolve a set of solutions which maintain the best trade-off between these two conflicting objectives.

4.1.1 Chapter Goals

The main goal of this chapter is to develop an MOGP approach based on MOEA/D, named MOGP/D, to solve the class imbalanced problem by maximising two conflicting objectives, the accuracy of the minority class and the accuracy of the majority class in imbalanced MS/MS spectra. It is expected that the proposed MOGP/D algorithm can evolve a Pareto front of classifiers along the optimal trade-off surface that offers the best compromises between the majority class and minority class accuracies. The classifier with the best trade-off can be used to preprocess the imbalanced MS/MS spectra prior to peptide identification in order to reduce the noise peaks and to retain the signal peaks. The following objectives are specifically investigated:

1. Investigating the stability of an MOGP/D method with the decrease in the S/N ratio in the MS/MS data in terms of convergence to the Pareto front and comparing the results with MOGP based on NSGA-II (named NSGP), the popular elitist non-dominated sorting genetic algorithm method [76].

2. Analysing the classification performance of the best compromise solutions evolved by MOGP/D and NSGP and comparing them with the best solutions evolved by the single objective GP (SGP) approach.

3. Investigating the impact of the best compromise solution by MOGP/D on improving the peptide identification reliability and compare it with SGP.

4.1.2 Chapter Organisation

The reminder of this chapter is organised as follows. Section 4.2 describes the proposed MOGP/D method. Section 4.3 explains the experiment design including the benchmark multi-objective algorithm, evaluating metrics and a set of experiments which address the goals of this chapter. Section 4.4 provides the results and analysis. Section 4.5 concludes this chapter.

4.2 The Proposed Approach

This section describes the proposed MOGP approach based on MOEA/D and explains the objective functions, the new weight vector initialisation method and the evolutionary parameters used in this method.

4.2.1 Overview of the Method

In this section, the proposed MOGP method based on the MOEA/D framework, for addressing the class imbalanced problem in MS/MS spectra, is explained and the new algorithm is named MOGP/D. First, the two conflicting objectives in the multi-objective classification of MS/MS spectra are introduced. Then the modification applied on MOEA/D to effectively initialising and allocating the weight vectors are described.

The pseudo-code of the evolutionary search algorithm to simultaneously evolve a set of GP solutions along the learning objectives is presented in Algorithm 1, and at the end of the section, the MOGP/D parameter settings are explained.

Objective Functions

The two conflicting objectives in the classification of highly imbalanced MS/MS spectra include accuracy of minority class (*SE*) and accuracy of

Algorithm 1: Pseudo-code of the proposed MOGP/D approach

```
Input  : MOP; NGen: number of generation as stopping criterion; N: number
         of sub-problems; a set of uniformly distributed N weight vectors; T:
         number of neighbours; $\sigma$: probability of selecting the parents from the
         neighbourhood.
Output: an external population (EP) as the final optimal Pareto front.
Initialisation:
Set EP = $\emptyset$;
Generate initial weight vectors based on Equation (4.3) and calculate the
  Euclidean distance between any two vectors;
For i = 1, .., $N$, find $T$ closest weight vectors to the weight vector of the $i$-th
  sub-problem, and denote B(i) as its neighbouring set;
Randomly initialise each GP individual to create the population $P$ where each
  individual in P is the candidate solution of the $i$-th sub-problem;
Initialise the reference point $z^*$;
$gen \leftarrow 0$
while $gen \leq maxGen$ do
    for $i$ = 1 to Popsize do
        Reproduction:
            $$Ne = \begin{cases} B(i), & \text{if } rand < \sigma \\ P, & \text{otherwise} \end{cases}$$
          Randomly select two solutions from Ne (either from the neighbouring
          set, B(i), or from the whole population, P) to generate a new solution $y$
          by using the genetic operators;
        Update of z: for each j=1,..., $N_{obj}$ if $f_j(y) < z_j$ then $z_j = f_j(y)$;
        Update of Neighbouring Solutions: update solutions of neighbouring
          sub-problems if the fitness value of y (F(y), based on Equation (2.5) on
          Page 54) is better than the solutions of sub-problem.
        Update of EP: remove all weight vectors dominated by F(y) from EP,
          and add F(y) to EP if it is not dominated by any vector in EP.
    end
    $gen \leftarrow gen + 1$
end
return EP
```

majority class (SP) which both are presented in Equation (4.1).

$$SE = \frac{TP}{TP + FN} \quad ; \quad SP = \frac{TN}{TN + FP} \tag{4.1}$$

where TP and FN count the number of correctly and incorrectly classifying signal peaks, respectively. TN and FP represent the number of correctly and incorrectly classifying noise peaks, respectively.

In order to convert the multi-objective classification to a minimisation problem, the two objective functions in Equation (4.1) are normalised into the following form, shown in Equation (4.2).

$$f_1(x) = 1 - SE \quad ; \quad f_2(x) = 1 - \ SP \tag{4.2}$$

Problem-specific Weight Vector Initialisation

Reference point, which represents an idealised solution, is one of the most effective ways to give preference information to the EMO algorithm. The preference information can be interpreted as the preferred goal that the decision-maker is wanting to get. As in standard MOEA/D, preference information is usually provided by using uniformly distributed weight vectors with the same reference point, here we use a problem-specific method to initialise the weight vectors.

Starting with an effective set of weight vectors can ensure generating a good approximation of the Pareto front. Therefore, the initial weight vectors are designed to satisfy the following conditions:

$$\sum_{i=1}^{N_{obj}} w_i = 1.0, \text{ and } w_i \in \{0, \frac{1}{N}, \frac{2}{N}, .., 1.0\} \tag{4.3}$$

where N is the number of sub-problems which equals to the population size.

As previously mentioned, the pseudo-code in Algorithm 1 presents the overall framework of the MOGP/D algorithm for classification of highly imbalanced MS/MS spectra. The input to the algorithm is the information of MOP (based on Equation (2.4) on Page 52) and a set of parameters, and the output is an external population (EP) used to store the solutions of the Pareto front.

4.2.2 MOGP/D Setup and Evolutionary Parameters

To represent the MOGP/D individuals, the tree-based GP structure is used. The same set of internal set and function set designed for the single objective GP approach from Table 3.2 on Page 84 is considered for the MOGP approaches as well. A set of spectral features extracted from the MS/MS spectra along with randomly generated floating point numbers are used as the GP terminal set. Four arithmetic operators, addition (+), subtraction (-), multiplication (×) protected division (%, dividing by zero gives the result of 1) and *sin* function are used as the function set of GP. For the purpose of binary classification strategy, if the output of the GP program, which is a floating point number, is positive (including zero), the instance under investigation is classified into the minority class (signal class), whereas a negative GP output points to the majority class (noise class).

The parameters used for both MOGP approaches in this chapter are as the following. For initialising the population, the ramped-half-and-half method is used. The population size is 512 and the evolutionary process runs for a maximum of 100 generations. For SGP, the population size and maximum number of generation were 1024 and 50, respectively. The maximum number of 100 iterations allows the EMO algorithm have higher chance for better convergence and diversity. However, to reduce the computational cost, the population size is considered half of that with SGP. Overall, the total number of evaluations for both SGP and MOGP algorithms remains the same. The crossover and mutation rates are 80% and 20%, similar to the parameter setting in the SGP approach in Table 3.2 (where crossover, mutation and elitism rates were 80%, 19% and 1%, respectively). In MOGP, selecting the fittest individuals at each generation ensures not loosing the non-dominated solutions during the evolutionary process, and this maintains elitism in the population. The maximum program depth is restricted to 8 in order to prevent bloating.

For MOGP/D, the number of neighbours, T, for each sub-problem is set to $\frac{N}{10}$ where N is the population size. The maximum number of individuals replaced by each child is 1, and the probability that parent solutions are selected from neighbourhoods, σ, is 0.85. The proposed MOGP/D method is implemented in Python 3.6 and uses DEAP (Distributed Evolutionary Algorithms in Python) package [199].

4.3 Experiment Design

4.3.1 Dataset

To run the experiments in this chapter, the synthetic dataset and the gold standard dataset from Table 3.6 (see Page 98) are used.

The synthetic dataset is used to investigate the stability of MOGP/D across different S/N ratios. A set of four commonly used spectral features are extracted from the synthetic dataset. More details about the features can be found in Section 3.2.1 (see Page 79) where the spectral features are explained.

The gold standard dataset is used to compare the classification performance of the best compromise solutions evolved by the multi-objective GP approach with the best GP evolved program of the single objective approach (CID-GP explained in Section 3.4.2, Page 107). A set of 40 spectral features, as explained in the design of CID-GP are extracted from the gold standard dataset. Moreover, the test set of this dataset is used to evaluate the effectiveness of the best compromise solution evolved by MOGP/D and the results are compared with CID-GP.

4.3.2 Comparison Benchmark Algorithms

NSGA-II [76] is an extension of the genetic algorithm for multiple objective optimisation, which previously has been successfully used for binary class

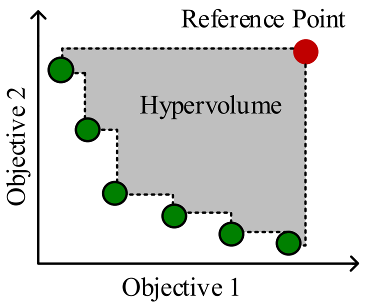


Figure 4.1: Calculating the HV value for two objectives with the presence of a reference point.

imbalanced problems [36]. The three important principles of this algorithm are non-dominated sorting, a crowding distance estimation procedure and a crowding comparison operator. NSGA-II combines the parent and offspring populations at every generation and then selects the fittest individuals for the next generation.

The NSGA-II framework as a dominance-based algorithm is used to compare with the MOEA/D framework as a decomposition based multi-objective evolutionary algorithm to investigate which is more appropriate with GP for binary classification problems with imbalanced data.

4.3.3 Evaluating Pareto Fronts

To examine the performance of the multi-objective algorithms the following indicators are used.

Hypervolume

The hypervolume (HV) indicator [200] is used to obtain a single figure indicating the convergence of the evolved Pareto front. It has been reported that HV is the most accepted metric in the EMO community specially for classification methods [201]. This is a good indicator to evaluate the performance of the evolved Pareto front generated by each EMO algorithm. Figure 4.1 illustrates how HV measures the size of the dominated space, bound from above by a reference point. Therefore, maximising A yields better approximation of the Pareto front.

Inverted generational distance (IGD) which is another popular metric to measure the convergence of the EMO algorithm is not used in this study as it requires the Pareto optimal front (true PF) in order to measure the convergence. However, in our problem, true PF is unknown. One possible approach is approximating the true PF by merging all non-dominated solutions obtained by the EMO algorithms over all runs and selecting the non-dominated solutions from the combined fronts. However, the results of performance comparison of EMO algorithms based on IGD highly depend on the specification of reference points as the points are not always uniformly distributed over the entire PF [202]. Therefore in this study, HV would be sufficient to compare the performance of MOGP methods.

Best Compromise Solution:

In order to offer a single solution from the Pareto front to the decision maker, the concept of the best compromise solution [203] is used. For each solution of the Pareto front, a normalised fuzzy membership value is calculated and the solution with the maximum value is selected as the best compromise solution. In this method, the *k*-th member of the Pareto front for its *i*-th objective value has a fuzzy membership value of μ_i^k. The membership function μ_i varies between 0 and 1 and is defined by Equation (4.4). The membership function μ_i, which varies between 0 and 1, demonstrates the *i*-th objective function F_i, and is defined by Equation (4.4).

$$\mu_i^k = \begin{cases} 1, & F_i \leq F_i^{min} \\ \dfrac{F_i^{max} - F_i}{F_i^{max} - F_i^{min}} & F_i^{min} < F_i < F_i^{max} \\ 0, & F_i \geq F_i^{max} \end{cases} \tag{4.4}$$

where F_i is the *i*-th objective function, and F_i^{min} and F_i^{max} are the minimum and maximum values of the *i*-th objective function among the set of solutions found along the Pareto front. The k-th solution of the Pareto front has the normalised membership function μ^k which can be expressed based on Equation (4.5).

$$\mu^k = \frac{\sum_{i=1}^{N_{obj}} \mu_i^k}{\sum_{k=1}^{M} \sum_{i=1}^{N_{obj}} \mu_i^k} \tag{4.5}$$

where M is the number of solutions in the Pareto front. The solution with having the highest value of μ^k is selected as the best compromise solution. More details about how to obtain the best compromise solution can be found at [203]. After finding the best compromise solution, its *A-acc* is measured based on Equation (4.6).

$$A\text{-}acc = 0.5 \times SE + 0.5 \times SP \tag{4.6}$$

4.3.4 Experiments

A set of experiments on the synthetic dataset (from Table 3.6) are conducted in order to investigate the performance of MOGP/D and NSGP across various ratios of S/N peaks on MS/MS data. Since the ratio of imbalance between the MS/MS datasets is different, these experiments are used to figure out which EMO algorithm produces better solutions along the Pareto front across different imbalanced ratios. As previously mentioned in Chapter 3, the synthetic dataset contains a set of six pairs of training and test sets each pair having different S/N ratios including 1:1, 1:2, 1:4, 1:6, 1:12, 1:20. 1:1 indicates a balanced dataset, while 1:20 implies 20 times of noise peaks over signal peaks in the dataset. The performance of each EMO algorithm is evaluated based on the indicators explained in Section 4.3.3.

After investigating the stability of the EMO algorithms on the synthetic dataset, another experiment to investigate the classification performance of MOGP/D and NSGP on the gold standard dataset is conducted and the results of the best compromise solutions of the two MOGP algorithms are compared with the best evolved GP program from the single objective GP approach proposed in Section 3.4.2. Finally, the effectiveness of the proposed

Table 4.1: Average (± standard deviation) HV of the *all* evolved Pareto fronts obtained by MOGP/D and NSGP on the *training sets* of the synthetic dataset across different S/N ratios over the 30 MOGP runs.

	1:1	1:2	1:4	1:6	1:12	1:20
MOGP/D	0.163±0.001 (↓)	0.160±0.001 (o)	**0.156±0.001** (↑)	**0.154±0.001** (↑)	**0.144±0.001** (↑)	**0.140±0.001** (↑)
NSGP	0.0.166±0.007	0.160±0.002	0.154±0.003	0.153±0.001	0.142±0.001	0.135±0.001

MOGP/D method in terms of improvement in the reliability of peptide identification is investigated and the results are discussed.

4.4 Results and Discussions

4.4.1 Analysis of the Overall Pareto Front Behaviour in Terms of HV

Table 4.1 presents the results of the average HV values of the *all* evolved Pareto fronts obtained by MOGP/D and NSGP on the six training sets of the synthetic dataset with each set having a different S/N ratio. To calculate the HV values, the reference point (0.5, 0.5) is used.

To compare the performance of the two EMO algorithms over the 30 GP runs, the statistical t-test with 95% confidence interval and two-tailed P value less than 0.0001 is considered. The signs below the HV values show the significance test results, where (↑)/(↓) indicates that MOGP/D is significantly better/worse than NSGP. Also (o) sign is used to represent that the result of MOGP/D is not significantly different from NSGP.

The overall trend in the results of Table 4.1 shows that with the decrease in the S/N ratio (i.e. increasing class imbalanced rate) in the MS/MS data on the training sets, the HV values of both EMO algorithms decrease. This indicates that higher imbalanced data makes the classification problem more

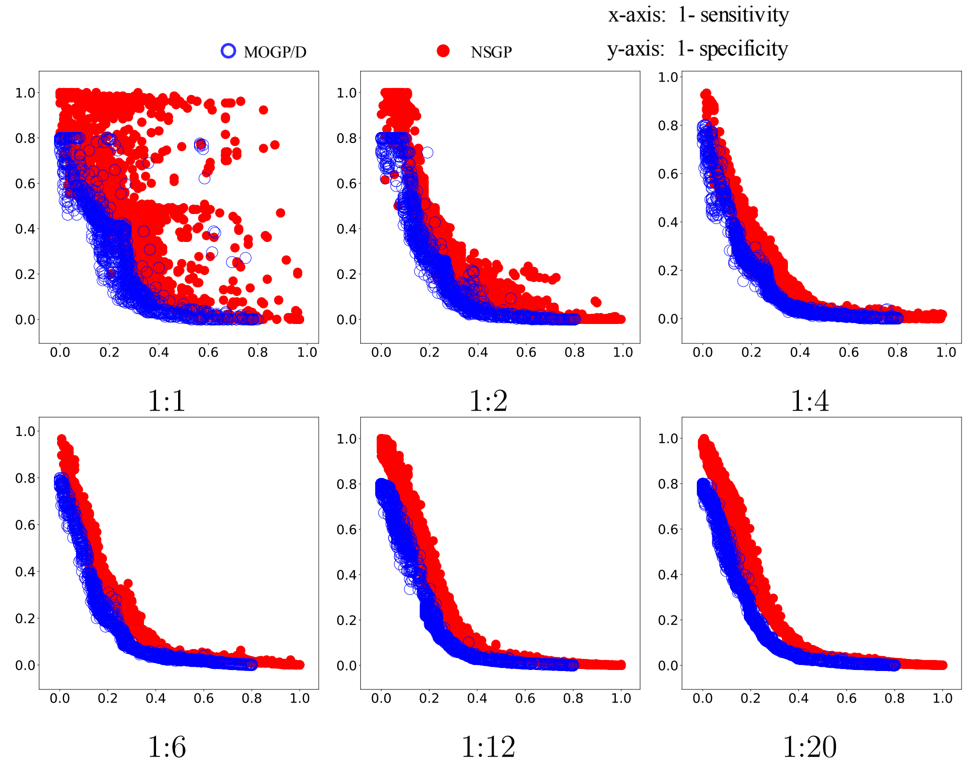


Figure 4.2: Classification performance of the *all* evolved solutions using MOGP/D and NSGP on the *test sets* with different S/N ratios.

difficult. Moreover, in terms of statistically comparing the performance of the two EMO algorithms, it can be seen that with the decrease in the S/N ratio in the training sets, MOGP/D achieves significantly better HV values than NSGP. This result could indicate that MOGP/D is more stable than NSGP in approximating the Pareto front when the S/N ratio is getting higher. As the problem is becoming more difficult, MOGP/D shows better convergence than NSGP.

However, the performance of NSGP when the dataset is balanced is statistically significantly better than MOGP/D. This is interesting and needs further analysis, which is done by visualising the performance of the evolved solutions by the two EMO methods over the 30 independent GP runs on the test sets of the synthetic dataset (see Figure 4.2).

Based on the results of Table 4.1, although the performance of NSGP

Table 4.2: Average (± standard deviation) GP tree size of the *all* evolved Pareto fronts obtained by MOGP/D and NSGP on the *training set* with S/N ratio of (1:1) over the 30 MOGP runs.

Method	GP tree size
MOGP/D	76.88 ± 17.87
NSGP	164.64 ± 33.87

on the balanced training set (1:1) is significantly better than MOGP/D, the plots in Figure 4.2 show that the results of NSGP on both balanced (1:1) and other imbalanced test sets are worse than MOGP/D. This means that most solutions of NSGP are dominated by those of MOGP/D on test sets. This indicates that the evolved solutions by NSGP have lower generalisation performance than those of MOGP/D.

Further analysis on the average (and standard deviation) size of the GP program solutions evolved by both methods over 30 MOGP runs is presented in Table 4.2. It can be seen that NSGP produces bigger GP programs than MOGP/D. To summarise the analysis of the size of the GP programs evolved by NSGP, an unnecessary growth of the GP tree known as bloat or code growth is seen. This problem could be resolved by considering the GP program size as a third objective besides the other two objectives related to the program functionality, since a number of research have been successfully used Pareto-based approaches for bloat controlling. However, without focusing on bloat control as the third objective, the plots in Figure 4.2 shows that MOGP/D is able to indirectly handle the bloat problem, resulting in a significant improvement in the objective values when the test sets are used.

4.4.2 Analysis of the Pareto Optimal in Terms of HV

Figure 4.3 illustrates the HV values of the Pareto fronts generated by combining all evolved Pareto fronts across the 30 MOGP runs into a set of non-dominated solutions. Combining all Pareto fronts in order to obtain a potentially different set of non-dominated solutions is a useful technique in EMO as it summarises the outcome of a series of MOP runs. The overall

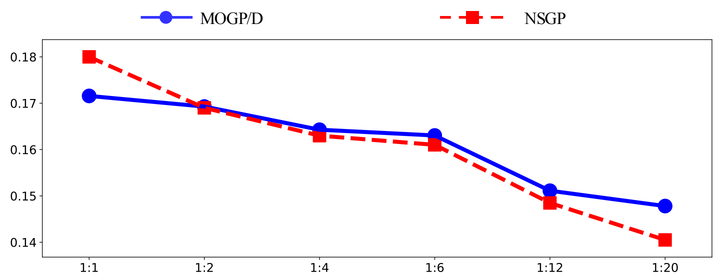


Figure 4.3: HV of the non-dominated fronts generated by combining all evolved Pareto fronts obtained by each EMO algorithm (MOGP/D and NSGP) over the 30 MOGP runs on the *training sets* of the synthetic dataset with each set having different S/N ratio.

trends in plots are consistent with the results from Table 4.1 and Figure 4.2, which means that with the decrease in the S/N ratio in the training sets, MOGP/D produces better solutions compared to NSGP. More analysis shows that the number of evolved solutions using MOGP/D is more than that of NSGP across all training sets.

4.4.3 Analysis of the Best Compromise Solutions

In the multi-objective classification of MS/MS data, after obtaining a set of non-dominated solutions, the decision maker needs to select a single classifier for preprocessing the MS/MS data prior to peptide identification. Therefore, here we use a fuzzy membership technique to find the best compromise solution among all evolved solutions. So, with respect to each Pareto front across the 30 runs, one single best compromise solution is obtained for each EMO algorithm. For each single best compromise solution, A-*acc* according to Equation (4.6) is calculated. Table 4.3 presents the results of A-*acc* of all best compromise solutions with respect to the 30 independent MOGP runs for two EMO algorithms on the training sets with different S/N ratios.

The overall trends in the results of Table 4.3 show that the decrease in the S/N ratio results in the decrease in the classification performance of evolved solutions by both MOGP/D and NSGP. From the results in Table 4.3, it can be seen that similar to the results of Table 4.1, with the decrease in the S/N

Table 4.3: Average (± standard deviation) accuracy of the best compromise solutions selected from the *all* evolved Pareto fronts obtained by MOGP/D and NSGP on the *training sets* with different S/N ratios over the 30 MOGP runs.

	1:1	1:2	1:4	1:6	1:12	1:20
MOGP/D	0.829±0.002 (↓)	0.828±0.004 (o)	0.825±0.001 (o)	**0.824±0.002** (↑)	**0.822±0.001** (↑)	**0.819±0.001** (↑)
NSGP	**0.831±0.013**	0.827±0.007	0.824±0.004	0.822±0.003	0.811±0.003	0.802±0.002

Table 4.4: Average (± standard deviation) accuracy of the best compromise solutions over the 30 MOGP runs evaluated on the *test sets*.

	1:1	1:2	1:4	1:6	1:12	1:20
MOGP/D	**0.704±0.072** (↑)	**0.757±0.027** (↑)	**0.796±0.009** (↑)	**0.804±0.005** (↑)	**0.805±0.004** (↑)	**0.804±0.004** (↑)
NSGP	0.556±0.101	0.700±0.032	0.751±0.015	0.750±0.009	0.761±0.007	0.754±0.005

ratio, the best compromise solutions evolved by MOGP/D outperform those of NSGP in terms of *A-acc* on the training sets. Looking more closely at the results of both algorithms when the dataset is balanced, (1:1), NSGP is statistically significantly better than MOGP/D, however based on the test results in Table 4.4 the performance of NSGP at (1:1) dramatically drops. Based on the results of Table 4.4 on the test sets, MOGP/D shows better performance than NSGP across all different S/N ratios.

Finally, to compare the results of MOGP with SGP, the best compromise solution of each EMO algorithm from its non-dominated front after combining all 30 Pareto fronts over the 30 GP runs are obtained. Figure 4.4 presents the classification performance of the best compromise solutions evolved by MOGP/D and NSGP along with the best SGP solution of GenGP in Section 3.3.3. The results of classification in terms of *SN*, *SP* and *A-acc* are

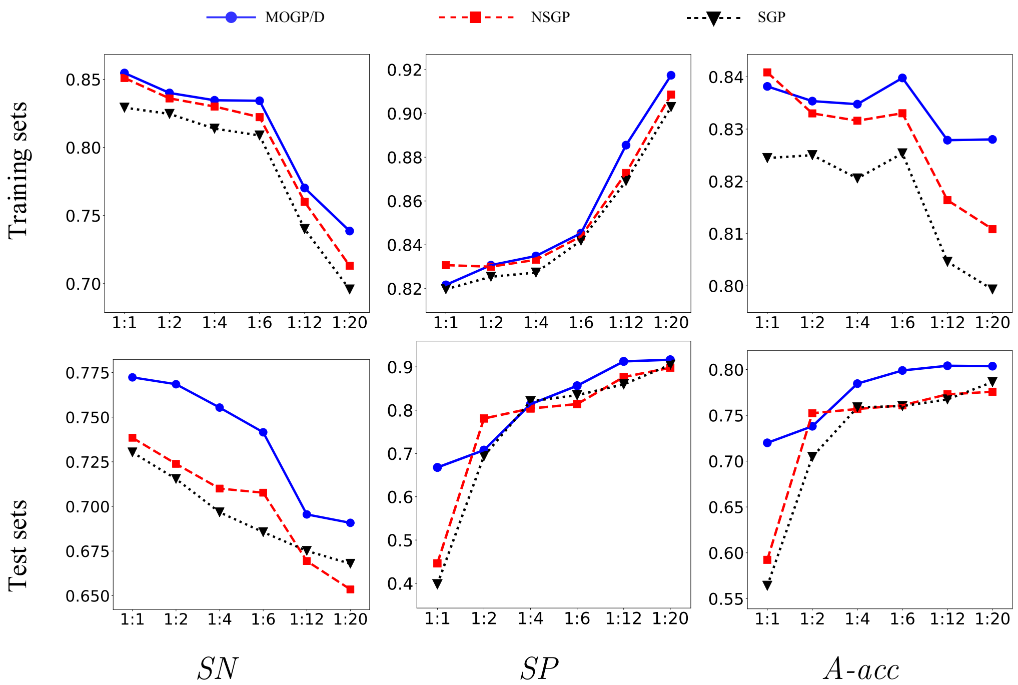


Figure 4.4: Classification performance of the best SGP and the best compromise solution of MOGP/D and NSGP on the *training* and *test sets* with different S/N ratios.

plotted. Starting from the *SN* results on the training set, it can be seen that all three methods have a steady decline from 1:1 to 1:6 followed by a sharp drop at 1:20. This shows that decreasing the S/N ratio results in misclassification of more signal peaks. However, the results show that MOGP/D and NSGP outperform SGP in terms of *SN* on the training sets. Also on the test set, the same decline in the *SN* results of all three methods can be seen. So in summary, MOGP/D outperforms both methods in terms of *SN* on both the training and test sets. This is very important in classification of peaks in MS/MS spectra, as retaining the signal peaks as much as possible is more important than filtering out the noise peaks.

The *SP* results on the training sets in Figure 4.4 shows that as the number of noise peaks increases, the classification performance of all three method on the majority class increases as well. Comparing the plots of *SN* and *SP* on both the training and test sets obviously shows the conflict between these

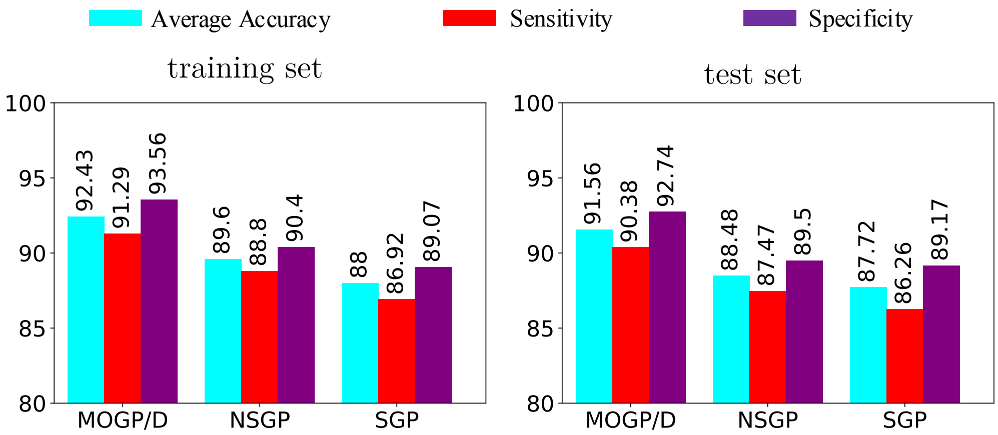


Figure 4.5: Classification performance of the best SGP and the best compromise solution of MOGP/D and NSGP on the gold standard dataset.

two objectives. Similar to the results of *SN* plots, with the decrease in the S/N ratio, MOGP/D outperforms NSGP and SGP in terms of *SP* on the training and test sets.

Finally, the results of *A-acc* show that MOGP/D and NSGP outperform SGP on the training sets. On the test sets, SGP and NSGP almost have the same performance, while MOGP/D outperforms both of them. Overall, with the decrease in the S/N ratio of the MS/MS data in the training set MOGP/D shows constant accuracy at 83% whereas these values for NSGP and SGP are at >81% and >80%, respectively.

4.4.4 Classification Performance on the Gold Standard Dataset

Figure 4.5 presents the classification performance of the best compromise solutions evolved by MOGP/D and NSGP along with the best SGP solution of CID-GP proposed in Section 3.4.2 on the gold standard dataset. The bar charts presenting the results of classification in terms of *A-acc*, *SN*, and *SP*. As the S/N ratio in the gold standard dataset is almost 1:30, the classification results on the training set in Figure 4.5 are consistent with those in Figure 4.4

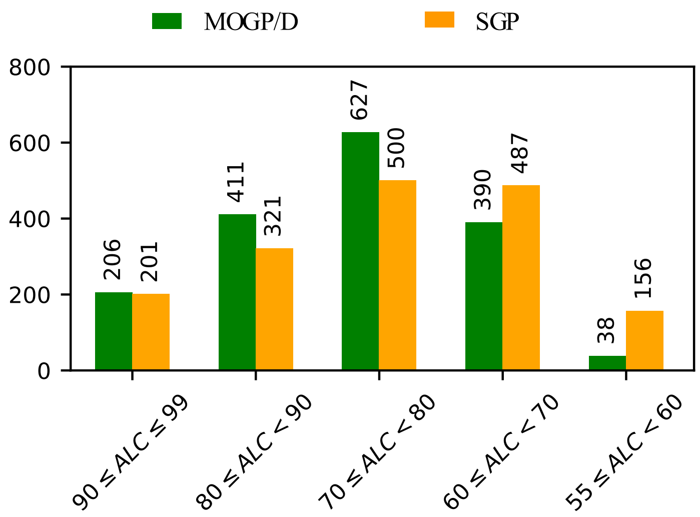


Figure 4.6: The results of peptide identification by PEAKS on the *test set* of the gold standard dataset preprocessed by the best compromise solution evolved by MOGP/D and the best GP program evolved by the single objective CID-GP in different ALC ranges.

at 1:20 S/N ratio, where both MOGP methods outperform SGP in terms of *A-acc*, *SN*, and *SP*.

As the gold standard dataset contains 40 spectral features compared to the synthetic dataset which contains only 4 features, and the gold standard dataset includes more number of instances, it can be seen that the classification performance of the best compromised solutions evolved by MOGP/D and NSGP are significantly improved compared to those on the synthetic dataset. Comparing both MOGP algorithms, it can be seen that MOGP/D outperforms NSGP by 2.49% and 2.91% of retaining more signal peaks on the training and test set of the gold standard dataset, respectively.

In summary, the MOGP/D algorithm has shown to be promising in evolving better solutions with compromises between the two conflicting objectives of *SN* and *SP* in the classification of highly imbalanced MS/MS data. In the next section, the preprocessed MS/MS data by the best compromise solution evolved by MOGP/D is submitted to PEAKS for peptide identification and the results are analysed to evaluate the effectiveness of the prepossessing method.

4.4.5 Performance Evaluation

The results of evaluating the effectiveness of MOGP/D in terms of improvement in the reliability of peptide identification on the test set of the gold standard dataset is presented in Figure 4.6. Overall, MOGP/D has significantly improved the peptide identification compared to SGP, since MOGP/D retains more signal peaks and removes more noise peaks compared to SGP.

Although there is not a significant difference between the results of the two methods with ALC higher than 90%, for $70 \leq ALC < 90$, the results shows that MOGP/D obviously improves the reliability of peptide identification by helping PEAKS identify more high confident peptides than SGP. Adding up all peptide identified with ALC higher than 70, the results show that MOGP/D improves the reliability of peptide identification by 21.72% ($= \frac{(206+411+627)-(201+321+500)}{(201+321+500)} \times 100$) compared to SGP.

4.5 Chapter Summary

The goal of this chapter was to develop an effective MOGP/D method based on the idea of MOEA/D to evolve a set of non-dominated solutions, along the optimal trade-off surface that offers the best compromises between the two conflicting objectives of *SN* and *SP*. The non-dominated solutions are used for classification of peaks in the MS/MS spectra. The goal has been successfully achieved by developing an MOEA/D based MOGP method. Compared with NSGP, an NSGA-II based MOGP method, MOGP/D evolves more solutions in the middle of Pareto front, pushing this front towards better minority (signal) and the majority (noise) class accuracies.

As the MS/MS spectra is highly imbalanced, the stability of the proposed MOGP method with the decrease in the S/N ratio in the MS/MS data was investigated and the results were compared with NSGP in terms of convergence to the Pareto front. The HV value was used as a single figure for the purpose of comparison. The results showed that with decreasing S/N, MOGP/D outperformed NSGP in terms of the HV values of the evolved

Pareto fronts on both the training and test sets.

For selecting a single solution from the evolved Pareto front, a fuzzy membership approach was used to obtain the best compromise solution. The single classifiers from MOGP/D and NSGP were compared with SGP. The results showed that MOGP/D outperformed NSGP and SGP in terms of *SN*, *SP* and *A-acc* on both the training and test sets. In other words, MOGP/D has shown to be more suitable for evolving a classifier that has the best trade-off between the two conflicting objectives of the majority and minority class accuracies in the problem of classification of imbalanced MS/MS spectra.

As CID-GP in Section 3.4.2 showed that adding more spectral features helped GP for more accurate distinguishing the signal peaks from noise peaks (see Figure 3.13 on Page 112) and CID-GP improved the reliability of peptide identification (see Figure 3.14 on Page 113), the gold standard dataset with having 40 spectral features was used to train the MOGP/D and NSGP methods. The results showed that MOGP/D outperformed both CID-GP, which is a single objective GP approach, and NSGP, knows as a multi objective GP approach. MOGP/D retains a larger number of signal peaks and this has positive influence on the peptide identification reliability.

So far two preprocessing methods are developed with the purpose of denoising the MS/MS spectra prior to peptide identification either by a *de novo* sequencing method or a database searching tool in order to help these tools improve the reliability of peptide identification. Both methods attempt to improve the peptide identification mainly at the amino acid level. Although they might have improved the peptide identification at the peptide level as well, our purpose was evaluating them at the amino acid level. That was the reason that the changes in the confidence scores were analysed.

As previously mentioned in Chapter 1, one of the challenges of the current *de novo* sequencing methods is accuracy at the peptide level. One possible solution is developing post-processing methods to improve the result of current *de novo* sequencing at the peptide level. But, before developing those methods, we will develop our own *de novo* sequencing method using GA, as we believe it opens the door for future improvements for *de novo* sequencing methods that use dynamic programming as their search strategy to solve the complex optimisation task of *de novo* sequencing. GA can potentially overcome problems associated with noise and missing ions in *de novo* sequencing of real MS/MS data. Customisation in GA allows introducing new features to deal with common problems in this field such as di-peptide conflicts in full-length *de novo* sequencing. Next chapter explains the GA-based *de novo* sequencing method in more detail.

Chapter 5

GA for De Novo Peptide Sequencing

5.1 Introduction

The complete CID peptide fragmentation gives a contiguous series of ion types called "ladder" [99]. Having the complete ion ladder, the *de novo* sequencing algorithm selects pairs of peaks and labels them if their mass differences are within the tolerance ranges of the amino acid's masses.

However, it is often that peptide fragmentations are neither sequential nor complete. The fragmentation events are somehow random and do not necessarily start from the N-terminus of the peptide to the C-terminus (left to right). Moreover, some cleavage sites are preferred over others, so more abundant peaks for preferred sites and fewer abundant ones for other positions will be produced in the spectrum. Moreover, peptides may not fragment at some positions, resulting in missing data. Even a human expert can have difficulty to interpret the neighbouring residues when the fragmentation sites are missing. Therefore, fragmentation incompleteness is a challenge for interpreting MS/MS data in peptide identification problems. Also, a real MS/MS spectra with hundreds of peaks normally contain background noise. Therefore, while exactly 1 of 20^l amino acid sequences can be considered as the potential correct prediction (l is the peptide length), *de novo* sequencing with internal fragment ions is recognised as a combinatorial problem [51].

There have been attempts to solve the *de novo* sequencing problem using different approaches. However, *de novo* sequencing of full-length peptides remains a challenging task. *De novo* sequencing can be formulated as an optimisation problem where the objective is to discover the most likely amino acid sequence that can be generated by the input spectrum [49]. *De novo* sequencing has been performed via stochastic optimisation using a genetic algorithm (GA) [182, 185], where a GA tries to optimise the amino acid sequence with respect to a scoring function. However, the existing *de novo* sequencing methods using GA often fail to discriminate the mismatches because the fitness functions could not capture various aspects of peak matching [150]. Moreover, the basic genetic operators used in these works are not capable enough to guide GAs during the evolutionary process to construct the fully matched sequence.

5.1.1 Chapter Goals

The goal of this chapter is to develop an effective *de novo* sequencing algorithm, called GA-Novo, using GAs to construct the full-length amino acid sequences of MS/MS spectra. Considering the ability of GA to explore a large search space of potential amino acid sequences, GA is expected to infer the most likely amino acid sequence directly from the spectrum. The following objectives are investigated in this chapter:

1. Developing a new fitness function that captures important spectral features and enables GA to discriminate the mismatches.

2. Developing an effective set of mutation and crossover operators that help GA construct the full-length amino acid sequence.

3. Designing an effective GA algorithm that can perform the *de novo* sequencing task, aiming at achieving a high number of fully matched sequences out of the input spectra.

5.1.2 Chapter Organisation

The rest of the chapter is organised as follows. Section 5.2 describes GA-Novo which is the proposed *de novo* sequencing method using GA. Section 5.3 explains the experimental design including the MS/MS dataset, evaluation, and GA parameters. Section 5.4 presents the experimental results and discussions. Summary of the chapter is explained in Section 5.5.

5.2 The proposed Method

As mentioned earlier, the problem of *de novo* sequencing is an optimisation problem where the input is an MS/MS spectrum s and the output is the most likely peptide sequence p. Therefore, the optimisation problem can be formulated based on Equation (5.1).

$$\text{Maximise number of matched ions } (s,\ t) \tag{5.1}$$

$$\text{subject to } |M(s) - M(p)| < 57$$
$$length(p) > 0$$

where peptide p is the decision variable and the objective function is maximising the matched ions. t is the theoretical spectrum of peptide p based on CID fragmentation rules. There are two constraints on the decision variable p. The first one indicates that the absolute value of mass difference between s and p should not exceed 57 which is the mass of amino acid ‘G’. Glycine has the smallest mass compared to the other amino acids and this condition ensures that the sequence p is not shorter or longer than the correct match. The second constraint is the length of p which should be greater than 0 as we do not want an empty sequence with no amino acids. Therefore, the sequencing optimisation problem aims at finding an amino acid sequence that has the maximum number of matched ions between the experimental spectrum s and its theoretical spectrum t.

5.2.1 Overview of the Method

Figure 5.1 presents the workflow of GA-Novo. Given the raw MS/MS experimental spectrum s, first a tag-based initialisation method is applied in order to create a set of candidate initial individuals for the GA algorithm. The candidate individuals are kept in a big initialisation pool. The individuals are evaluated and based on three criteria, the fitness value, Nterm score and Cterm score are selected to generate the initial population for GA.

The evolutionary process starts with applying selection in order to create four pools for different purposes where the size of each pool is a third of the total population size. To the best of our knowledge, this is the first pool based GA and is proposed with the purpose of generating more suitable individuals for GAs.

The helper pool contains top best individuals in terms of fitness values. The individuals in N-term and C-term pools are the top best individuals in terms of Nterm and Cterm scores, respectively. The Nterm score is able to check whether a matched b-ion is a random match or not. Similarly Cterm score checks if a matched y-ion is a random mach. These two scores are explained in more detail on Page 153. The individuals in last pool, tournament selection pool, are selected using tournament selection based on their fitness values.

There are five genetic operators, two crossovers, two mutations and an elitism. The individuals for Nterm-Cterm crossover are selected from the first three pools. Other genetic operators get their individuals directly from the tournament pool. Nterm-Cterm crossover is designed to construct individuals with correct amino acid matches from N-terminus and the C-terminus and possibly from the middle of the sequence, whereas two-point crossover aims to repair the individuals from middle. The mutation operators randomly flip flop the each bit/amino acid in the sequence. In each generation, elitism keeps the best three individuals in terms of overall fitness value, Nterm and Cterm score. The evolutionary process repeats until the termination criterion which is the number of generations is met. The method returns the best individual

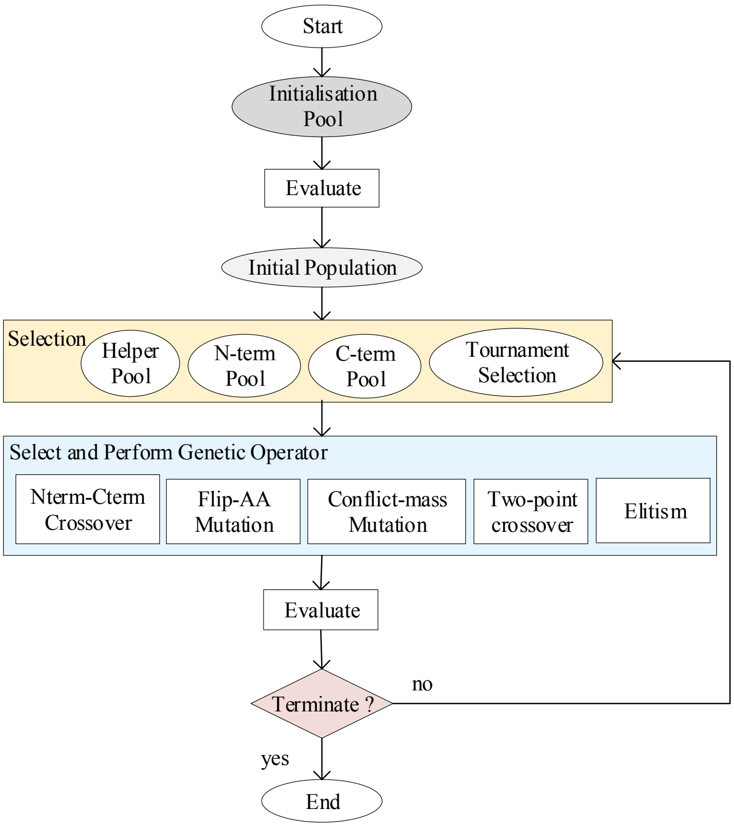


Figure 5.1: The workflow of GA-Novo.

in terms of overall fitness value. More details about the components in this flowchart are as follows.

5.2.2 Representation

Each GA individual has a variable length and is represented by a sequence of single-letter amino acids, for example "AAALAAADAR". Each individual contains three fitness scores including the overall fitness value (from the fitness function in Equation (5.5) on Page 151, Nterm and Cterm scores which are explained later.

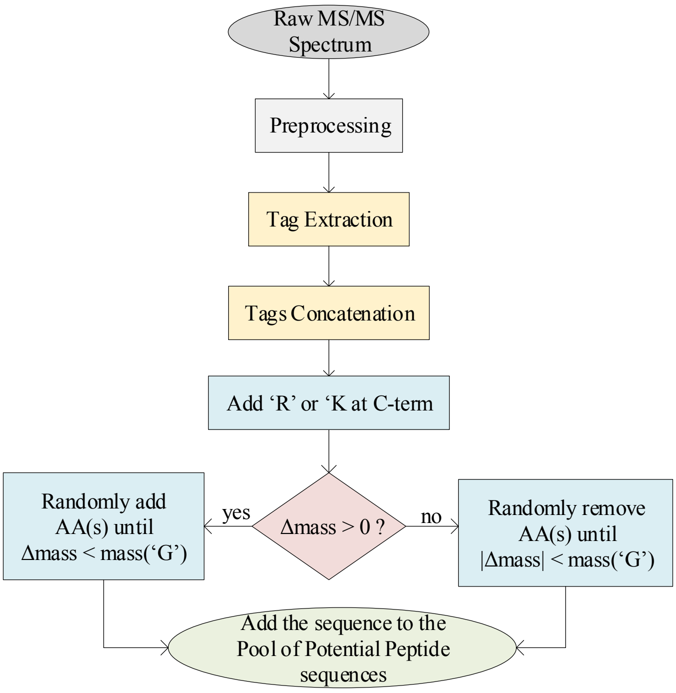

Raw MS/MS Spectrum
Preprocessing
Tag Extraction
Tags Concatenation
Add ‘R’ or ‘K at C-term
Δmass > 0 ?
yes
no
Randomly add AA(s) until Δmass < mass(‘G’)
Randomly remove AA(s) until |Δmass| < mass(‘G’)
Add the sequence to the Pool of Potential Peptide sequences

Figure 5.2: The workflow of the tag-based initialisation method.

5.2.3 Tag-based Initialisation Method

A domain dependent initialisation method is used to generate initial individuals for GA. Since GA is supposed to gradually construct the more likely amino acid sequence for the input experimental spectrum, using domain knowledge in initialisation helps GA avoid constructing incorrect sequences. The workflow of this method is illustrated in Figure 5.2. The overall goal of this method is to construct peptide sequences which are preferably partially matched with the spectrum and having as small as possible mass difference ($\Delta mass$) with the spectrum. Later in Section 5.4.2, this initialisation method is compared with a random initialisation approach.

The input to the workflow of the initialisation method is an MS/MS spectrum (experimental spectrum) and the output is a set of peptide sequences corresponding to the spectrum. The workflow starts with preprocessing the input spectrum. Then all 3-letter tags are extracted from the preprocessed spectrum in tag extraction step. In tags concatenation step, each time 2, 3 or 4 tags are randomly selected and concatenated to construct a sequence with length 6, 9 or 12. These numbers are in the range of the peptides' length that fall in the precursor mass range of spectra used in this chapter. Since all tryptic peptides have either amino acids 'R' or 'K' at the end, these two amino acids are randomly added to the end of the sequences from tags concatenation step.

As mass difference is a constraint, it is important to construct the sequences with $|\Delta masses| \geq 0$. So the rest of the workflow checks whether or not the mass difference between each constructed sequence and the spectrum is less than the mass of amino acid 'G' which has the smallest mass compared to the other amino acids. $\Delta mass$ is calculated based on Equation (5.2).

$$\Delta mass = M(s) - M(p) \tag{5.2}$$

where $M(p)$ and $M(s)$ are parent mass (mass of peptide p) and the precursor mass (mass of the spectrum s), respectively. Therefore, based on the $\Delta mass$ value, appropriate amino acids are randomly added to/removed from the sequence and the resulting peptide sequence is sent to the pool of possible peptide sequences corresponding to the input experimental spectrum. The preprocessing step and the tag extraction are explained in the following sections.

Spectrum Preprocessing

The data preprocessing step on MS/MS spectra reduces the noise and adds some of the missing data. The MS/MS noise reduction step has been done based on the noise reduction method proposed in SEQUEST [92]. Given a spectrum, first the whole m/z range is divided into 10 windows (regions). In each window, if the number of existing peaks exceeds 9, there should be some possible noise, which needs to be eliminated from that window. Noisy peaks are removed based on their intensity values. The peak intensity with the highest frequency is required to be found as it would be the noise threshold value. To find that, all peaks with the same intensity values are counted. Then the intensity value with the highest frequency is considered as the noise threshold. Therefore, all peaks whose intensities are smaller than the noise threshold will be removed from that window.

After removing these noisy peaks, the next step is normalising peak intensities. In each window, each peak's intensity is replaced with its square root and then all intensities are normalised by dividing into the highest intensity. Therefore, after normalisation, the highest peak intensity in each window equals 1.

As presented in Table 2.1 (see Page 33), a complete peptide fragmentation gives a contiguous series of ions. However, sometimes due to the low ion fragmentation efficiency of the mass spectrometer, some ions are not available in the spectrum. Then each peak in the spectrum is checked for the existence of its complementary peak which will be added if required. By finding the complementary ion peaks, undetected ions can be added to the spectrum. The sum of the two complementary ions' masses should be equal to the precursor mass of the spectrum. Therefore, the corresponding complementary peak are added with the same intensity to the corresponding *m/z* value. This algorithm only considers singly charged fragmentation ions.

So far, the spectrum is denoised and all necessary peaks are added into it. Now the next step as shown in Figure 5.2 is extracting all 3-letter tags from the spectrum.

Tag Extraction

In tag extraction, all 3-letter tags from the N-terminus to the C-terminus are extracted from the spectrum. Figure 5.3 illustrates the tag extraction process from a simplified MS/MS spectrum (a real spectrum has more peaks). The MS/MS spectrum s consists of a list of peaks each having an m/z value and an intensity value (peak height). Assume the spectrum is represented by two vectors of m/z values and intensities $s = (M, I)$, where $M =$

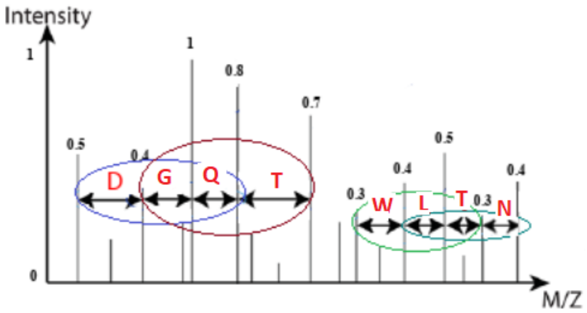


Figure 5.3: An example of the tag extraction process from a simplified MS/MS spectrum.

$(m_1, m_2, m_3, ..., m_n)$ and $I = (I_1, I_2, I_3, ..., I_n)$. Considering the M vector, two peaks construct a peak pair if their m/z values satisfy $|m_i - m_j - mass(a)| \leq \tau$ where $1 \leq i \leq j \leq n$, $mass(a)$ is the mass of one of the 20 popular amino acids and τ is the MS/MS mass tolerance. A tag with length one is represented by $t(i, j)$ and a label of a corresponding to its amino acid. Two tags $t(i, j)$ and $t(i', j')$ are considered sequential if $j = i'$. So all 3-letter tags from the spectrum will be extracted and are used in the initialisation method. Figure 5.3 shows a few examples of 3-letter tags: DGQ, GQT, WLT and LTN. In the next step, tags concatenation, these tags are concatenated to each other to create a full-length amino acid sequence.

Tags concatenation

In tags concatenation, the 3-letter tags are randomly concatenated to construct longer (full-length) sequences. As the precursor mass range of spectra used in this study is limited to 1150, the peptides' length that fall in this range could be around 6 to 15. Therefore, each time 2, 3 or 4 tags are randomly selected and concatenated to construct a sequence with length 6, 9 or 12. Since all tryptic peptides have either amino acids 'R' or 'K' at the end, these two amino acids are randomly added to the end of the sequences. Then, the mass difference between the concatenated peptide sequence and the experimental spectrum is calculated based on Equation 5.2. As shown in

Figure 5.2 if the absolute value of the mass difference is more than the mass of amino acid ‘G’, then according to the value of $\Delta mass$, appropriate amino acids are randomly added to/removed from the sequence.

Two concatenated sequences from the tags in Figure 5.3 for the experimental spectrum with precursor mass of 760 are presented as follow.

- ‘DGQGQTR’ with parent mass of 760.3 which is resulted from concatenation of DGQ + GQT and adding R to the end of the sequence.

- ‘GQTWTK’ with parent mass 719.3 which is resulted from concatenation of GQT + WLT, adding K to the end of the sequence and removing L to relax the mass difference.

5.2.4 Fitness

Fitness Function

The fitness function evaluates the quality of matching between an input experimental spectrum and a peptide sequence constructed by GA-Novo. As s is composed of a set of m/z values with their corresponding intensities, and p is a linear combination of amino acids, these two entities should be converted to a format to allow the spectrum to be matched with the peptide. Therefore based on the CID fragmentation rules of doubly charged peptides (as our research only focuses on doubly charged peptides), for each peptide its corresponding theoretical MS/MS spectrum of all b- and y-ions is virtually constructed [190]. The theoretical spectrum only contains m/z values with equal intensities of 1. Both b-/y-ion ladders in Table 2.1 along with internal fragments are constructed in the theoretical spectrum.

Equation (5.3) and Equation (5.4) show how to calculate the b-ions and y-ions of the theoretical spectrum t, respectively. Having the peptide p with length l, the mass of each theoretical b-ion in its b-ion ladder is calculated based on Equation (5.3) [90].

$$b_j = \sum_{i=1}^{j} mass(a_i) + 1 \tag{5.3}$$

where $mass(a_i)$ is the mass of i-th amino acid in peptide p, b_j is the j-th b-ion of p and $1 \leq j \leq l-1$. The constant 1 is the result of a proton transfer that could take place from the amide nitrogen of the adjacent amino acid residue that precedes the peptide bond, which undergoes the CID dissociation.

Equation (5.4) [44] presents how to calculate the theoretical y-ions from the y-ion ladder of peptide p.

$$y_j = \sum_{i=l-j}^{l} mass(a_i) + 19 \tag{5.4}$$

where $mass(a_i)$ is the mass of i-th amino acid in peptide p, y_j is the j-th y-ion of p and $1 \leq j \leq l-1$. It can be seen that to calculate the y-ions, the constant 19 has been added to the sum of residue masses. The reason is that y-ions are the fragment ions that possess the -COOH of the precursor ion. In fact the mass of singly charged y-ions are calculated as sum of the amino acid residue masses in the fragment + a mole of water (18) + a proton transfer (1).

Therefore converting the peptide p to the theoretical spectrum t, can let us to match/compare the two experimental and theoretical spectra against each other in order to measure the similarity between them. Peaks in the theoretical spectrum are matched against the peaks in the experimental spectrum within the MS/MS mass tolerance of τ.

Finally, Equation (5.5) presents the new fitness function for measuring the goodness of the peptide spectrum match (PSM).

$$fitness(PSM) = \frac{\sum I_{matched}}{\sum_{i=1}^{n} I_i} - \frac{|\Delta mass|}{\text{Prec.}_{mass}} + \frac{Nterm + Cterm - \sum N_{unmatched}}{length(P)} \tag{5.5}$$

where $I_{matched}$ is the sum of intensities of those peaks in the experimental spectrum s which are matched with theoretical spectrum t corresponding to the peptide p. As normally b-/y-ions tend to have higher intensities, the total intensities of all matched peaks could be a better indicator for distinguishing a correct match rather than only considering the total number of matched peaks. The total intensities of matched peaks is normalised by dividing into the total intensities of the whole spectrum s. $\Delta mass$ is the mass difference between parent mass of peptide p and the spectrum precursor mass (Prec.$_{mass}$). Since the total mass of the predicted peptide by GA is expected to be equal to the precursor mass of the spectrum, the absolute value of Δmass is considered as a penalty to avoid getting undesirable short or long peptides. $Nterm$ is the number of sequential b-ion matches from N-terminus (left to right) and $Cterm$ is the number of sequential y-ion matches from C-terminus (right to left) of the theoretical spectrum t. These terms check the quality of match from both sides of the theoretical spectrum and reward the match. As normally those b-/y-ions in the middle part of the spectrum tend to have higher intensities, whereas those on the other two sides particularly N-terminus have lower intensities, without these two terms in the fitness function there will be a chance of ending up to a peptide sequence which is partially matched with the spectrum only from the middle. Therefore, with these two terms, a peptide which has a few b-/y-ions matched from two sides but not from middle, still has the chance to survive. In this case, the peptide gets a reasonable fitness value and has a chance to remain in the population, going through the evolutionary process for further improvement. $N_{unmatched}$ indicates the number of b-/y-ions in the theoretical spectrum t which are not a match against the spectrum s. The three terms are divided into the length of peptide.

It is worth mentioning that as b-y-ions of the whole sequence (not internal fragment ions) have higher chance to have their neutral losses presented in the MS/MS spectrum, there will be further step to produce the neutral losses such as H2O and NH3 for each match b-/y-ions. Therefore, if any b-/y-ion (of the whole sequence) is a match, its neutral losses such as H2O and NH3 will be produced to be match against the experimental spectrum. This is a bonus for the match b-/y-ions. Moreover, if any b-ion of the whole spectrum is a match, another neutral loss, CO, will be produced to be a match against the spectrum, which is another bonus for the match b-ions. Further bonus is calculating charge 2 of the precursor mass to check whether it is a match or not. However, as $N_{unmatched}$ is a penalty for unmatched peaks, the penalty only will be considered for the b-/y-ions of the whole sequence, neither internal fragment ions nor neutral losses. The reason is that it does not always guarantee that internal fragment ions and neutral ions are presented in the MS/MS spectrum. If they are presented, we will get advantages and consider a bonus for any match against them, otherwise we do not consider any penalty for their absence.

Apart from the fitness value produced by the fitness function above, the two terms $Nterm$ and $Cterm$ (without being divided into the peptide length), are also kept as additional fitness scores for each individual. These values are used later to apply a new crossover operator and are explained in the following section.

Nterm and Cterm Scores

The idea of calculating these two terms comes from the ion ladder of sequences and the CID fragmentation rules. As mentioned earlier in this section, the mass of any theoretical b-ion can be calculated based on Equation (5.3). Similarly theoretical y-ions can be calculated based on Equation (5.4). Moreover, the complementary theoretical b- and y- ions in each row of Table 2.1 have the mathematical relation presented in Equation (5.6).

$$b_j + y_{l-j} = PM(p) + 2 \tag{5.6}$$

$$PM(p) = \sum_{i=1}^{l} mass(a_i) + mass(H_2O) \tag{5.7}$$

$$\text{Prec.}_{mass} = \text{pepmass} \times \text{charge - charge} \times \text{mass(Proton)} \tag{5.8}$$

To calculate the b-ions in the theoretical spectrum, Equation (5.3) is used. Having the total mass of the peptide (parent mass in Equation (5.7)), the y-ions can be calculated either by Equation (5.6) or Equation (5.4). Therefore, for calculating the Nterm score, first all b-ions are calculated. Then, in Equation (5.6) instead of $PM(p)$ which the mass of the peptide, Prec.$_{mass}$ which is the precursor mass (from Equation (5.8), where pepmass is mass of the fragmented ion, charge is the precursor charge state and mass of Proton equals to 1.00727647 atomic mass units) is replaced, and y-ions are calculated. Let's call these y-ions as experimental y-ions (because we used mass of the spectrum). The experimental y-ions are compared with theoretical y-ions from Equation (5.4). Starting from y_1, if any two sequential experimental y-ions are equal to the corresponding theoretical y-ions, the Nterm score will be increase by one.

The Nterm score is able to check whether a matched b-ion is a random match or not. Similarly, Cterm is calculated by using Equation (5.4) and applying the similar process. The values of Nterm and Cterm scores do not necessarily indicate the exact amino acid matches in the sequence. For example a sequence with Nterm = 2 does not indicate that the first 2 amino acids from N-terminus are exact matches compared to the ground-truth peptide. The reason is that we are not aware of the ground-truth during the matching process. However, these two scores are able to check the quality of match from each side of the spectrum, and check whether it is a random match from one side or a potential correct match from two sides of the spectrum.

5.2.5 Nterm-Cterm Crossover

A new domain specific crossover is designed for this problem. The crossover mates two parents each having at least one exact match one b-/y-ion from N-terminus and C-terminus. The goal is to mate these two parents in the way that the new offspring would have exact b-/y-ion matches from both sides and possibly from the middle as well. Figure 5.4 illustrates the Nterm-Cterm crossover workflow.

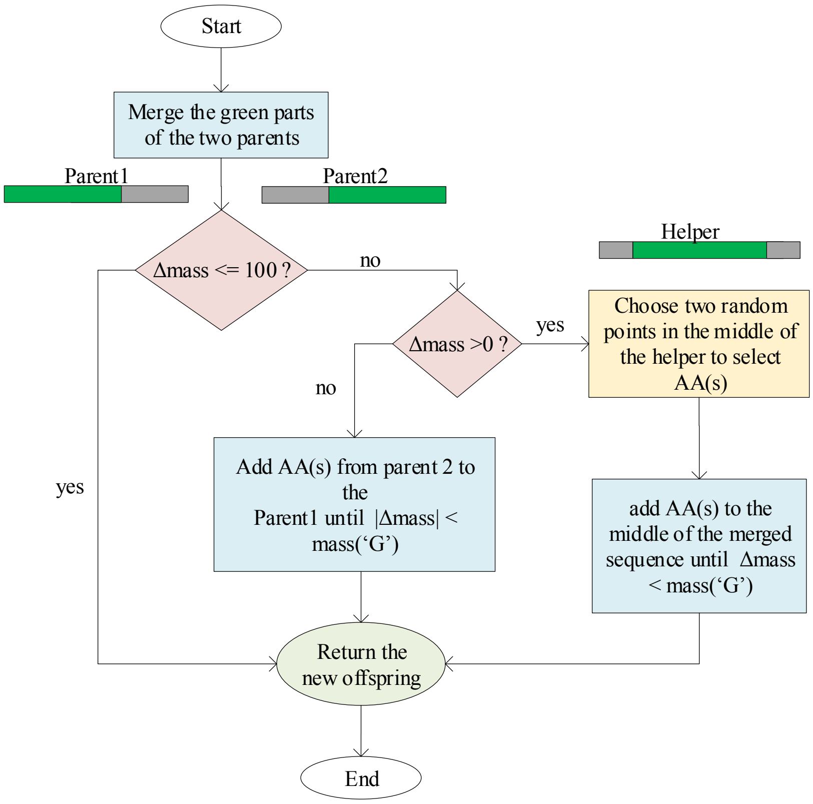

Start
Merge the green parts of the two parents
Parent1
Parent2
Helper
Δmass <= 100 ?
no
yes
Δmass >0 ?
yes
no
Choose two random points in the middle of the helper to select AA(s)
Add AA(s) from parent 2 to the Parent1 until |Δmass| < mass('G')
add AA(s) to the middle of the merged sequence until Δmass < mass('G')
Return the new offspring
End

Figure 5.4: The workflow of Nterm-Cterm crossover operator.

The input of the crossover is three GA individuals, two individuals as parents and one as a helper, and the output is a new offspring. At first, the exact match parts (the green parts) from both parents are concatenated. Here the $\Delta mass$ condition is more relaxed, allowing up to 100Da mass difference. If absolute value of $\Delta mass$ is more than 100, then the new concatenated sequence will be checked whether it needs to remove/add amino acids from/to the sequence. A negative $\Delta mass$ indicates that the sequence is long and needs removing a few amino acids from it and vice versa for a positive value. The reason is that based on Equation (5.7) a long sequence has more amino acids and possibly it could have a bigger parent mass compared to a shorter sequence with less number of amino acids. Since there might be some overlap

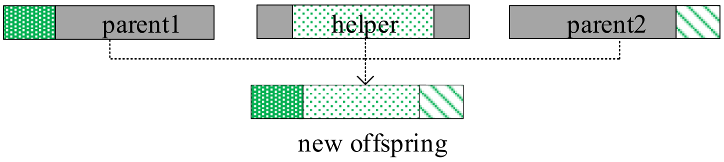


Figure 5.5: Schematic of how a new offspring with better fitness values is created.

between the green parts of the two parents, these $\Delta mass$ conditions help the operator avoid constructing a bad offspring having a big $\Delta mass$ penalty in its fitness value. Therefore, when $\Delta mass$ is negative, for removing the overlap the green part of the parent 1 is considered as the N-terminus of the new sequence and each time one amino acid from C-terminus (the most right) of the parent 2 is added to the new sequence until the $\Delta mass$ criterion is met.

If $\Delta mass$ is positive, it is required to add a few amino acids in the middle of the green parts of the two parents. Here instead of adding random amino acids, another individual as helper is used. The helper parent has a high fitness value which possibly could indicate having more matched peaks in the middle. So two crossover points are picked randomly from the middle of helper parent and the amino acids in between those two points are added to the middle of the new sequence one by one until the mass difference criterion is met. Figure 5.5 illustrates this process when the three parents are used to create a new offspring with better fitness values.

5.2.6 Flip-AA Mutation

The flip-AA mutation randomly picks one amino acid from the sequence and replaces it with one of the 19 amino acids ('I' and 'L' are considered identical). The mutation operator is not allowed to alter the last amino acid in the sequence as it is always supposed to be either 'R' or 'K' for a tryptic peptide sequence.

Table 5.1: The dictionary of conflict masses where mass of a single amino acid conflicts with di-peptides.

single AA	di-peptide	mass
W	DA, AD, EG, GE, VS, SV	186
R	VG, GV	156
Q	AG, GA	128
N	GG	114

Table 5.2: The set of peptide spectrum matches used in this chapter.

no. of PSMs	peptide length range	avg. length of peptides	Charge No.	Precursor mass range	fragment ion (Da)
120	7-12	9.5	2	<1150	0.5

5.2.7 Conflict-mass Mutation

There are situations where the mass of a single amino acid conflicts with the mass of two amino acids (di-peptides). For example, the mass('W') = mass ("DA") = 186. A dictionary of such conflict masses is provided and shown in Table 5.1. The conflict-mass mutation operator checks whether the sequence contains any amino acid in the conflict mass dictionary and if so randomly replaces the amino acid with any of the corresponding di-peptides.

5.3 Experiment Design

5.3.1 Dataset

From the comprehensive full factorial LC-MS/MS benchmark dataset [78], a set of 120 doubly charged peptide-spectrum matches (PSMs) with a minimum Mascot peptide identification score of 45, minimum peptide length of 7 amino acids and maximum length of 12 is selected. Based on Table 5.2, the average length of the peptides is 9.5. The spectra have a precursor of less than 1150 Da and the fragment ion of 0.5 is used as the value of tolerance τ. The so-called "ground-truth" is used to test the performance of *de*

Table 5.3: GA-Novo parameters

Parameter	Value	Parameter	Value
Initialisation Pool Size	1,000	Population Size	300
Size of Sub-pools	100	Generations, Runs	50, 30
Flip-AA Mutation Rate	0.1	Conflict-mass Mutation Rate	0.15
two-point Crossover Rate	0.35	Nterm-Cterm Crossover Rate	0.40
Elitism Rate	0.01	Selection	Tournament, 7

novo sequencing algorithms.

5.3.2 Parameters, Evaluation and Benchmark Algorithms

The parameters in Table 5.3 are used to setup the GA algorithm. GA-Novo is implemented in Python 3.6 and uses DEAP package [199]. To evaluate the accuracy of *de novo* sequencing results, the *de novo* peptide sequences constructed by the algorithm are compared with the real peptide sequences from the ground-truth dataset. The total recall and precision metrics at the amino acid level are calculated based on the following equations:

$$\text{precision} = \frac{\text{total number of matched amino acids}}{\text{total length of predicted peptide sequences}} \tag{5.9}$$

$$\text{recall} = \frac{\text{total number of matched amino acids}}{\text{total length of ground-truth peptide sequences}} \tag{5.10}$$

The performance of GA-Novo is compared with PEAKS [version 8.0] [77] and with PepyGen [185] which is a freely available *de novo* sequencing tool using GA. The metrics in both Equation (5.9) and Equation (5.10) measure the accuracy of the results at the amino acid level. The following metric is also used to evaluate the results of both algorithms at the peptide level.

$$\text{recall}_{peptide\ level} = \frac{\text{total number of fully correctly predicted peptide sequences}}{\text{total number of ground-truth peptides}} \tag{5.11}$$

5.4 Results and Discussions

This section presents a set of different experiments. The first experiment uses GA-Novo for *de novo* sequencing of 120 MS/MS spectra in the dataset and the results are compared with those of PEAKS and PepyGen. The rest of this section analyses the effectiveness of two main components used in GA-Novo namely tag-based initialisation method and the domain dependent Nterm-Cterm crossover. More analysis on the evolutionary process of GA-Novo is also presented in this section.

5.4.1 Performance Comparison Between GA-Novo, PEAKS and PepyGen

This section compares the overall performance of GA-Novo with PEAKS and PepyGen. All spectra in the dataset (Table 5.2) are used to assess the performance of both algorithms. These spectra contain noise and possibly incomplete ion ladders.

For each spectrum given to PEAKS, the top scored sequence is taken as the output of *de novo* sequencing. PEAKS was run with an error tolerance of 0.8 Da and tryptic digestion.

For GA-Novo and PepyGen, the experiments are repeated for 30 independent runs with 30 different random seeds for each input spectrum. For each spectrum in each run, the best fit sequence constructed by GAs are taken as the outputs of both GA methods.

To compare the results of GA-Novo in 30 runs with PEAKS, one sample statistical t-test with 95% confidence interval is used to compare the performance of two methods. Table 5.4 presents the results of *de novo* sequencing

Table 5.4: The results of sequencing 120 MS/MS spectra by GA-Novo and PEAKS.

Algorithm	Precision	Recall	recall$_{pep.\ level}$	avg. len. of partial matches	avg. len. of predicted sequences
GA-Novo	0.89 ± 0.03 (+)	0.88 ± 0.03 (+)	0.64 ± 0.06 (+)	8.4 ± 0.27 (+)	9.4 ± 0.1 (=)
PEAKS	0.85	0.84	0.56	8.06	9.43
PepyGen	0.42 ± 0.05	0.41 ± 0.05	0.14 ± 0.04	3.9 ± 0.2	9.1 ± 0.2

by these two methods. (+) in the table indicates the difference between the results of GA-Novo and PEAKS is considered to be statistically significant and (=) indicates not statistically significant.

From the results of Table 5.4, it can be seen that the results of GA-Novo in most cases are statistically significantly better than those of PEAKS and PepyGen. GA-Novo outperforms PEAKS by 4% increase in precision and 4% increase in recall at the amino acid level. Moreover, the accuracy of fully matched peptide sequences predicted by GA-Novo, at the peptide level, is 8% higher than PEAKS. The reason of having lower recall at the amino acid level compared to the precision at the amino acid level in the results of PEAKS and GA-Novo is that, they mainly construct either equal or slightly shorter peptide compared to the real peptide. Also, the results show that in overall GA-Novo is able to find more partially matched sequences compared to PEAKS, as the average length of partially matched sequences for GA-Novo is 8.4 which is statistically significantly better than the result of PEAKS.

As shown in Table 5.2 on Page 157, the average length of peptides in this dataset is 9.5, while sequences predicted by GA-Novo and PEAKS have the average length of about 9.4. No doubt that this value is close to the average length of the peptides in ground-truth as the goal of both algorithms is constructing full-length individuals. The sequences “AMVEVFLER” and “DAGTLLWLGK” are two examples of when PEAKS failed to predict the whole sequences, whereas GA-Novo could successfully construct the fully matched peptides. The sequences were predicted by PEAKS as "TT**VEVFLER**" and "W**GTLLWLGK**" while the first two amino acids in both sequences were predicted wrongly. More analysis on the results of PEAKS shows that it sometimes fails to predict the conflict masses from Table 5.1, whereas GA-Novo gets benefit of its domain dependent mutation operator, conflict-mass mutation, and avoids the mismatches.

Also from the results shown in Table 5.4, it can be seen that GA-Novo outperforms PepyGen by 45% and 47% increase in precision and recall at the amino acid level, respectively. Moreover, the accuracy of fully matched peptide sequences predicted by GA-Novo, at the peptide level, is 50% higher than PepyGen. The reason of low performance of PepyGen is due to its simple GA design which is not able to construct the correct peptide sequences. PepyGen does not apply the tag-based initialisation. Also, in the design of its fitness function the two terms Nterm and Cterm do not exist and this makes the algorithm to fail constructing the fully matched sequences. PepyGen, uses only simple mutation and crossover operators and this makes the algorithm not to be able to create fit individuals for the next generations.

Although the results show that GA-Novo is able to construct the full-length of sequences (9.4 relatively close to 9.5), GA-Novo also sometimes fails to construct the fully matched sequences (8.4). Comparing the difference between the average length of the ground-truth peptides, 9.5, and the average length of partial matches constructed by GA-Novo, 8.4, the result shows that in overall GA only fails to fully match either one or two amino acids. The reason of this mismatch is the incomplete conflict mass of di-peptides shown in Table 5.1. As mentioned previously in Table 5.1 where the mass of di-peptides conflicts with the mass of one single amino acid, there are other situations where the mass of two di-peptides conflict with each other. This table should be further extended to cover all other possible conflict masses.

The next sections analyse the effectiveness of initialisation method and the domain dependent crossover operator used in GA-Novo, and shows how these two components help GA in *de novo* sequencing.

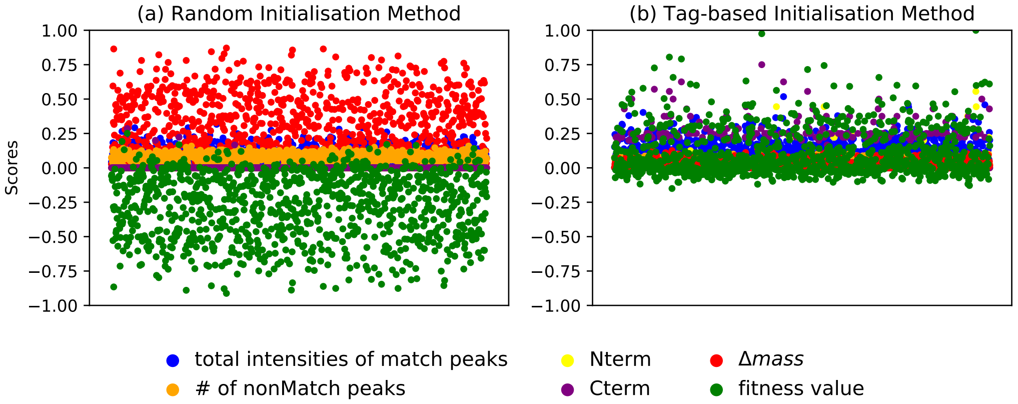


Figure 5.6: Plots of 1,000 individuals generated by random and tag-based initialisation.

5.4.2 Tag-based Initialisation vs. Random initialisation

Figure 5.6 illustrates two plots presenting the overall fitness value and the values of its five terms included in the fitness function (see Equation (5.5)). As the random initialisation method does not use any domain knowledge and randomly generates sequences between length 7 and 12, it can be seen from Figure 5.6 (a) that the fitness values of majority of population is below zero. The reason of such low fitnesses is that the random initialisation does not pay attention to $\Delta mass$, mass difference, which is a penalty in fitness function. Generating short or long peptide sequences results in a big $\Delta mass$ penalty. However, the tag-based initialisation plot (see Figure 5.6 (b)), shows how the $\Delta mass$ values are small in this method and the overall fitness values are bigger than the random initialisation method. Obviously, the $\Delta mass$ scores in this plot are very low and hardly can be seen compared to the those in the random method.

The results in Table 5.5 show that the best individual out of 1,000 individuals in a single run of random method is “YVMNEAR” with a fitness value of 0.25. In this table, each sequence is shown by its overall fitness value and five different terms from fitness function, including I, D and N which indicate the total intensities of matched peaks, $\Delta mass$ and the number of unmatched

Table 5.5: The best individual in a single run tag-based and random initialisation methods using the spectrum of "AAALAAADAR" peptide.

		Fitness scores					
	Sequence	Fitness	I	D	N	Nterm	Cterm
Ground-Truth	AAALAAADAR	2.1950	0.595	0.000003	0.0	0.8	0.8
Random Initial.	YVMNE**AR**	0.25	0.057	0.020099	0.071	0	0.28
Tag-based Initial.	RVA**AAA**W**R**	**1.14**	0.528	0.000027	0.0	0	0.625

Table 5.6: The statistics on three fitness scores in 30 different runs of tag-based and random initialisation methods using the spectrum of "AAALAAADAR" peptide.

	Fitness value				Nterm				Cterm			
	Min	Max	Avg.	Std.	Min	Max	Avg.	Std.	Min	Max	Avg.	Std.
Random Initial.	-0.97	0.32	-0.29	0.24	0	1.57	0.002	0.06	0	2.57	0.01	0.15
Tag-based Initial.	-0.15	1.07	0.1	0.17	0	4.83	0.04	0.35	0	5.93	0.45	0.98
Significance	(+)				(=)				(+)			

peaks, respectively. Nterm and Cterm scores are normalised here. Based on the tag-based method, the best individual is "RVAAAAWR" with fitness value of 1.14. Therefore, the fitness value of the best individual produced by the tag-based initialisation method is 4.7 times bigger than the one in random initialisation. As mentioned above the fitness value of the ground-truth is 2.19, the tag-based initialisation method could be a better start point for GA.

The statistics results in Table 5.6 show the significance of comparison between the results of the two methods. An unpaired statistical t-test with 95% confidence interval is used to compare the performance of two methods. This table presents the statistics on the overall fitness value, Nterm and Cterm scores. Please notice that Nterm and Cterm scores are not normalised here.

From the results of Table 5.6, it can be seen that the average fitness values and Cterm scores of tag-based initialisation method are statistically significantly better than random based method. However, Nterm scores are not statistically significantly better than random method, thanks to the tag extraction step which is sometimes not able to extract partially matched 3-letter tags from N-terminus of the spectrum due to the missing b-ions in this area. During peptide fragmentation, peptides may not fragment at some positions and leave no information, resulting in missing data. That is why the first two fragments b_1 and b_2 ions are seldom observed in the spectrum.

Overall based on the results in both Tables 5.5 and Table 5.6, tag-based method constructs better/fitter individuals compared to random initialisation, as tag-based method focuses on concatenating randomly 2, 3 or 4 tags. Then the method reduces the absolute mass differences between the constructed sequences and the spectrum by randomly inserting/removing random amino acids into the sequences. As known domain knowledge, each tryptic peptide ends in either ‘K’ or ‘R’, this heuristic has been applied randomly on the sequences constructed by this method as well. As a result, this method decreases the mass differences and increases the number of match ions, resulting in an increase in the total intensities of the match ions. Back to the best sequence produced by the tag-based initialisation, “RVAAAAWR” in Table 5.5, it is expected this sequence goes through the GA evolutionary process and after a few generations converts to the exact match.

5.4.3 Analysis the Effectiveness of Nterm-Cterm Crossover

This section presents two examples when Nterm-Cterm crossover is applied on different individuals and also shows the performance of this operator across 30 different runs.

Table 5.7 and Table 5.8 show how new Nterm-Cterm crossover can result in whole sequence exact match. By looking at Nterm and Cterm scores of the Nterm and Cterm parents in this table, it can be seen that these parents have quite big values that could indicate potential exact amino acid matches from each side. Considering the sequence of amino acids of these parents and knowing the ground-truth, it can be seen that the two parents have a few number of exact amino acid matches. However, concatenating the exact match amino

Table 5.7: An example of applying Nterm-Cterm crossover on two long partially matched parents that have matched amino acids overlap.

	Sequence	fitness	I	D	N	Nterm score	Cterm score
$Nterm_{parent}$	**AAALA**GGWR	0.79	0.21	0.031	0.05	4	2
$Cterm_{parent}$	NV**LAAADAR**	1.34	0.58	0.000002	0.02	0	7
$Helper_{parent}$	RG**LAAAD**VK	0.58	0.59	0.00003	0.01	0	0
$Offspring$	**AAALAAADAR**	2.19	0.59	0.000003	0.000	8	8

Table 5.8: An example of applying Nterm-Cterm crossover on two short partial match parents and a helper parent to fill the middle gap.

	Sequence	fitness	I	D	N	Nterm score	Cterm score
$Best\ Nterm_{parent}$	**AAA**PEPSEQK	0.1173	0.118	0.14	0.060	2	0
$Best\ Cterm_{parent}$	PEPSEQ**AR**	0.4477	0.237	0.014	0.025	0	2
$Best\ helper_{parent}$	RG**LAAAD**TK	0.2952	0.309	0.002	0.011	0	0
$Offspring$	**AAALAAADAR**	2.1950	0.595	0.000003	0.000	8	8

acids (shown in bold) results in a false sequence “AAALALAAADAR” which is not desired. As the proposed crossover method was explained previously, when two highly fit parents are concatenated with consideration of removing the overlap between them, an exact match as the new offspring is more likely to be obtained. As both parents have enough Nterm and Cterm match amino acids, the third parent, helper, is not used here.

The second example in Table 5.8 shows two parents with only a few number of exact match amino acids. As the concatenated sequence still does not meet the mass difference criterion, the third parent is used to fill the gap. It can be seen from the fitness scores of the helper parent that it is not necessary to have a high Nterm or Cterm score, as the helper parent is chosen based on its overall fitness value. Here also an exact match is obtained. However, it is worth mentioning that applying this operator does not always results in whole sequence exact match, but mainly there is an

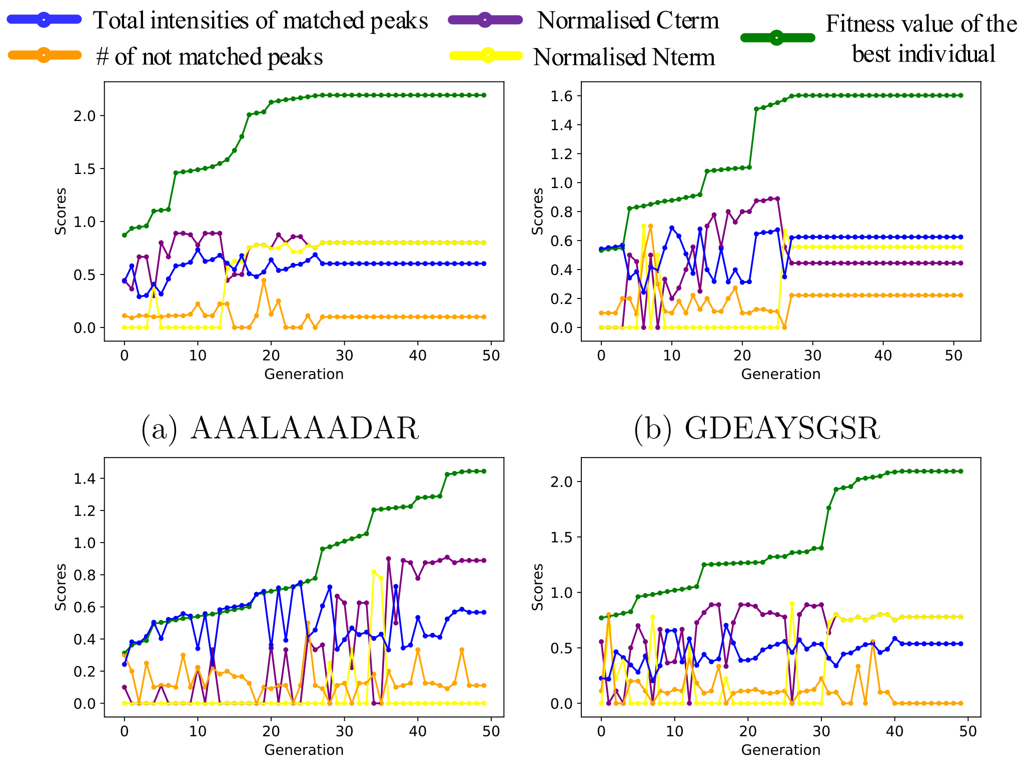
Table 5.9: Performance evaluation of Nterm-Cterm crossover operator using the spectrum of “AAALAAADAR” peptide in different scenarios.

	$\Delta f_{cx,N_{parent}}$	$\Delta f_{cx,C_{parent}}$	$\Delta f_{cx,H_{parent}}$
30 runs “Best” individuals	0.62	0.37	0.28
single run “Random” individuals	0.4	0.11	0.014

improvement in the fitness value of the new offspring. Therefore, clearly this operator shows that multi-parent crossover can be more effective than two parent crossover as suggested in literature [204, 205, 206].

Table 5.9 presents the overall performance of Nterm-Cterm crossover on a number of individuals produced by tag-based initialisation method. In the first row, the tag-based initialisation method is used in 30 independent runs, each run producing 1,000 individuals. In each run, out of 1,000 individuals three individuals with having the best Nterm score, Cterm score and fitness value are chosen to be Nterm parent, Cterm parent and helper parent, respectively. Then the Nterm-Cterm crossover is applied on the parents of each run and the average delta fitness values are calculated for all runs. It can be seen that in overall the fitness values of the offsprings improved by 62% compared to the Nterm parent, 37% to Cterm parent, and 28% compared to the helper parent.

Similarly, the second row of Table 5.9 presents the results of improvement in the fitness values of new offsprings produced by Nterm-Cterm crossover in a single run, but randomly choosing 30 individuals as parents which are not necessarily the best scored parents. The results show that in this case also in average, there is 4% improvement in the fitness score of the new offspring compared to its Nterm parent, 11% compared to Cterm parents and 1.4% compared to the helper parent. One reason of not having a significant improvement in this results is that the parents are not filtered. That is why in design of the GA algorithm, presented in Figure 5.1, the individuals in two Nterm and Cterm pools must have at least an Nterm or Cterm score of one.



(a) AAALAAADAR

(b) GDEAYSGSR

(c) TFGMEGLFR

(d) ANAYGHGLER

Figure 5.7: The evolutionary process of GA-Novo converging to the whole sequence exact matches using 4 different MS/MS spectra.

5.4.4 Evolutionary Process of GA-Novo

The plots in Figure 5.7 illustrate the evolutionary process of GA-Novo using 4 real MS/MS spectra from the dataset. These examples demonstrate how GA during the evolutionary process constructs the full sequences which turned out to be the exact matches. Considering the fitness scores during the evolutionary process, it can be seen that the fitness plots in Figure 5.7 (a) and Figure 5.7 (b) are converged to the exact matches in the middle of the evolutionary process (less than 30 generations), whereas the other two plots required more generations to find the exact matches due to the potential missing ions or more noise (more challenging spectra).

5.5 Chapter Summary

The goal of this chapter was developing an effective *de novo* sequencing algorithm that constructs full-length sequences. The goal has been successfully achieved by developing an effective GA algorithm that constructs the peptide sequences that match the input MS/MS spectra.

The key developments presented in this chapter are a new domain dependent fitness function, a new initialisation method and two new genetic operators that were particularly designed for the GA algorithm. The GA fitness function was able to capture main spectral features, guiding GA to produce the fully matched peptides. The initialisation method was an excellent start point to accelerate the evolutionary process. The tag-based initialisation method helped GA start with better/fitter initial population, improving its convergence, and providing high quality individuals for the GA components. The genetic operators helped GA maintain the diversity in the population and gradually convert partial matches to fully matched sequences. The results showed that GA-Novo achieved higher number of fully matched sequences compared to PEAKS. GA-Novo outperformed PEAKS by 4% higher precision and 4% higher recall at the amino acid level and 8% higher recall at the peptide level. Also GA-Novo outperformed PepyGen, a GA-based *de novo* sequencing tool with a significant margin of 45% at the peptide level.

Clearly GA showed promising application in searching for finding the most likely full-length sequence in *de novo* sequencing. Although GA-Novo outperformed its other counterparts at both amino acid level and peptide level, still it ended up with some false positives. A possible reason could be its fitness function which is not discriminative enough. The next chapter develops an effective method which addresses this problem by developing a new PSM scoring function which can reduce the rate of false positives in the results of *de novo* sequencing. Moreover, in the final chapter of this thesis in Section 7.3, future developments of GA-Novo is suggested.

Chapter 6

GP for Re-scoring and Re-ranking the Peptide-Spectrum Matches

6.1 Introduction

De novo peptide sequencing algorithms have been widely used in proteomics to analyse tandem mass spectra (MS/MS) and assign them to peptides, but quality-control methods to evaluate the confidence of *de novo* peptide sequencing are lagging behind. We do not want to assign a spectrum to a peptide which is not presented in the biological sample, as incorrect peptide assignments result in incorrect protein identifications. The search scores in the current *de novo* peptide sequencing algorithms do not always guarantee to find the true (correct) matches from many false matches. Therefore, it is essential to apply a post-processing step as a PSM validation phase on the results of *de novo* sequencing in order to improve peptide identification accuracy and sensitivity.

A fundamental part of a quality control method is the scoring function used to evaluate the quality of PSMs. PSM scoring is similar to the document-query scoring in information retrieval, where the search engine uses a ranking algorithm to determine the relevance of each document (web page) to the input query. Accurate ranking provides users relevant results on the top of the search results [56].

The PSM ranking function is expected to perform two tasks: (1) producing appropriate scores to each PSM, (2) giving the highest score to the correct PSM which results in distinguishing the correct match from other incorrect ones. The first task looks like a regression task and the second one is somehow similar to a classification problem, and the important point is that the correct PSM should get the highest score amongst the other candidates for the same spectrum. Therefore, the ranking function can be treated as a classification or a regression method.

Since machine learning algorithms often solve either a classification or a regression problem, not two problems together, it is worth investigating the capability of GP in this regard. To the best of our knowledge, this would be the first GP approach that solves a regression and a classification problem simultaneously.

6.1.1 Chapter Goals

The main goal of this chapter is to develop a novel GP approach, named GP-PostNovo, that solves a regression task and a classification task simultaneously, aiming to generate an effective GP-based PSM scoring function to re-score and re-rank the *de novo* peptide sequence predictions. It is expected that the new GP-based PSM scoring function increases the rate of the full-length correct peptides predicted by any individual *de novo* peptide sequencing tool. The following objectives are specifically investigated:

1. Design an appropriate strategy which enables GP to learn from two different sources of training sets, one for regression and the other for classification.

2. Design an appropriate terminal set composed of a diverse set of effective features that can capture different aspects of the quality of a PSM.

3. Design an appropriate fitness function which measures the performance of each GP individual in terms of appropriate regression and classification metrics, leading GP towards building a powerful discriminative PSM scoring function in order to distinguish between the correct PSMs and incorrect ones.

4. Evaluate the effectiveness of the GP evolved PSM scoring function to post-process the results of *de novo* sequencing tools in terms of the rate of full-length correct peptides and compare the performance of GP-PostNovo with other GP-based and Non-GP methods.

6.1.2 Chapter Organisation

The rest of the chapter is organised as follows. Section 6.2 describes the proposed GP method. Section 6.3 presents the design of the experiments. The results and discussions are presented in Section 6.4. Finally, the summary of the chapter is presented in Section 6.5.

6.2 The Proposed GP-PostNovo Method

6.2.1 Overview of the Method

Figure 6.1 illustrates the flowchart of the proposed GP-PostNovo method designed for re-scoring and re-ranking the results of *de novo* sequencing methods. The process is composed of three phases: (1) data preparation, which involves the process of creating the training and test data; (2) learning phase, where GP builds the PSM scorning function using the training set, and (3) performance evaluation phase, where the new GP evolved PSM scoring function is used to post-process the results of *de novo* peptide sequencing by a *de novo* sequencing tool, aiming at giving the greatest score to the correct candidate peptide (true match) in the candidate peptide list produced by the

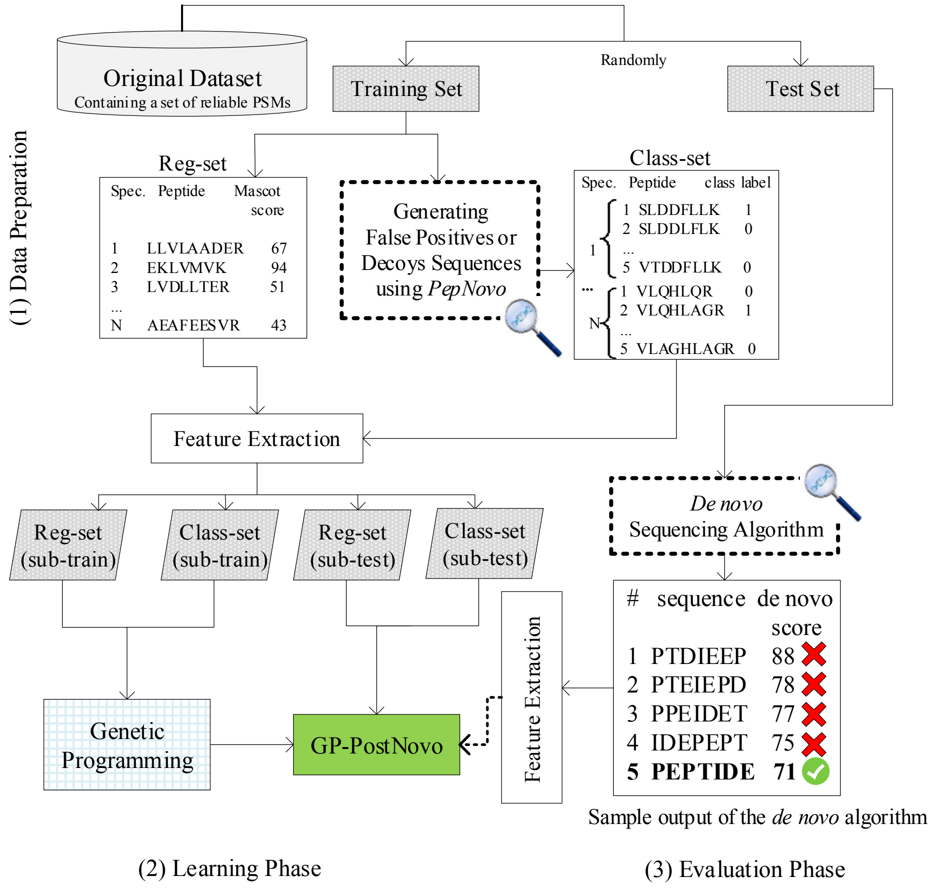


Figure 6.1: The workflow of the proposed GP-PostNovo method.

de novo tool for each MS/MS spectrum in the test set. More details about each phase is explained as follows.

(1) Data Preparation

The first step of the flowchart, data preparation, starts with selecting a set of MS/MS spectra with known identification from the original database [78]. The spectra are randomly selected and split into two sets: a training set, which is used by GP to learn and build a scoring model, and a test set, which is the unseen data and is used to evaluate the effectiveness of the GP model.

The training set is created in two forms: *reg-set* and *class-set*. In *reg-set* for each spectrum, only its correct peptide is considered. Therefore having N spectra in the training set, *reg-set* has N peptides. It should be noticed that N peptides mean N PSMs because peptides are matched with the spectra to create a PSM, and each PSM is one instance in the dataset. Therefore *reg-set*, which is used by GP for symbolic regression, has N instances. With feature extraction, a set of features (from Table 6.1) which measure the goodness of match between the spectrum and the candidate peptide, is extracted from each PSM. Therefore, each instance in *reg-set* is represented by a feature vector (as input/dependent variables of regression) and a Mascot score (as output/target variable of regression) indicating the degree of reliability of the match.

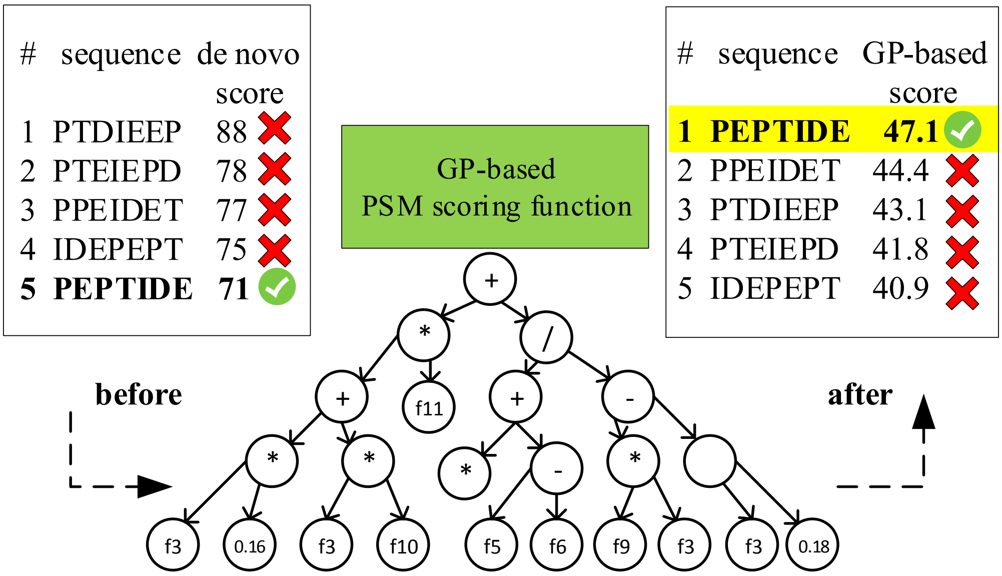
In *class-set* for each spectrum, a list of candidate peptides including one correct peptide and four incorrect peptides is considered. As *class-set* is used by GP for classification, the correct peptides are positive instances and incorrect peptides are served as negative instances. Therefore having N spectra in the training set, *class-set* has $N \times 5$ PSMs (instances). To generate the negative instances (incorrect peptides), all N spectra in the training set are given to PepNovo [44], a freely available *de novo* sequencing tool, to perform *de novo* sequencing. As the ground-truth is already available, which means the corresponding correct peptide for each spectrum is known (positive instances), among the list of peptides generated by PepNovo, for each spectrum four full-length false matches are taken and considered as negative instances (incorrect peptides). Therefore, as it can be seen in Figure 6.1, *class-set* contains N groups of PSMs where each group belongs to one spectrum and is composed of five peptides including one correct peptide that has the class label of 1 and four incorrect peptides that have class label of 0. The instances in *class-set* are sent to features extraction step to extract the same set of features that previously were extracted from the instances in *reg-set*.

(2) Learning phase

The design of learning phase allows GP to evolve computer programs in classification and regression tasks. The GP fitness function is composed of the summation of the regression error and the classification error. The former is used to evaluate the performance of the GP program using *reg-set* in a regression task in terms of relative sum of squared error (RSS), and the latter evaluates the performance of the same GP program in terms of the error rate of the classification, *misRank* rate, using *class-set* in a classification task. During the evolutionary process, each training set is split into two sets of sub-train and sub-test sets with each set having 70% and 30% of total instances in each set. The sub-train sets are used to train GP and sub-test sets are used to evaluate the GP model. More details about the GP evolutionary process will be described in Section 6.2.4.

(3) Evaluation phase

The third phase of the flowchart is evaluating the effectiveness of the GP evolved PSM scoring function using the test set. All MS/MS spectra in the test set are given to a *de novo* sequencing tool which is considered to improve its *misRank* rate by applying the new PSM scoring function generated by GP. After performing the *de novo* sequencing, for each spectrum, a set of top five peptides is taken as the results of identifications. Therefore, having M test spectra, the test set contains a total number of M × 5 instances. The same set of features similar to *reg-set* and *class-set* is extracted from the instances in the test set as well. As the ground-truth is already available from the original dataset, the sequences produced by the *de novo* tool are evaluated to measure the identification rate of full-length peptides sequencing. For each spectrum, the top-ranked PSM is taking as the identification result and is compared with the ground-truth peptide to check whether it is a full-length correct match or not. If the predicted sequence by the *de novo* sequencing method exactly matches the ground-truth peptide, the value of total number



#	sequence	de novo score	
1	PTDIEEP	88	✗
2	PTEIEPD	78	✗
3	PPEIDET	77	✗
4	IDEPEPT	75	✗
5	**PEPTIDE**	**71**	✓

#	sequence	GP-based score	
1	**PEPTIDE**	**47.1**	✓
2	PPEIDET	44.4	✗
3	PTDIEEP	43.1	✗
4	PTEIEPD	41.8	✗
5	IDEPEPT	40.9	✗

Figure 6.2: Correctly re-ranking the PSMs using GP.

of true matches increases by one. However, there might be some cases that although the true match (correct PSM) exists among the other four false matches (incorrect PSMs), the *de novo* sequencing method does not give the greatest match score (first rank) to it (see the box showing the sample output of the *de novo* algorithm in Figure 6.1). Therefore, the true match is missed and instead the *de novo* sequencing method reports a false match as its identification result. In this situation, it is expected that by applying the new GP evolved PSM scoring function on the results of the *de novo* sequencing tool, those missed true matches can be identified, re-scored and re-ranked by GP to get the top rank in each group. This process is illustrated in Figure 6.2. It is worth mentioning that the GP evolved PSM scoring function as a post-processing method is only expected to find the true matches that already exist in the candidate lists of the spectra, and give them the top rank scores. If the correct peptides do not exist among the top five candidates, there is nothing that the GP scoring function can do with those spectra.

Table 6.1: Features used to represent the PSMs.

Feature name	Description
f_1 deltaMass	Mass difference between the experimental spectrum s and the peptide sequence p
f_2 $\mathrm{I}_{matched}$	Sum of intensities of those peaks in s which are matched with theoretical spectrum of p
f_3 $\mathrm{N}_{matched}$	# of matched peaks in theoretical spectrum of p
f_4 $\mathrm{N}_{not\text{-}matched}$	# of not matched peaks in theoretical spectrum of p
f_5 Nterm	# of matched b-/y-ions from N-terminus (left to right) of p
f_6 Cterm	# of matched b-/y-ions from C-terminus (right to left) of p
f_7 GA-Novo	Linear combination of different match sub scores
f_8 Cos	Fixed length Normalised Dot product between the two vectorised spectra of s and t (theoretical spectrum of p)
f_9 Euc	Fixed length normalised Euclidean distance between s and t
f_{10} Hamming	Hamming distance between s and t
f_{11} SeqFix	Fixed length SEQUEST-like score between s and t
f_{12} SeqVar	Variable length SEQUEST-like score between s and t

6.2.2 Feature Extraction

Table 6.1 shows a set of 12 features extracted from each PSM. These features measure the goodness of match between the experimental spectrum s and the peptide p from different perspectives. For being able to match the peptide sequence p against the experimental spectrum s, a theoretical spectrum t based on the known CID fragmentation rules of doubly charged peptides [190] is constructed from the peptide sequence. More details about how to construct t can be found in Section 5.2.4 (see Page 150).

As can be seen from the list of features in Table 6.1, the first seven features in this table are those which have been previously used in designing the fitness function of GA-Novo in Chapter 5. The purpose of using these features is to see how GP can find the hidden relationship between these features.

Features $\{f_8, f_9, f_{10}, f_{11}, f_{12}\}$ vectorise the experimental spectrum s and the theoretical spectrum t into two binned vectors with same length and use distance based measures to calculate the similarity between the two vectors.

This similarity indicates the goodness of the match between the spectrum s and the peptide p.

The vectors in features $\{f_8, f_9, f_{10}, f_{11}\}$ have fix length of 4,000 bins. This number comes from dividing the largest possible precursor mass of the spectra from the dataset used in this chapter into the fragment ion tolerance used to generate the spectra, i.e. 2000 divided by 0.5. However, the feature f_{12} has a variable length which is determined based on dividing the precursor mass of each spectrum under investigation into the fragment ion tolerance. For the fix length features, the process of vectorising the experimental and the theoretical spectra starts with splitting each spectrum into 4,000 bins. For the experimental spectrum, the value of each bin equals to the total intensities of those peaks which fall into the corresponding bin in the vector. In the case of the theoretical spectrum, the value of each bin is ether 0 or 100 as the intensity of each peak in the theoretical spectrum is 100%. If the theoretical spectrum contains a peak within the range of the bin, the value of the bin equals to 100, otherwise 0.

The *Cos* feature, f_8, calculates the normalised dot product between the experimental vector x and the theoretical vector y using Equation (6.1).

$$cos\ \theta = \frac{x.y}{||x|| \times ||y||} \tag{6.1}$$

where $||x||$ is the absolute value (magnitude) of vector x and $0 \leq cos\ \theta \leq 1$. The ideal value of feature *cos* is 1, which indicates the two vectors are identical and that all peaks in s are matched against those in t.

The *Euc* feature, f_9, measures the euclidean distance between the two vectored spectra using Equation (6.2).

$$euc\ (x, y) = \frac{\sqrt{\Sigma(x_i - y_i)^2}}{||x|| \times ||y||} \tag{6.2}$$

The *Hamming* feature, f_{10}, first converts the two experimental and theoretical vectors into binary vectors and then calculates the normalised hamming distance between the two vectored spectra.

As both f_9 and f_{10} are used to measure the dissimilarity between the two entities, the ideal value for both of them is 0.

SeqFix and *SeqVar*, f_{11} and f_{12} both use Equation (6.1) to calculate the similarity between the two vectors. However, the difference between f_{11} and f_{12} in terms of their functionality with f_8 is that both f_{11} and f_{12} apply a preprocessing step before vectorisation the experimental spectra. The preprocessing is inspired of the preprocessing method in SEQUEST [92]. The preprocessing step removes the precursor mass and keeps top 200 most intense peaks in the spectrum s. Then the whole spectrum is divided into 10 windows with each window having peaks with normalised intensities to the maximum value of 50. After preprocessing the experimental spectrum s, the rest of process is similar to vectorisation explained for f_8 where the spectra are split into n bins.

6.2.3 GP Program Representation

In the proposed GP method, an individual with its tree-based structure represents a scoring function that assigns a real number to a PSM as its match score when performing either classification or regression. The GP method is designed with a terminal set including a set of features from Table 6.1 along with random floating point constants, and a function set containing four arithmetic operators of $\{+, -, \times, /(\text{protected})\}$. The protected division returns the value 1 if a division by zero has taken place. Table 6.2 shows the GP parameters used to run the experiments. The proposed method is implemented in Python 3.6, using DEAP package [199].

Table 6.2: Genetic programming parameters

Parameter	Value
Function Set	$\{+, -, \times, /(\text{protected})\}$
Terminal Set	{Features from dataset, random Constants}
Initial Population	Ramped-half-and-half
Population Size	600
Maximum Generations	100
Mutation Rate	19%
Elitism Size	1%
Crossover Rate	80%
Selection	Tournament, Size = 5

6.2.4 An Effective Fitness Function for Re-scoring and Re-ranking the PSMs

The fitness function to evaluate the goodness of GP individuals is composed of two parts.

$$fitness_{(ind_i)} = RSS_{(ind_i)} + misRank_{(ind_i)} \tag{6.3}$$

where $RSS_{(ind_i)}$, relative sum of squared error, is the fitness value of the i-th GP individual on *reg-set* and it shows how the GP individual does not fully represent the actual relationship between the target variable and the dependent variables in the dataset. $misRank_{(ind_i)}$ presents the classification error rate of the same GP individual on *class-set* and it measures the incapacity of the GP individual to give the greatest score (first rank) to the correct peptide. So $misRank_{(ind_i)}$ indicates the rate of the correct PSMs that did not get the first rank. As we try to minimise the classification and regression error rates during the evolutionary process, for a GP individual the best fitness value based on Equation (6.3) would be close to 0.

The first term of this equation is calculated using the following equation:

$$RSS_{(ind_i)} = \frac{\Sigma_{j=1}^{N}(\hat{Y}_j - Y_j)^2}{\Sigma_{j=1}^{N}(\overline{Y} - Y_j)^2} \tag{6.4}$$

where $\hat{Y}_j$ is the output of the i-th GP individual for instance *j*, and Y_j is the actual output value of instance *j*. $\overline{Y}$ is the mean of the target values, and N is the total number of MS/MS spectra in *reg-set*. A GP individual with the best performance has RSS close to 0.

$misRank_{(ind_i)}$ for each GP individual is calculated based on Equation (6.5). Having N MS/MS spectra in *class-set* indicates having N groups of PSMs where each group contains one correct PSM and four incorrect PSMs.

$$misRank_{(ind_i)} = 1 - \frac{\Sigma_{j=1}^{N} hit(j)}{N} \tag{6.5}$$

where *hit(j)* is the total number of first ranked correct PSMs in each group. As already mentioned in Equation (6.3) that GP tries to minimise the fitness function, the average value of *hit(j)* is subtracted from 1 so that the best value of $misRank_{(ind_i)}$ would be 0, which reflects that all spectra of *class-set* were correctly identified (first rank). The following equation explains how *hit(j)* is calculated.

$$hit(j) = \begin{cases} 1, & \text{if } \textit{score}(\text{correct PSM}) > \textit{score} \text{ (incorrect PSMs)} \\ 0, & \text{otherwise} \end{cases} \tag{6.6}$$

So when the i-th GP individual is used to classify the PSMs in each group, if the output of the GP individual (means *score* function) using the correct PSM in group j is greater than the output of each four incorrect PSMs in the same group, the hit value of group j ($hit(j)$) gets a value of 1, otherwise 0.

The pseudo-code presented in Algorithm 2 shows the learning phase process in the flowchart of Figure 6.1, where GP is building the scoring model using two training sets: *reg-set* and *class-set*. The inputs to this algorithm are the instances in *reg-set* and *class-set* and the output is the best GP

Algorithm 2: The pseudo-code of the learning phase in the proposed GP method.

```
Input :
  • reg-set (sub-train): a set of PSMs as instances which have $n$ features and
    Mascot scores as target values.
  • class-set (sub-train): a set of PSMs as instances which have $n$ features and
    class labels of either 1 or 0 indicating correct or incorrect PSM,
    respectively.
Output: The best evolved GP $ind^*$ in terms of the fitness value on the
        training set.
Initialisation: Randomly initialise each individual to create the
  population $P$
$gen \leftarrow 0$
while $gen \leq maxGen$ do
    for $i = 1$ to $Popsize$ do
        Evaluation:
          • Evaluate the performance of $ind_i$ on the
            reg-set in terms of RSS
          • Evaluate the performance of $ind_i$ on the
            class-set in terms of $misRank_{(ind_i)}$
          • Calculate the overal fitness of $ind_i$ based on:
            $fitness_{(ind_i)} = RSS_{(ind_i)} + misRank_{(ind_i)}$
    end
    for $i = 1$ to $Popsize$ do
        if $fitness_i < fitness_{ind^*}$ then
            $ind^* \leftarrow ind_i$
        end
    end
    Evolution: Generate new population to the size of $Popsize$ by
      • Reproduce the most fit individual, $ind^*$ (elitism)
      • Apply the crossover and mutation operators
    $gen \leftarrow gen + 1$
end
return $ind^*$
```

evolved PSM scoring function in terms of the overall fitness (Equation (6.3)) on the training sets.

The GP evolutionary process starts with the creation of the initial population with size of *Popsize*. During the evolutionary process by using the instances from *reg-set*, each GP individual as the solution to a regression problem learns to assign a real number as match score to a PSM as accurate as possible. Meanwhile, the same GP individual tries to solve a classification problem by using the instances from *class-set* to learn distinguishing between the correct PSM and incorrect ones, aiming at giving the greatest match score to the correct PSM compared to its opponent incorrect PSMs corresponding to the same spectrum.

The overall fitness value for each GP individual is calculated and the best GP individual in each generation is transferred to the next generation using the elitism operator. Crossover and mutation as genetic operators are applied on the individuals from the current generation to populate the subsequent generation. The evolutionary process continues until the stopping criterion, reaching to *maxGen*, is met. The best GP individual based on the total fitness value on the training sets is returned as the output of the learning phase. Later the best GP individual (the best GP evolved PSM scoring function) is used to re-score and re-rank the PSMs in the test set. More details about the learning phase and performance evaluation phase are explained in Section 6.3.3.

6.3 Experiment Design

6.3.1 Dataset

Table 6.3 presents the datasets used to run the experiments in this chapter. Each dataset contains different number of MS/MS spectra which are selected from the original benchmark dataset [78]. The training set is composed of two sets namely *reg-set* and class-set. Reg-set only contains target

Table 6.3: The MS/MS spectra used in this study.

dataset		# of spectra	# of target PSMs	# of decoy PSMs
Training set	*reg-set*	3,515	3,515	-
	class-set	3,515	3,515	3,515 × 4
Test set		120	120	

PSMs, whereas *class-set* is composed of the same target PSMs in *reg-set* along with a set of four decoy PSMs for each target PSM. The test set contains the same MS/MS spectra used in Table 5.2 (see Page 157). Using the same set of spectra allows us to investigate the impact of the proposed method on improving the peptide identification with PEAKS and GA-Novo which previously were used in Chapter 5. Table 6.3 shows more details about these sets.

6.3.2 Benchmark Algorithms

As GP is used to build a scoring function which gives a real number to a PSM, it is compared with Random Forest (RF), Support Vector Regression (SVR) and Support Vector Machines (SVMs) which are benchmark algorithms in solving regression and classification problems. Similar to the proposed GP method, the other methods use the same training sets to learn the scoring functions.

As the proposed GP method solves two tasks of regression and classification at the same time, another GP-based method, named GP-PSM, which generates the PSM scoring functions by only solving a regression problem is also developed. To train GP-PSM, *reg-set* is used and the evolved GP programs are evaluated in terms of *RSS* (GP-PSM fitness function). This method is used for comparing with the proposed GP method in order to evaluate the effectiveness of the proposed simultaneous regression and classification strategy. The GP parameters used in this model are the same as

GP-PostNovo from Table 6.2.

To evaluate the effectiveness of the scoring functions built by the methods above, the spectra in test set are submitted to two *de novo* sequencing algorithms: PEAKS [9] and GA-Novo (the GA-based *de novo* sequencing method proposed in Chapter 5). Then the effectiveness of each PSM scoring function as a post-processing method is evaluated on the output of each *de novo* sequencing method in terms of *misRank* before and after the post-processing. Equation (6.5) presents *misRank* which is the ratio of total number of correct PSMs that did not get the first rank to the total number of MS/MS spectra.

6.3.3 Experiments

Experiment I: Learning PSM Scoring Functions

This experiment performs the second step, the learning phase, as explained in the design of the proposed method and illustrated in Figure 6.1. The two training datasets (*reg-set* and *class-set* in Table 6.3) are divided into two sub-train and sub-test sets with each set having 70% and 30% of the PSMs, respectively. The experiment is set up to build the PSM scoring functions by GP according to the pseudo-code in Algorithm 2 which explains how GP learns the scoring function using the two training sets. GP uses the sub-train sets of *reg-set* and *class-set* to build the model and evaluates the model using the sub-test sets. The results of GP on both sub-train and sub-test sets are measured based on Equation (6.3).

To compare the GP model with other non-GP benchmark algorithms, RF, SVR and SVM are used to build the PSM scoring models as well. Unlike each GP computer program (GP tree) which is able to perform either regression or classification task at each single evolution, RF, SVR and SVM are only able to perform one task during the model training. Therefore, each method is trained separately using the training set which is appropriate for the corresponding task. So RF, SVR and GP-PSM use *reg-set* to solve a regression problem and they are evaluated on *reg-set* in terms of RSS (Equation (6.4) which is the regression error metric. Moreover, *class-set* is used by RF and SVM to build the PSM scoring models and they both are evaluated in terms of the classification error rate, $misRank$, on *class-set*.

Later in Experiment II, all models built by the five algorithms are used to re-score and re-rank the results of the *de novo* sequencing algorithms using the spectra in the test set, aiming at decreasing the classification error rate, $misRank$.

Experiment II: Evaluating the Effectiveness of GP, RF and SVR

This experiment performs the evaluation phase which is the last phase of the flowchart in Figure 6.1. The experiment evaluates the effectiveness of all scoring functions obtained from the previous experiment using GP, RF, SVR and SVM in terms of improving the *misRank* rate of the results of *de novo* sequencing by two *de novo* sequencing methods, PEAKS and GA-Novo. The experiment starts by giving the MS/MS spectra from the test set in Table 6.3 to PEAKS and GA-Novo separately to perform *de novo* sequencing. From the output of each *de novo* algorithm, for each spectrum, a group of top 5 peptides is taken as the result of identification. The top-rank peptide in each group is compared with the ground-truth to evaluate the *misRank* rate of each *de novo* algorithm. Then the results of *de novo* sequencing by each algorithm are given as the input to the scoring functions generated by the five scoring models obtained in Experiment I for re-scoring and re-ranking the PSMs.

All scoring models from Experiment I are used to evaluate their effectiveness for post-processing the results of PEAKS and GA-Novo in terms of improving the *misRank* rate of *de novo* peptide sequencing. As already the corresponding peptides of the MS/MS spectra is known, the *misRank* rate before and after applying the post-processing can be measured.

Table 6.4: Fitness values of GP-PostNovo.

	sub-train	sub-test
30 runs	0.69 ± 0.03	0.71 ± 0.02
best run	0.65	0.69

Table 6.5: The results of RF and SVR in terms of RSS on *reg-set*.

Method	sub-train	sub-test
RF-regression	0.15	0.51
SVR	0.75	0.87
GP-PSM	0.48	0.50

6.4 Results and Discussions

6.4.1 Results of Experiment I

Table 6.4 presents the fitness values (i.e. Equation (6.3)) of GP in the learning phase of the proposed method (see Figure 6.1) using the training set including *reg-set* and class-set. The GP experiments are repeated 30 times using 30 different random seeds and the results of the average (± standard deviation) of the 30 runs are provided in this table. Moreover, the results of the best GP individual on sub-train sets based on Equation (6.3) is also presented in this table. As other methods (i.e. RF, SVM, SVR, and GP-PSM) are only able to perform either classification or regression, their results cannot be presented in this table. Instead, RF-regression, SVR and GP-PSM which solve regression problems and RF-classification along with SVM that are both performing classification are compared with each other.

Table 6.5 presents the results of RF-regression, SVR and GP-PSM on *reg-set* in terms of RSS. From the results of this table, it can be seen that RF-regression has better performance on sub-train but has low generalisation on sub-test of *reg-set* compared to GP-PSM. One possible reason of the

Table 6.6: The results of RF and SVM in terms of *misRank* on *class-set*.

Method	sub-train	sub-test
RF-classification	**0.21**	**0.79**
SVM	0.84	0.90

deterioration of the performance of RF and SVR on test sets could be overfitting in these methods. The spacial design of the GP approach helps GP to avoid this issue, resulting a reasonable performance on train sets of both reg-set and class-set and the best performance on test sets.

Similarly, Table 6.6 presents the results of RF-classification and SVM on *class-set* in terms of *misRank*. From the results, it can be seen that RF-classification has better *misRank* on both sub-train and sub-test of *class-set* than SVM and this means that RF-classification outperforms SVM in classification of PSMs. But still both methods have low generalisation possibly due to the overfitting.

As previously mentioned that unlike GP-PostNovo, other algorithms used here cannot learn simultaneously from different training sources, here we cannot compare the performance of RF, SVM and SVR with GP-PostNovo based on sub-train sets. However, the performance of each method can be evaluated on each sub-test set individually. Therefore, we further evaluate these PSM scoring functions (models) by applying them on the sub-test sets of *class-set* and *reg-set* to evaluate the methods based on Equation (6.3).

Table 6.7 presents the results of each PSM scoring model on sub-test sets of *reg-set* and *class-set* based on *RSS* and *misRank*, respectively and shows the overall fitness values of each method based on Equation (6.3). On sub-test of *reg-set* it can be seen that the results of RF-classification and SVM are far worse than other three methods. The reason is that these two methods are trained to build a classification model which outputs either 0 or 1, not a float point figure. Therefore, these two methods have very low performance on a sub-test set designed for a regression problem. Overall, it can be seen

Table 6.7: Evaluating the PSM scoring functions on *sub-test* sets of *reg-set* and *class-set*.

Method	sub-test (*reg-set*) *RSS*	sub-test (*class-set*) *misRank*	total *fitness*
RF-regression	0.51	0.30	0.81
RF-classification	8.84	0.79	9.63
SVR	0.87	0.38	1.25
SVM	9.08	0.90	9.98
GP-PSM	0.50	0.27	0.79
GP-PostNovo	**0.49**	**0.20**	**0.69**

that GP-PostNovo outperforms other methods on each sub-test, achieving the best fitness value in terms of Equation (6.3). The second and third best methods are GP-PSM and RF-regression, respectively.

In the next experiment, all PSM scoring models above are applied to the test set to check the effectiveness of each model. For GP-PostNovo, all 30 GP individuals from 30 independent runs along with the best GP program among all of them (in terms of Equation (6.3) on the training set) are selected to run Experiment II. More details are explained in the following section.

6.4.2 Results of Experiment II

The Results of Peptide Identification by PEAKS and GA-Novo

Before presenting the results of the performance evaluation phase of the models on the test set, more detail about the performance of each *de novo* sequencing algorithm used to generate the instances in the test sets is given here to help understand the *misRank* rate of each *de novo* tool before applying the PSM scoring functions generated by GP, RF, SVR and SVM.

Table 6.8 presents the results of *de novo* peptide sequencing by PEAKS and GA-Novo. To calculate the total number of predicted/identified first

Table 6.8: The results of *de novo* sequencing by PEAKS and GA-Novo using the MS/MS spectra from *test set*.

		FP		
Method	TP	# of correct PSMs which are not first ranked (missed correct PSMs)	# of spectra that their correct PSM does not exist among the 5 candidates	*misRank*
PEAKS	67	25	28	0.44
GA-Novo	84	24	12	0.3

ranked correct PSMs (the second column in the table) by each *de novo* sequencing method, for each spectrum the first ranked peptide is taken as the identification result and is compared with the ground-truth peptide to check whether it is a correct match or not. From the results of Table 6.8, it can be seen that out of 120 MS/MS spectra, PEAKS and GA-Novo only found 67 and 84 first ranked correct PSMs, respectively. However, 25 and 24 correct PSMs (the third column in the table) did not get the first rank by PEAKS and GA-Novo, respectively. Those PSMs are wrongly ranked by the *de novo* sequencing methods so we call them missed correct PSMs. In this case, the PSM scoring models are applied on the results of PEAKS and GA-Novo to check if they are able to re-score the correct PSMs and put them at the first rank. A PSM scoring model as a post-processing tool also should not mis-rank other correct PSMs that already got the first rank by the *de novo* sequencing methods. It is worth mentioning that out of 120 MS/MS spectra in this set, 28 and 12 of them were sequenced by PEAKS and GA-Novo, without having the correct peptides among the five candidate peptides for each spectrum. So when the *de novo* sequencing method does not identify the correct peptide in the candidate peptide list, the post-processing method cannot do anything further. So in summary, based on the results of Table 6.8, the *misRank* rate of PEAKS using the spectra in test set is 0.44 and that value for GA-Novo is 0.3. Therefore, it is expected that by applying the new

Table 6.9: The results of post-processing the output of PEAKS using the spectra from *test set* by RF, SVM, SVR and GP.

Method	# of missed correct PSMs got first-rank (out of 25)	# correct PSMs got missed (out of 67)	TP	FP	*misRank*
RF-regression	13	5	75	46	0.38
RF-classification	3	10	60	60	0.50
SVR	9	6	70	50	0.42
SVM	0	32	35	85	0.71
GP-PSM (30 runs)	17.79 ± 1.74	9.27 ± 1.44	75.52 ± 2.17	44.48 ± 2.17	0.37 ± 0.02
GP-PostNovo (30 runs)	**21.56 ± 1.77**	**4.20 ± 1.05**	84.36 ± 2.32	35.64 ± 2.32	**0.30 ± 0.02**
GP-PSM (best run)	20	8	79	41	0.34
GP-PostNovo (best run)	**24**	**3**	**88**	**32**	**0.27**

PSM scoring functions, the *misRank* rate reduces by decreasing the number of PSMs in the third column of Table 6.8.

Post-processing the Results of PEAKS

Table 6.9 presents the results of post-processing the output of PEAKS, which previously used the spectra from the test set to perform *de novo* peptide sequencing, by RF, SVR, SVM and GP. The second column of Table 6.9 shows the number of correct PSMs which previously did not get the first rank by PEAKS but now after applying the post-processing method these PSMs got the first rank. The third column presents the number of correct PSMs that were correctly identified by PEAKS but they are wrongly ranked (not first rank) by the new PSM scoring models. From the results of Table 6.9 it can be seen that RF-regression, SVR and the two GP methods improving the results of PEAKS *de novo* peptide sequencing by finding more missed correct PSMs (the second column) which results in decreasing the FPs and improving the *misRank* rank. It can be seen that GP-PostNovo has the highest number of identified missed correct PSMs (the second column) and the lowest number of mis-ranked correct PSMs (the third column) compared to all other methods.

In terms of identification of missed correct PSMs, GP-PostNovo is able to identify 96% $(= (\frac{24}{25} \times 100))$ of the missed correct PSMs whereas RF-regression, SVR, RF-classification only found 52%, 36% and 12%, respectively. The identification rate of missed correct PSMs for SVM is zero. Also the results of mis-identification of the correct PSMs (third column) by non-GP methods show that the GP methods are able to keep the correct PSMs which are already got the first rank by PEAKS at top ranks. Particularly, GP-PostNovo wrongly re-ranks the correct PSMs by 4% $(= (\frac{3}{67} \times 100))$, whereas RF-regression, SVR, RF-classification and SVM get the rate of 7%, 8%, 15% and 48%, respectively. The comparison between GP-PostNovo and GP-PSM shows that the scoring strategy in GP-PostNovo results in finding a larger number of missed PSMs while sacrificing a smaller number of correct PSMs which have already been identified by the *de novo* sequencing method.

Overall, GP-PostNovo reduces the *misRank* rate of PEAKS full-length peptide sequencing by 17% $(= (0.44 - 0.27) \times 100)$.

Post-processing the Results of GA-Novo

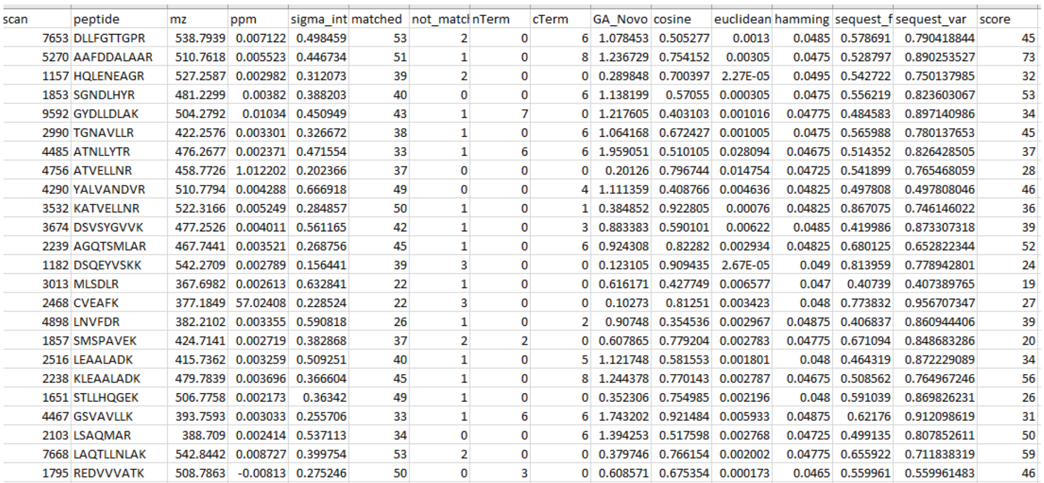
Table 6.10 presents the results of post-processing the output of GA-Novo, by the scoring functions generated by RF, SVR, SVM and the two GP-based methods. Similar to the results of post-processing PEAKS, GP-PostNovo outperforms other methods by increasing the identification rate of missed correct PSMs at 83% $(= (\frac{20}{24} \times 100))$, whereas RF-regression, SVR, RF-classification and SVM get the rate of 46%, 33%, 21% and 0%, respectively. Both GP methods try to minimise the rate of mis-identification of correct PSMs compared to the other methods. In summary, the results of the best run and 30 runs of GP-PostNovo show that the scoring functions have promising performance in terms of minimising the *misRank* of full-length peptide *de novo* sequencing by GA-Novo. This means that employing GP-PostNovo results in 15% $(= (0.3 - 0.15) \times 100)$ reduction in *misRank* rate of GA-Novo.

Table 6.10: The results of post-processing the output of GA-Novo using the spectra from *test set* by RF, SVM, SVR and GP.

Method	# of missed correct PSMs got first-rank (out of 25)	# correct PSMs got missed (out of 67)	TP	FP	*misRank*
RF-regression	11	5	90	30	0.25
RF-classification	5	15	74	46	0.38
SVM	0	40	44	76	0.63
SVR	8	6	86	34	0.29
GP-PSM **(30 runs)**	18.54 ± 1.74	12.70 ± 1.92	88.84 ± 2.29	31.16 ± 2.29	0.26 ± 0.02
GP-PostNovo (30 runs)	**19.98 ± 1.87**	**5.30 ± 1.83**	97.68 ± 3.00	22.32 ± 3.0	**0.19 ± 0.02**
GP-PSM (best run)	19	10	92	28	0.23
GP-PostNovo (best run)	**21**	**2**	**102**	**18**	**0.15**

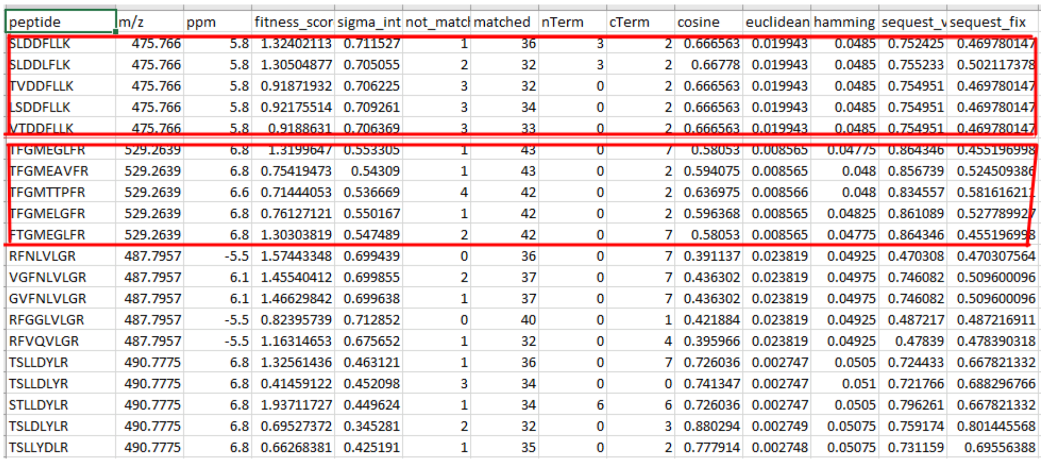
One question arises here that why other methods including RF-regression and SVR did not get results as good as GP. Further analysis on the scores that the PSMs got after re-ranking by these methods shows that in many cases there were two PSMs including a correct PSM and an incorrect one belonging to the same group where they both got the same score. In this situation based on the policy of our method, it is considered as a FP since the correct PSM gets the same rank as the incorrect PSM. However, GP-PostNovo is more discriminative and does not often give the same score to two PSMs. It is worth mentioning that re-scoring and re-ranking the missed correct PSMs from the results of both *de novo* sequencing methods is a challenging task. The reason is that the decoy sequence (the incorrect candidate sequence) is very similar to the target sequence and the difference might be in the position of one or a few amino acids, which is the reason that the *de novo* sequencing methods were unable to pick up the correct candidate sequence.

Figure 6.3 and Figure 6.4 shows a view of the instances in the training set and the test set, respectively. Comparing the peptide column of both training and test sets shows the challenge amino acid permutation complexity. As it can be seen, the candidate peptides in test set, in each group of 5 instances



scan	peptide	mz	ppm	sigma_int	matched	not_matcl	nTerm	cTerm	GA_Novo	cosine	euclidean	hamming	sequest_f	sequest_var	score
7653	DLLFGTTGPR	538.7939	0.007122	0.498459	53	2	0	6	1.078453	0.505277	0.0013	0.0485	0.578691	0.790418844	45
5270	AAFDDALAAR	510.7618	0.005523	0.446734	51	1	0	8	1.236729	0.754152	0.00305	0.0475	0.528797	0.890253527	73
1157	HQLENEAGR	527.2587	0.002982	0.312073	39	2	0	0	0.289848	0.700397	2.27E-05	0.0495	0.542722	0.750137985	32
1853	SGNDLHYR	481.2299	0.00382	0.388203	40	0	0	6	1.138199	0.57055	0.000305	0.0475	0.556219	0.823603067	53
9592	GYDLLDLAK	504.2792	0.01034	0.450949	43	1	7	0	1.217605	0.403103	0.001016	0.04775	0.484583	0.897140986	34
2990	TGNAVLLR	422.2576	0.003301	0.326672	38	1	0	6	1.064168	0.672427	0.001005	0.0475	0.565988	0.780137653	45
4485	ATNLLYTR	476.2677	0.002371	0.471554	33	1	6	6	1.959051	0.510105	0.028094	0.04675	0.514352	0.826428505	37
4756	ATVELLNR	458.7726	1.012202	0.202366	37	0	0	0	0.20126	0.796744	0.014754	0.04725	0.541899	0.765468059	28
4290	YALVANDVR	510.7794	0.004288	0.666918	49	0	0	4	1.111359	0.408766	0.004636	0.04825	0.497808	0.497808046	46
3532	KATVELLNR	522.3166	0.005249	0.284857	50	1	0	1	0.384852	0.922805	0.00076	0.04825	0.867075	0.746146022	36
3674	DSVSYGVVK	477.2526	0.004011	0.561165	42	1	0	3	0.883383	0.590101	0.00622	0.0485	0.419986	0.873307318	39
2239	AGQTSMLAR	467.7441	0.003521	0.268756	45	1	0	6	0.924308	0.82282	0.002934	0.04825	0.680125	0.652822344	52
1182	DSQEYVSKK	542.2709	0.002789	0.156441	39	3	0	0	0.123105	0.909435	2.67E-05	0.049	0.813959	0.778942801	24
3013	MLSDLR	367.6982	0.002613	0.632841	22	1	0	0	0.616171	0.427749	0.006577	0.047	0.40739	0.407389765	19
2468	CVEAFK	377.1849	57.02408	0.228524	22	3	0	0	0.10273	0.81251	0.003423	0.048	0.773832	0.956707347	27
4898	LNVFDR	382.2102	0.003355	0.590818	26	1	0	2	0.90748	0.354536	0.002967	0.04875	0.406837	0.860944406	39
1857	SMSPAVEK	424.7141	0.002719	0.382868	37	2	2	0	0.607865	0.779204	0.002783	0.04775	0.671094	0.848683286	20
2516	LEAALADK	415.7362	0.003259	0.509251	40	1	0	5	1.121748	0.581553	0.001801	0.048	0.464319	0.872229089	34
2238	KLEAALADK	479.7839	0.003696	0.366604	45	1	0	8	1.244378	0.770143	0.002787	0.04675	0.508562	0.764967246	56
1651	STLLHQGEK	506.7758	0.002173	0.36342	49	1	0	0	0.352306	0.754985	0.002196	0.048	0.591039	0.869826231	26
4467	GSVAVLLK	393.7593	0.003033	0.255706	33	1	6	6	1.743202	0.921484	0.005933	0.04875	0.62176	0.912098619	31
2103	LSAQMAR	388.709	0.002414	0.537113	34	0	0	6	1.394253	0.517598	0.002768	0.04725	0.499135	0.807852611	50
7668	LAQTLLNLAK	542.8442	0.008727	0.399754	53	2	0	0	0.379746	0.766154	0.002002	0.04775	0.655922	0.711838319	59
1795	REDVVVATK	508.7863	-0.00813	0.275246	50	0	3	0	0.608571	0.675354	0.000173	0.0465	0.559961	0.559961483	46

Figure 6.3: The instances in the training set.



peptide	m/z	ppm	fitness_scor	sigma_int	not_matcl	matched	nTerm	cTerm	cosine	euclidean	hamming	sequest_v	sequest_fix
SLDDFLLK	475.766	5.8	1.32402113	0.711527	1	36	3	2	0.666563	0.019943	0.0485	0.752425	0.469780147
SLDDLFLK	475.766	5.8	1.30504877	0.705055	2	32	3	2	0.66778	0.019943	0.0485	0.755233	0.502117378
TVDDFLLK	475.766	5.8	0.91871932	0.706225	3	32	0	2	0.666563	0.019943	0.0485	0.754951	0.469780147
LSDDFLLK	475.766	5.8	0.92175514	0.709261	3	34	0	2	0.666563	0.019943	0.0485	0.754951	0.469780147
VTDDFLLK	475.766	5.8	0.9188631	0.706369	3	33	0	2	0.666563	0.019943	0.0485	0.754951	0.469780147
TFGMEGLFR	529.2639	6.8	1.3199647	0.553305	1	43	0	7	0.58053	0.008565	0.04775	0.864346	0.455196998
TFGMEAVFR	529.2639	6.8	0.75419473	0.54309	1	43	0	2	0.594075	0.008565	0.048	0.856739	0.524509386
TFGMTTPFR	529.2639	6.6	0.71444053	0.536669	4	42	0	2	0.636975	0.008566	0.048	0.834557	0.58161621
TFGMELGFR	529.2639	6.8	0.76127121	0.550167	1	42	0	2	0.596368	0.008565	0.04825	0.861089	0.527789927
FTGMEGLFR	529.2639	6.8	1.30303819	0.547489	2	42	0	7	0.58053	0.008565	0.04775	0.864346	0.455196998
RFNLVLGR	487.7957	-5.5	1.57443348	0.699439	0	36	0	7	0.391137	0.023819	0.04925	0.470308	0.470307564
VGFNLVLGR	487.7957	6.1	1.45540412	0.699855	2	37	0	7	0.436302	0.023819	0.04975	0.746082	0.509600096
GVFNLVLGR	487.7957	6.1	1.46629842	0.699638	1	37	0	7	0.436302	0.023819	0.04975	0.746082	0.509600096
RFGGLVLGR	487.7957	-5.5	0.82395739	0.712852	0	40	0	1	0.421884	0.023819	0.04925	0.487217	0.487216911
RFVQVLGR	487.7957	-5.5	1.16314653	0.675652	1	32	0	4	0.395966	0.023819	0.04925	0.47839	0.478390318
TSLLDYLR	490.7775	6.8	1.32561436	0.463121	1	36	0	7	0.726036	0.002747	0.0505	0.724433	0.667821332
TSLLDLYR	490.7775	6.8	0.41459122	0.452098	3	34	0	0	0.741347	0.002747	0.051	0.721766	0.688296766
STLLDYLR	490.7775	6.8	1.93711727	0.449624	1	34	6	6	0.726036	0.002747	0.0505	0.796261	0.667821332
TSLDLYLR	490.7775	6.8	0.69527372	0.345281	2	32	0	3	0.880294	0.002749	0.05075	0.759174	0.801445568
TSLLYDLR	490.7775	6.8	0.66268381	0.425191	1	35	0	2	0.777914	0.002748	0.05075	0.731159	0.69556388

Figure 6.4: The instances in the test set.

are quite similar to each other with each instance having very close feature values to the instances belonging to the same spectrum/group. Therefore, the PSM scoring function generated by either GP or other methods should have a strong discriminative ability to give the highest score to the correct peptide in the test set. The learning strategy in GP-PostNovo, which makes GP to evolve computer programs to solve the regression and classification problems simultaneously, enables GP to assign appropriate scores to PSMs, giving the greatest score to the correct PSM which brings it ahead of all incorrect ones. However, other methods give almost the same float point scores to the candidate peptides belonging to the same group. Further analysis on the scores that the candidate peptides got after re-ranking by other non-GP based methods shows that in many cases there were two candidate peptides including a correct peptide and an incorrect peptide belonging to the same group where they both got the same score. In such cases, the instances (PSMs) have almost the same feature values, therefore the scoring function should be discriminative enough to be able to distinguish the correct PSM from the incorrect PSMs.

6.4.3 Analysis on the Best GP Evolved Program

Table 6.11 presents the frequency of the appearance of all features in the GP tree of GP-PSM and GP-PostNovo. As previously mentioned, GP-PSM solves a regression task, while GP-PostNovo solves a classification and a regression problem at the same time. Therefore, it is not unexpected that GP-PostNovo covers GP-PSM and generates a bigger tree. Analysis of the GP tree evolved by GP-PostNovo and GP-PSM reveals interesting relationships between the features which are discussed as follows.

The features in Table 6.11 can be divided into two groups, spectral features (from f_1 to f_7) and distance-based features (from f_8 to f_{12}). The spectral features are inspired of CID fragmentation rules, whereas distance-based features use distance measures to calculate the similarity/dissimilarity between the two vectors corresponding to the experimental spectrum and the theoretical spectrum. From Table 6.11, it can be seen that none of the three methods use deltaMass feature, f_1, as it produces very small values almost close to zero for most of the PSMs. So GP discarded this feature as it does not seem discriminative enough to help GP distinguish a correct match from the incorrect ones.

The GP tree produced by GP-PSM (see Figure 6.5) shows that the left sub-tree mainly consists of the combination of all spectral features (separated by the blue dashed line), whereas the right sub-tree mostly contains the distance-based features (showed by a purple dotted line). However, Anal-

Table 6.11: Frequency of the appearance of the features in the three GP-based PSM scoring functions.

	Feature	GP-PSM	GP-PostNovo
f_1	deltaMass	0	0
f_2	$\text{I}_{matched}$	8	8
f_3	$\text{N}_{matched}$	4	9
f_4	$\text{N}_{not\text{-}matched}$	4	0
f_5	Nterm	2	3
f_6	Cterm	1	5
f_7	GA-Novo	-	2
f_8	Cos	7	5
f_9	Euc	0	2
f_{10}	Hamming	2	4
f_{11}	SeqFix	3	10
f_{12}	SeqVar	1	0
Sum of the appearance of all features.		32	48

ysis on the GP tree of the GP-PostNovo method (see Figure 6.6) shows that a number of times features from different groups are used together by the arithmetic operators e.g., f_3 with f_{10}, or f_2 along with f_{11}. This is shown by drawing the blue and purple dotted lines around the spectra and distance-based features, respectively. This might indicate that the combination of features from different groups could be more discriminative rather than combining all features from the same group.

As in the Chapter 5, the two features Nterm and Cterm showed to be very useful to find the best match from N-terminus and C-terminus of the spectrum, these two features appeared twice in the form of $(f_5 * f_6)$ by GP-PostNovo (see the green dotted lines around the nodes in the GP tree shown in Figure 6.6), whereas such a relationship was not discovered by the best evolved tree by GP-PSM. Moreover, due to the incomplete fragmentations, some spectra might have less peaks in the N-terminus and more peaks in the C-terminus. That is probably the reason that GP-PostNovo selected Cterm

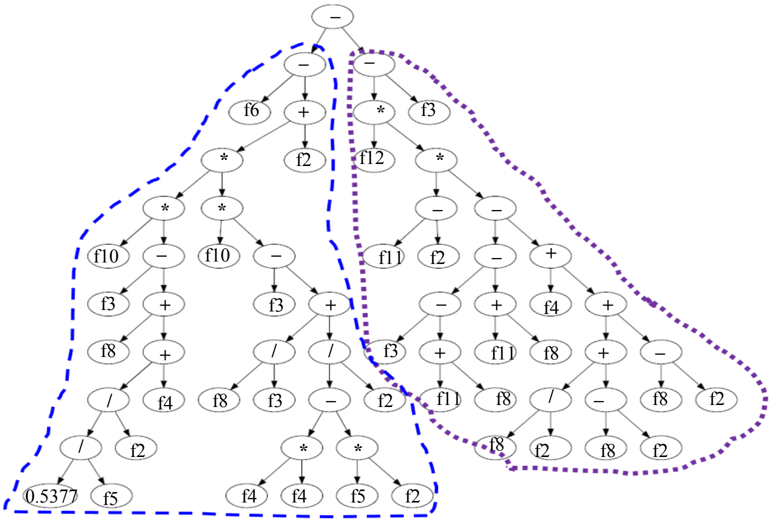


Figure 6.5: The best GP evolved program of GP-PSM.

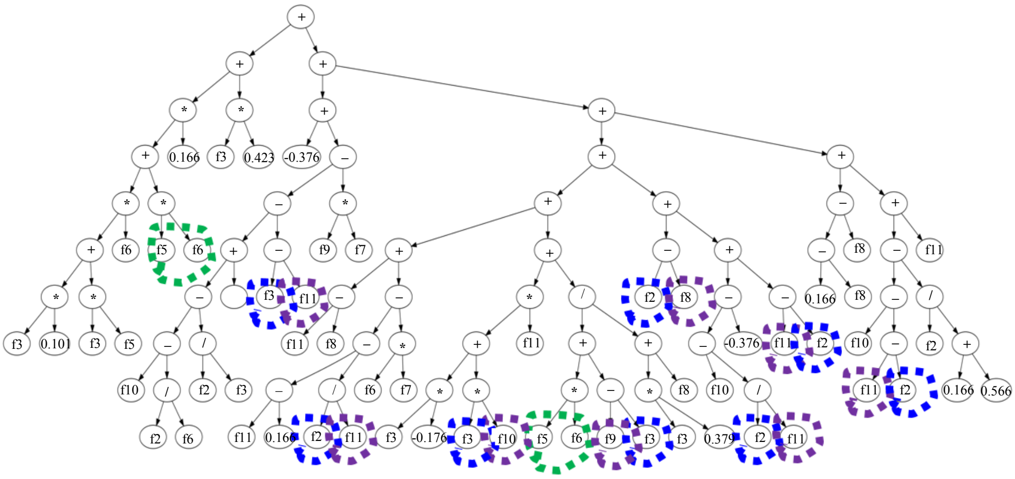

+
+ +
* * + +
+ 0.166 f3 0.423 -0.376 – + +
* * – * + + – +
+ f6 f5 f6 + – f9 f7 + + – + – f8 – f11
* * – f3 f11 – – * / f2 f8 – – 0.166 f8 – /
f3 0.101 f3 f5 – / f11 f8 – * + f11 + + – -0.376 f11 f2 f10 – f2 +
f10 / f2 f3 – / f6 f7 * * * – * f8 f10 / f11 f2 0.166 0.566
f2 f6 f11 0.166 f2 f11 f3 -0.176 f3 f10 f5 f6 f9 f3 f3 0.379 f2 f11

Figure 6.6: The best GP evolved program of GP-PostNovo.

features more frequent than Nterm features, but this pattern was not seen in GP-PSM. Due to the frequent selection of f_3 and f_{12}, GP-PostNovo does not select the other two features f_4 and f_{12} as they might seem to have similar functionalities to GP.

6.5 Chapter Summary

The goal of this chapter was developing a new method for post-processing the results of *de novo* sequencing methods, aiming at optimally re-ordering the PSMs and minimising the *misRank* of the full-length *de novo* peptide sequencing. The goal has been successfully achieved by developing a GP-based method, named GP-PostNovo, which automatically generated effective scoring functions that can re-score and re-rank the results of *de novo* sequencing and increased the accuracy of *de novo* sequencing at the peptide level. A set of 12 features, which represented the important characteristics about the goodness of the match between each spectrum and its corresponding peptide, were extracted from the data. During the learning process, GP incorporated the important features into its scoring function. The fitness function of GP which combines the regression error and mis-ranking error allowed GP to minimise the regression error rate which helped GP give accurate scores to PSMs and classification error rate, which led GP to distinguish the correct PSM from the incorrect ones. So as the result, the correct PSM corresponding to the input spectrum would be ranked ahead of all incorrect candidate ones belonging to the same spectrum.

For comparing the results of GP with other benchmark machine learning algorithms, RF, SVM and SVR were used to build the PSM scoring functions. GP and other methods were trained using a set of MS/MS spectra with known identification corresponding to unique peptides from full factorial LC-MS/MS proteomics benchmark dataset. To evaluate the effectiveness of the PSM scoring functions including GP, two *de novo* sequencing methods, PEAKS and GA-Novo were used to perform *de novo* sequencing using a test set containing 120 MS/MS spectra from the original dataset. All GP-based and Non-GP based scoring functions generated in this chapter were applied to the outputs of each *de novo* sequencing method. The results showed that GP-PostNovo outperformed all methods in terms of mis-ranking reduction. GP-PostNovo increased the identification of full-length correct peptides of

PEAKS and GA-Novo by 17% and 15%, respectively. Among other non-GP methods, RF-regression was the second best post-processing method, by 6% and 5% improvement in identification of full-length correct peptides from the results of PEAKS and GA-Novo, respectively.

Chapter 7

Conclusions

This chapter concludes each research objectives of this thesis, providing main findings from each individual chapter. Directions for future developments and potential research areas are presented at the end of this chapter.

In recent years, various new methods and software tools have been proposed for *de novo* sequencing. Although these methods were successfully applied to perform *de novo* sequencing in various cases, they have limitations which make these methods less accepted by the community. We believe that the performance of these methods can be further improved by identifying the challenges that these methods have faced and developing computational approaches to overcome the limitations.

Through this thesis, three main steps including preprocessing, sequence optimisation and post-processing were identified that could potentially assist the performance of the current methods. From the machine learning point of view, these steps could be formulated as three tasks of classification, optimisation and regression. As evolutionary algorithms showed promising results in solving these problems, the overall goal of this thesis was shaped as investigating the capability of evolutionary algorithms particularly GP and GAs in addressing these task in order to improve the peptide identification with *de novo* peptide sequencing. The goal was successfully achieved by proposing four effective EA-based methods and the results have shown that each method was able to improve the *de novo* sequencing results at either the amino acid level, peptide level or both.

The rest of this chapter provides conclusions for each research objectives proposed in this thesis and new directions are discussed for further improvement in the future.

7.1 Achieved Objectives

This thesis has successfully achieved the following research objectives:

- This thesis developed a GP-based method for preprocessing the imbalanced MS/MS spectra, aiming at improving the reliability of peptide identification by removing more noise peaks and retaining more signal peaks (chapter 3). The proposed GP method proved to be the most stable method across various ratios of signal to noise compared to six well-known classification algorithms. The proposed method uses an effective weighted fitness function that accounts for both the minority and the majority class accuracies (sensitivity and specificity) in the evolved classifiers. The design of this fitness function allows GP for more noise reduction and more signal retention when classifying the MS/MS data. The proposed method was used to denoise the MS/MS spectra prior to peptide identification with a *de novo* sequencing software and a database searching tool. The results show that the confidence of peptide identification for both methods are successfully improved.

- This thesis proposed a multi-objective approach, named MOGP/D (chapter 4), for the purpose of evolving a Pareto front of classifiers along the optimal trade-off surface that offers the best compromises between the two conflicting objectives of sensitivity and specificity in classification of imbalanced MS/MS spectra. In order to generate the non-dominated solutions for Pareto front approximation, MOGP/D utilises the MOEA/D framework which decomposes an MOP into a set of N scalar objective optimisation sub-problems, trying to optimise these N scalar optimisation sub-problems simultaneously. MOGP/D has shown to be promising for evolving a set of non-dominated classifiers that have the best trade-off between the two conflicting objectives of the majority and minority class accuracies in the problem of classification of imbalanced MS/MS spectra. The non-dominated solutions (classifiers) can be used for classification of peaks in the MS/MS spectra and the decision maker can choose one of them based on his/her preference.

- This thesis developed a GA-based method, GA-Novo, which utilises the "global" search ability of GA to solve the complex optimisation task of *de novo* peptide sequencing, aiming at full-length *de novo* sequencing of MS/MS spectra. Through four main components proposed in this method, GA-Novo outperformed PEAKS, the most commonly used *de novo* sequencing algorithm [9, 77], by a reasonably large margin of 8% at the peptide level. The major contributions presented in this work are: (1) a new tag-based initialisation method which helps GA find better solutions and converge faster; (2) a new fitness function which evaluates the quality of match between the input spectrum and the candidate peptide sequence from different aspects; (3) Nterm-Cterm crossover which is designed for creating superior offsprings inheriting the best parts of their parents; (4) conflict-mass mutation which is designed to solve the problem of di-peptides and conflict masses.

- Through this thesis an effective PSM scoring function, named Post-Novo, that contributes towards minimising the false discovery rate in the results of *de novo* peptide sequencing was proposed. This is the first time that GP is used in a framework to solve a regression task and a classification task at the same time. Through regression GP learnt to give appropriate scores to the PSMs and via classification GP improved its scoring ability by learning to give the highest scores to the correct PSMs. PostNovo was able to re-score and "optimally" re-rank the results of *de novo* sequencing with PEAKS and GA-Novo and successfully minimised the false identification rate of these two algorithms at the peptide level. Evaluating the performance of the proposed method at the peptide level is so rigorous, however it allows us to measure the common *de novo* sequencing errors, such as amino acid permutations.

7.2 Main Conclusions

Overall, this thesis contributed to the field of evolutionary algorithms, by proposing new applications of GP and GA for more accurate and efficient identification of peptides from MS/MS with *de novo* sequencing. This thesis finds that the proposed EA-based methods had contributions to substantially improving peptide identification with *de novo* sequencing particularly full-length *de novo* sequencing on CID spectra.

The main conclusions for the four research objective drawn from the four contribution chapters (Chapter 3 to Chapter 6) are presented in this section.

7.2.1 Single Objective GP for Preprocessing MS/MS spectra

This thesis proposes the first GP-based preprocessing method for denoising MS/MS spectra in order to improve the reliability of peptide identification. The experimental results on a large-scale dataset show that the proposed GP-based preprocessing method (CID-GP) improves the reliability of peptide identification and increases the identification rate of PEAKS by 26.6% compared to the un-preprocessed data and 19.3% over the best value of threshold-based method. The proposed method helps PEAKS identify more highly confident peptides with scores $70 \leq ALC \leq 99$ compared to the other cases. Moreover, with 95% confidence interval, the results of the

SEQUEST database search tool using the data preprocessed by GP are statistically significantly better than those with the un-preprocessed data and the best threshold-based method.

Important Ion Types

This thesis performed a detailed investigation on different CID fragmentation ion types to figure out the important ion types in peptide identification that have contribution towards high confidence peptide identification. This was done by analysing how different peptide identification tools interact with different sets of ions/peaks presented in the spectrum. It is found that considering only CID singly-charged ions as signal peaks and the rest of the peaks as noise peaks can guarantee to obtain a reasonable high confidence peptide identification with both database searching and *de novo* sequencing. This finding helps make a clear definition of background noise in the CID data and allows to have a better decision on selecting the suitable ion types for labelling the peaks in MS/MS datasets which are used by the machine learning methods.

Interpretability of GP

From a set of different spectral features and fragmentation rules introduced in the literature, GP with its implicit feature selection ability was able to identify a set of features that contribute to predicting the ion types. For example, analysis of the evolved GP trees shows which isotopic and neutral losses features provide more evidence for finding the singly-charged b-/y-ions in CID spectra. The interpretability of GP helps identify the important spectral and fragmentation rule features that have positive influence on interpretation of CID fragmentation patterns. This finding is useful for the existing machine learning based methods that only rely on raw data to learn the fragmentation patterns, but they require a large amount of data which makes the training process computationally expensive.

7.2.2 Multi-objective GP for Imbalanced MS/MS spectra

This work showed how multi-objective can be used in GP to deal with the imbalanced MS/MS data. The proposed method showed a significant reduction of noisy peaks in the MS/MS spectra and increased the quality of spectra, leading to improve the reliability of peptide identification with PEAKS. In the range of $70 \leq ALC \leq 99$, MOGP/D improved the reliability of peptide identification by more than 21% compared with CID-GP, the single objective GP in chapter 3.

Bloat Control

The analysis on the average size of the GP programs evolved by NSGP and MOGP/D over 30 MOGP runs shows an unnecessary growth of the GP trees known as bloat or code growth in NSGP on the training sets. This results in low generalisation of NSGP on the test sets. It is found that MOGP/D is able to partially handle the bloat problem, resulting in a significant improvement in the objective values on the test sets compared to NSGP. This finding indicates that the GP size does not need to be considered as the third objective in the design of MOGP based on MOEA/D, but further investigation is needed.

Stability to Noise Ratios

In real-world systems, we do not only try to find the optimal design but also a robust design [207]. This thesis finds that MOGP based on MOEA/D is more stable than MOGP based on NSGA-II in classification of peaks in imbalanced MS/MS data across various noise ratios in terms of convergence to the Pareto front. Moreover, the best compromise solutions evolved by MOGP/D compared to the those of NSGP and single objective GP achieved better results in terms of retaining more number of signal peaks and filtering out more noise peaks across different S/N ratios on both the training and test sets.

7.2.3 GAs for Sequence Optimisation

De novo peptide sequencing includes global optimisation on noisy and incomplete data. In this thesis, we developed a GA-based *de novo* sequencing method which is reasonably robust to noise and missing ions and provides an effective solution for predicting the amino acid sequence of an input spectrum. Amino acid permutation is so challenging in *de novo* sequencing but the proposed method is capable of finding full-length sequences which have not been correctly predicted by its counterpart *de novo* sequencing tool. GA-Novo outperformed PEAKS by 8% higher accuracy at the peptide level (fully matched peptide sequences) and 4% higher accuracy at the amino acid level (partially matched sequences). Moreover, GA-Novo outperformed a GA-based *de novo* sequencing method, which uses random initialisation coupled with simple genetic operators, by a large margin of 47% at the amino acid level and 50% at the peptide level.

Representation

Among other evolutionary computation techniques, such as particle swarm optimisation (PSO) from swarm intelligence (SI) or genetic programming (GP) from evolutionary algorithms (EAs), GA is the most suitable method to solve the problem of *de novo* sequencing. The variable-length representation of GA used in this study allows keeping a peptide sequence containing a series of amino acids with any size. Therefore, this method can be used to sequence any peptide with any length. The representation can be easily adapted to include the numerical values of post-translation modifications as well.

Initialisation

Initialisation determines where the algorithm starts the search process. It is found that the tag-based initialisation used in the design of GA-Novo was a better starting point for GA and provided fitter initial population that can benefit the whole evolutionary process. In addition, the spacial design of the initialisation method and the ability of GA in searching a big search space makes GA-Novo to be independent of using a preprocessing method to denoise the spectra prior to *de novo* sequencing.

Effective Genetic Operators

The thesis finds that introducing various domain-dependent genetic operators along with standard operators (i.e., two point crossover) gives GA chances for better convergence and diversity. Although traditional two parents crossover operators are more “biology inspired”, the Nterm-Cterm crossover designed for GA-Novo shows that a multi-parents operator generates higher quality individuals.

Accurately predicting di-peptides is very challenging for the current *de novo* sequencing methods due to the vastly increased search space of the peptides. It is found that designing an appropriate mutation operator that considers a dictionary of conflict masses enables GAs to identify the correct di-peptides. This results in constructing full-length peptides whereas other methods failed.

7.2.4 Controlling PSM False Discovery

The low accuracy of the existing *de novo* sequencing methods at the peptide level due to the lack of suitable PSM-scoring functions to measure the goodness of a match was the reason to propose a new quality control strategy in this thesis. Chapter 6 proposes a new GP-based approach to validating the results of *de novo* sequencing in order to increase the accuracy of prediction at the peptide level. GP was used to generate computer programs as ranking functions which are supposed to post-process the results of peptide *de novo* sequencing to minimise the false identification. To evaluate the impact of the proposed method on the existing *de novo* sequencing methods, PEAKS, and GA-Novo (proposed in Chapter 5) were used to perform *de novo* sequencing on a test set containing 120 MS/MS spectra. GP-PostNovo post-processed the results of both methods and outperformed all its counterpart methods in terms of mis-ranking reduction. The proposed method improved the performance of PEAKS and GA-Novo by 17% and 15% at the peptide level, respectively.

Simultaneous Learning and Solving

This thesis finds the ability of GP in learning from different sources of training sets at the same time and solving two tasks simultaneously, while other machine learning methods do not have this capability. The application of the proposed method is not only limited to optimising the scores of PSMs, any problem which is involved with scoring and ranking can benefit from this approach. For example, search engine queries and recommender systems that use ranking algorithms to retrieve the relevant results to the user queries can gain advantage from this method. Clearly the ability of GP in automatically evolving a model that best fits the dataset without having a prior knowledge is also a key point here.

7.3 Future Work

Finally, future developments of research in this area are highlighted in this section.

7.3.1 Appropriate Genetic Operators for Post-translational Modifications (PTMs)

Considering different types of mutation operators in GA might be useful for better addressing the amino acid permutation complexity. It also can be helpful to identify the potential PTMs. For example, designing appropriate mutation operators to substitute an amino acid according to a probability from a substitution matrix or from a list of known PTMs are potential directions that can be considered in the future research. Depending on the input parameters by user, these mutations which lead to decreasing the number of gaps in the candidate amino acid sequences can be applied. Moreover the dictionary of di-peptide conflict masses can be further extended to cover all possibilities and this could potentially increase the number of fully matched sequences.

7.3.2 GA and Deep Learning for De Novo Sequencing

Our work opens a door for combining GA with deep learning to solve the complex optimisation problem of *de novo* sequencing. Deep Learning methods which have been recently proposed for peptide and protein identification, can be further enhanced with more advanced search algorithms as the authors have stated [11]. GA can be used to replace dynamic programming due to its good ability to search the large search space of possible amino acid sequences at the presence of noise. While deep learning can be used to produce a probability distribution over the amino acid classes, GA can filter out redundant amino acids or fill up the gaps due to the missing ions or PTMs.

7.3.3 Generic PSM Scoring Function

Clearly, the ability of GP to automatically evolve a model without any prior knowledge about the structure of the model has been successfully presented in this thesis. This study trains GP on a particular class of fragmentation techniques, however, the GP-based PSM scoring function has the potential for further improvement if it is trained on different spectra with different charge numbers and precursor masses. This could be done by collecting well-annotated, gold standard training and testing datasets. While this study shows several putative applications of GP to improve the peptide identification, subsequent improvements of the models will presumably expand the scope of the identification of modified and unmodified peptides.

Moreover, it is worth investigating the effectiveness of using the GP-based PSM scoring model as the fitness function of GA-Novo. The current fitness function of GA contains a set of terms (scores) which their combination is not optimised properly. It is expected that the learnt scoring function by GP could potentially improve the results of GA-Novo. Because it is trained using thousands of MS/MS spectra and is learned the fragmentation pattern reasonably well.

7.3.4 Multi-objective Optimisation

The post-processing GP-based method to identify true and false PSMs can be extended to a multi-objective approach to maintaining a trade-off between specificity and sensitivity which are two competitive objectives in the results of *de novo* peptide sequencing. Moreover, *de novo* sequencing of modified peptides can be handled via multi-objective with conflicting objectives of scores such as false identification rate and constraints such as the number of PTMs or GP program size.

7.3.5 Combined use of MS1 and MS2

Since the focus of this thesis is at MS2 peaklists, it is worth exploring PIF (precursor or product ion fraction) at the MS1 level and investigating the effectiveness of using this information in-conjunction with the MS2 information. This could be achieved by exporting the raw mass spectrometry data to mzML or mzXML using open-source MS-based proteomics softwares (e.g. Maxquant [208]) and exploring the level of chimericity in the dataset. PIF values might contribute as an important feature for preprocessing and elucidate and/or highlight the “imbalance” signal to noise as discussed in this thesis.

7.3.6 An EA-assisted Peptide Identification Framework

The three components proposed in this thesis are responsible to improve the peptide identification with *de novo* sequencing. As each component (objective of this thesis) is proposed and developed individually, in our future work each component will be implemented as a *plug-in*. So each plug-in can be integrated into the workflow of the current peptide identification tools in a coherent manner in order to improve the peptide identification from MS/MS spectra. Moreover, we will carefully put all components in a pipeline to make our first EA-assisted end-to-end peptide identification framework.

Glossary

amino acid

Amino acid are organic compounds that combine to form proteins. There are 20 common amino acids where each contains an amino group (i.e. NH2) and a carboxylic acid group (i.e. COOH) with a side-chain structure (R) specific for each amino acid.

amino acid level

The fraction of partially matched peptides.

b-/y-ions

They are the most common peptide fragments observed in low energy collisions. In the CID fragmentation technique, the amino acid sequence of an MS/MS spectrum can be determined by the mass differences between b- (or y-) ions.

biological samples

Samples such as blood, urine, tissue, saliva, and many other types which are collected for a variety of reasons from trial subjects.

biomarker

Refers to anything that can be used as a measurable indicator of a particular biological disease state or condition.

classification

Classification is a task of classifying a new unseen observation into a set of groups that are already known based on the labelled training datasets.

collision-induced dissociation

An extensively studied technique which is known to be highly suitable for the identification of peptide sequences. In this technique, fragmentation happens at the peptide bonds, producing b-/y-ions.

C-terminus

Refers to the end of an amino acid chain.

database search

Peptide identification can be performed by comparing the input experimental MS/MS spectra with theoretical spectra simulated for each peptide sequence in a protein sequences database.

de novo

In Latin means “starting from the beginning” and literally means “again”).

de novo peptide sequencing

The process of determining the amino acid sequence of peptides directly from MS/MS spectra without using a protein database.

enzymes

Being biological catalysts, enzymes are able to accelerate the chemical reactions.

evolutionary algorithm

A search technique which is based mainly on Darwinian principle of natural selection. Evolutionary algorithms (EAs) employ techniques such as recombination, mutation, natural selection and survival of the fittest in order to evolve a population of individuals to solve the problem.

evolutionary computation

Being a subfield of artificial intelligence, evolutionary computation (EC) is a family of population-based problem solving techniques whose employs principles based on the theory of biological evolution to get involved in many optimisation problems.

experimental MS/MS spectrum

MS/MS spectra generated through high-throughput proteomics experiments.

fragmentation

Dissociating the precursor ions is called fragmentation. The pattern in the mass spectrum of a fragmented molecule can be used to determine structural information of the molecule.

Genetic Algorithms

Genetic algorithm (GA) represents an individual as a fixed-length bit string called it a chromosome. GA aims to decode the chromosomes to get the solution for the problem being faced by employing genetic operators.

Genetic Programming

Genetic Programming (GP) is an evolutionary algorithm which uses a variable-length individual representation to evolve population of computer programs to automatically build or evolve a model to tackle the problem.

imbalanced

In classification problems when the class distribution is not uniform among the classes, the dataset is called imbalanced..

ladder

A complete peptide fragmentation gives a contiguous series of ion types which is called ladder.

mass spectrometer

Measures the masses within a sample by ionizing the sample and sorting the ions based on their mass-to-charge ratio (m/z). These masses later can be used for identification of the proteins or peptides in the samples.

mass spectrometry

Mass spectrometry has been practically recognised as the primary tool for protein identification and measures the mass-to-charge ratio of ions.

The mass spectrometry (MS^1) analyses the ionised peptides and generates an MS spectrum which is a plot composed of the m/z values of the ions (precursor ions) and their corresponding relative intensities.

MS spectrum

An MS spectrum corresponds to a protein. A mass spectrum is a stick diagram of the number of ions detected as a function of their m/z ratios.

MS/MS spectrum

An MS/MS spectrum corresponds to a peptide. It consists of a list of peaks each having a mass-to-charge ratio (m/z) value and an intensity value (peak height). The m/z values are results of ionizing the biological samples and their intensities indicate the abundance of ions.

N-terminus

Refers to the start of a protein or a polypeptide.

optimisation

An optimisation problem involves defining an objective function and aims at minimising or maximising it by systematically choosing appropriate input values.

parent mass

The mass of the peptide, which is called parent mass, equals to the total mass of its amino acids plus mass of water.

peptide

Short chains of amino acids (from 2 to 50 amino acids) joined by peptide bonds.

peptide identification

The process of assigning an MS/MS spectrum to a peptide is called peptide identification. Peptide identification can be performed by comparing the input experimental MS/MS spectrum with theoretical spectra predicted for each peptide sequence in a protein sequence database (*database search* method). Alternatively, instead of searching the experimental MS/MS spectrum against a database, peptide sequences can be extracted directly from the spectrum with the *de novo* sequencing approach.

peptide level

The fraction of fully matched peptides.

peptide-spectrum match

Matching an MS/MS spectrum to a peptide. This pair is called a peptide-spectrum match (PSM). The goodness of the match is measured by a scoring function.

post-translational modification

A naturally occurring chemical modification of a protein. Post-translational modifications (PTMs) can alter the properties of a protein by proteolytic cleavage at a peptide bond or by addition of a modifying group to one or more amino acids.

precursor peptide ion

In MS/MS technique, a selected precursor ion is fragmented into fragment ions whose m/z values are measured by mass spectrometry to generate an MS/MS spectrum.

protease

An enzyme that breaks the peptide bonds of proteins.

protein

Micro-molecules that are made up of a long chain of amino acids linked together in a linear sequence.

protein identification

The process of assigning an MS spectrum to a protein is called protein identification. An MS spectrum can be identified by matching the measured masses of the spectrum to the corresponding peptide masses of a protein from a protein database. This process is called protein identification by peptide mass fingerprinting (PMF) method.

proteomics

Proteomics is the study of proteomes and their functions. A proteome is a set of proteins produced in an organism, system, or biological context. Proteomics analysis is the systematic identification and quantification of proteins, particularly their sequences, structures and functions at a certain point in time. Identification of protein sequences and their modifications is very important in proteomics because it allows researchers discovering possible genetic diseases in an organism.

regression

Regression is a process that aims to discover the relationship between inputs and outputs. A regression problem attempts to find a mathematical model that predicts a real value for each input example and measures the error of the prediction in an iteratively way.

residue

An amino acid within a peptide chain is called a residue due to the loss of water.

tandem mass spectra

Tandem mass spectrometry (MS/MS) involves multiple steps of mass spectrometry along with fragmentation occurring in between the steps [84].

theoretical spectrum

The theoretical spectrum contains only signal peaks (b- and y-ions) with no noise peaks. It is virtually constructed based on the CID fragmentation rules.

trypsin

An enzyme that helps us digest protein.

Bibliography

[1] Akhilesh Pandey and Matthias Mann. Proteomics to study genes and genomes. *Nature*, 405(6788):837–846, 2000.

[2] W Michael Caudle, Sheng Pan, Min Shi, Thomas Quinn, Jake Hoekstra, Richard P Beyer, Thomas J Montine, and Jing Zhang. Proteomic identification of proteins in the human brain: Towards a more comprehensive understanding of neurodegenerative disease. *Proteomics Clinical Applications*, 2(10-11):1484–1497, 2008.

[3] Hanno Steen and Matthias Mann. The ABC's (and XYZ's) of peptide sequencing. *Nature Reviews Molecular cell biology*, 5(9):699–711, 2004.

[4] Robin Gras, David Hernandez, Patricia Hernandez, Nadine Zangge, Yoann Mescam, Julien Frey, Olivier Martin, Jacques Nicolas, and Ron D Appel. Cooperative metaheuristics for exploring proteomic data. *Artificial Intelligence Review*, 20(1-2):95–120, 2003.

[5] Alexey I Nesvizhskii. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of Proteomics*, 73(11):2092–2123, 2010.

[6] Jimmy K Eng, Tahmina A Jahan, and Michael R Hoopmann. Comet: an open-source MS/MS sequence database search tool. *Proteomics*, 13(1):22–24, 2013.

[7] Sangtae Kim and Pavel A Pevzner. MS-GF+ makes progress towards a universal database search tool for proteomics. *Nature Communications*, 5:5277, 2014.

[8] Fengchao Yu, Ning Li, and Weichuan Yu. PIPI: PTM-invariant peptide identification using coding method. *Journal of Proteome Research*, 15(12):4423–4435, 2016.

[9] Bin Ma, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. PEAKS: powerful software for peptide *de novo* sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry*, 17(20):2337–2342, 2003.

[10] Bin Ma. Novor: real-time peptide *de novo* sequencing software. *Journal of the American Society for Mass Spectrometry*, 26(11):1885–1894, 2015.

[11] Ngoc Hieu Tran, Xianglilan Zhang, Lei Xin, Baozhen Shan, and Ming Li. *De novo* peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31):8247–8252, 2017.

[12] Thilo Muth, Felix Hartkopf, Marc Vaudel, and Bernhard Y Renard. A potential golden age to come- current tools, recent use cases, and future avenues for *de novo* sequencing in proteomics. *Proteomics*, 18(18):1700150, 2018.

[13] Hao Yang, Hao Chi, Wen-Jing Zhou, Wen-Feng Zeng, Chao Liu, Rui-Min Wang, Zhao-Wei Wang, Xiu-Nan Niu, Zhen-Lin Chen, and Si-Min He. pSite: Amino acid confidence evaluation for quality control of *de novo* peptide sequencing and modification site localization. *Journal of Proteome Research*, 17(1):119–128, 2017.

[14] Ngoc Hieu Tran, Xianglilan Zhang, and Ming Li. Deep Omics. *Proteomics*, 18(2):1700319, 2018.

[15] Rui Qiao, Ngoc Hieu Tran, Ming Li, Lei Xin, Baozhen Shan, and Ali Ghodsi. DeepNovoV2: Better *de novo* peptide sequencing with deep learning. *arXiv preprint arXiv:1904.08514*, 2019.

[16] Ari M Frank. A ranking-based scoring function for peptide- spectrum matches. *Journal of Proteome Research*, 8(5):2241–2252, 2009.

[17] Bin Ma. Challenges in computational analysis of mass spectrometry data for proteomics. *Journal of Computer Science and Technology*, 25(1):107–123, 2010.

[18] Thomas Bäck, David B Fogel, and Zbigniew Michalewicz. *Evolutionary computation 1: Basic algorithms and operators*, volume 1. CRC press, 2000.

[19] Agoston E Eiben and Jim Smith. From evolutionary computation to the evolution of things. *Nature*, 521(7553):476, 2015.

[20] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms.* Oxford university press, 1996.

[21] Pedro G Espejo, Sebastián Ventura, and Francisco Herrera. A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 40(2):121–144, 2010.

[22] Joseph L Awange and Béla Paláncz. Symbolic regression. In *Geospatial Algebraic Computations*, pages 203–216. Springer, 2016.

[23] Alexandros Agapitos, Roisin Loughran, Miguel Nicolau, Simon Lucas, Michael O'Neill, and Anthony Brabazon. A survey of statistical machine learning elements in genetic programming. *IEEE Transactions on Evolutionary Computation*, 2019.

[24] Dipankar Dasgupta and Zbigniew Michalewicz. *Evolutionary algorithms in engineering applications.* Springer Science & Business Media, 2013.

[25] Dan Simon. *Evolutionary optimization algorithms.* John Wiley & Sons, 2013.

[26] Jiarui Ding, Jinhong Shi, Guy G Poirier, and Fang-Xiang Wu. A novel approach to denoising ion trap tandem mass spectra. *Proteome Science*, 7(1):9, 2009.

[27] Bernhard Y Renard, Marc Kirchner, Flavio Monigatti, Alexander R Ivanov, Juri Rappsilber, Dominic Winter, Judith AJ Steen, Fred A Hamprecht, and Hanno Steen. When less can yield more–computational preprocessing of MS/MS spectra for peptide identification. *Proteomics*, 9(21):4978–4984, 2009.

[28] James P Cleveland and John R Rose. Identification of b-/y-ions in MS/MS spectra using a two stage neural network. *Proteome Science*, 11(1):S4, 2013.

[29] Wenguang Shao and Henry Lam. Denoising peptide tandem mass spectra for spectral libraries: A bayesian approach. *Journal of Proteome Research*, 12(7):3223–3232, 2013.

[30] Urvesh Bhowan, Mark Johnston, Mengjie Zhang, and Xin Yao. Reusing genetic programming for ensemble selection in classification of unbalanced data. *IEEE Transactions on Evolutionary Computation*, 18(6):893–908, 2013.

[31] Muhammad Iqbal, Bing Xue, Harith Al-Sahaf, and Mengjie Zhang. Cross-domain reuse of extracted knowledge in genetic programming for image classification. *IEEE Transactions on Evolutionary Computation*, 21(4):569–587, 2017.

[32] Binh Tran, Bing Xue, and Mengjie Zhang. Genetic programming for multiple-feature construction on high-dimensional classification. *Pattern Recognition*, 93:404–417, 2019.

[33] Felipe Viegas, Leonardo Rocha, Marcos Gonçalves, Fernando Mourão, Giovanni Sá, Thiago Salles, Guilherme Andrade, and Isac Sandin. A genetic programming approach for feature selection in highly dimensional skewed data. *Neurocomputing*, 273:554–569, 2018.

[34] Divyaansh Devarriya, Cairo Gulati, Vidhi Mansharamani, Aditi Sakalle, and Arpit Bhardwaj. Unbalanced breast cancer data classification using novel fitness functions in genetic programming. *Expert Systems with Applications*, 140:112866, 2020.

[35] Urvesh Bhowan, Mark Johnston, Mengjie Zhang, and Xin Yao. Evolving diverse ensembles using genetic programming for classification with unbalanced data. *IEEE Transactions on Evolutionary Computation*, 17(3):368–386, 2013.

[36] Urvesh Bhowan, Mark Johnston, Mengjie Zhang, and Xin Yao. Evolving diverse ensembles using genetic programming for classification with unbalanced data. *IEEE Transactions on Evolutionary Computation*, 17(3):368–386, 2012.

[37] Bach Hoai Nguyen, Bing Xue, Peter Andreae, Hisao Ishibuchi, and Mengjie Zhang. Multiple reference points based decomposition for multi-objective feature selection in classification: Static and dynamic mechanisms. *IEEE Transactions on Evolutionary Computation*, 2019.

[38] Alexandre Sawczuk Da Silva, Hui Ma, Yi Mei, and Mengjie Zhang. A hybrid memetic approach for fully automated multi-objective web service composition. In *2018 IEEE International Conference on Web Services (ICWS)*, pages 26–33. IEEE, 2018.

[39] Jingrui Zhang, Qinghui Tang, Po Li, Daxiang Deng, and Yalin Chen. A modified MOEA/D approach to the solution of multi-objective optimal power flow problem. *Applied Soft Computing*, 47:494–514, 2016.

[40] Purva Goel, Sanket Bapat, Renu Vyas, Amruta Tambe, and Sanjeev S Tambe. Genetic programming based quantitative structure–retention relationships for the prediction of kovats retention indices. *Journal of Chromatography A*, 1420:98–109, 2015.

[41] Soha Ahmed, Mengjie Zhang, Lifeng Peng, and Bing Xue. *A Multi-objective Genetic Programming Biomarker Detection Approach in Mass Spectrometry Data*, volume 9597, pages 106–122. Springer International Publishing, 2016.

[42] Ngoc Hieu Tran, M Ziaur Rahman, Lin He, Lei Xin, Baozhen Shan, and Ming Li. Complete *de novo* assembly of monoclonal antibody sequences. *Scientific reports*, 6:31730, 2016.

[43] Thilo Muth and Bernhard Y Renard. Evaluating *de novo* sequencing in proteomics: already an accurate alternative to database-driven peptide identification? *Briefings in bioinformatics*, 19(5):954–970, 2017.

[44] Ari Frank and Pavel Pevzner. PepNovo: *de novo* peptide sequencing via probabilistic network modeling. *Analytical Chemistry*, 77(4):964–973, 2005.

[45] Michael Lund Nielsen. *Characterization of Polypeptides by Tandem Mass Spectrometry Using Complementary Fragmentation Techniques.* PhD thesis, Acta Universitatis Upsaliensis, 2006.

[46] Lijuan Mo, Debojyoti Dutta, Yunhu Wan, and Ting Chen. MSNovo: a dynamic programming algorithm for *de novo* peptide sequencing via tandem mass spectrometry. *Analytical Chemistry*, 79(13):4870–4878, 2007.

[47] Kyowon Jeong, Sangtae Kim, and Pavel A Pevzner. UniNovo: a universal tool for *de novo* peptide sequencing. *Bioinformatics*, 29(16):1953–1962, 2013.

[48] Hao Yang, Hao Chi, Wen-Jing Zhou, Wen-Feng Zeng, Kun He, Chao Liu, Rui-Xiang Sun, and Si-Min He. Open-pNovo: *de novo* peptide sequencing with thousands of protein modifications. *Journal of Proteome Research*, 16(2):645–654, 2017.

[49] Bobbie-Jo M Webb-Robertson and William R Cannon. Current trends in computational inference from mass spectrometry-based proteomics. *Briefings in Bioinformatics*, 8(5):304–317, 2007.

[50] Marina Spivak, Jason Weston, Léon Bottou, Lukas Käll, and William Stafford Noble. Improvements to the percolator algorithm for peptide identification from shotgun proteomics data sets. *Journal of Proteome Research*, 8(7):3737–3745, 2009.

[51] Changjiang Xu and Bin Ma. Complexity and scoring function of MS/MS peptide *de novo* sequencing. In *Comput. Syst. Bioinformatics Conf*, volume 5, pages 361–369, 2006.

[52] Matthew T Olson, Jonathan A Epstein, and Alfred L Yergey. *De novo* peptide sequencing using exhaustive enumeration of peptide composition. *Journal of the American Society for Mass Spectrometry*, 17(8):1041–1049, 2006.

[53] Christopher Hughes, Bin Ma, and Gilles A Lajoie. *De novo* sequencing methods in proteomics. In *Proteome Bioinformatics*, pages 105–121. Springer, 2010.

[54] Bernd Fischer, Volker Roth, Franz Roos, Jonas Grossmann, Sacha Baginsky, Peter Widmayer, Wilhelm Gruissem, and Joachim M Buhmann. NovoHMM: a hidden markov model for *de novo* peptide sequencing. *Analytical Chemistry*, 77(22):7265–7273, 2005.

[55] Joshua E Elias, Francis D Gibbons, Oliver D King, Frederick P Roth, and Steven P Gygi. Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nature Biotechnology*, 22(2):214–219, 2004.

[56] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[57] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[58] Chengjian Tu, Quanhu Sheng, Jun Li, Danjun Ma, Xiaomeng Shen, Xue Wang, Yu Shyr, Zhengping Yi, and Jun Qu. Optimization of search engines and postprocessing approaches to maximize peptide and protein identification for high-resolution mass data. *Journal of Proteome Research*, 14(11):4662–4673, 2015.

[59] Hao Chi, Haifeng Chen, Kun He, Long Wu, Bing Yang, Rui-Xiang Sun, Jianyun Liu, Wen-Feng Zeng, Chun-Qing Song, Si-Min He, et al. pNovo+: *de novo* peptide sequencing using complementary hcd and etd tandem mass spectra. *Journal of Proteome Research*, 12(2):615–625, 2012.

[60] Andrew Trotman. Learning to rank. *Information Retrieval*, 8(3):359–381, 2005.

[61] Vladan Babovic and Maarten Keijzer. Genetic programming as a model induction engine. *Journal of Hydroinformatics*, 2(1):35–60, 2000.

[62] Guido Smits and Mark Kotanchek. Pareto-front exploitation in symbolic regression. *Genetic programming theory and practice II*, pages 283–299, 2005.

[63] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

[64] Michael O'Neill and David Fagan. The elephant in the room: Towards the application of genetic programming to automatic programming. In *Genetic Programming Theory and Practice XVI*, pages 179–192. Springer, 2019.

[65] Shu-Heng Chen. *Genetic algorithms and genetic programming in computational finance*. Springer Science & Business Media, 2012.

[66] Viktor Manahov and Hanxiong Zhang. Forecasting financial markets using high-frequency trading data: Examination with strongly typed genetic programming. *International Journal of Electronic Commerce*, 23(1):12–32, 2019.

[67] Yi-Shian Lee and Lee-Ing Tong. Forecasting energy consumption using a grey model improved by incorporating genetic programming. *Energy Conversion and Management*, 52(1):147–152, 2011.

[68] Biranchi Panda, K Shankhwar, Akhil Garg, and MM Savalani. Evaluation of genetic programming-based models for simulating bead dimensions in wire and arc additive manufacturing. *Journal of Intelligent Manufacturing*, 30(2):809–820, 2019.

[69] Mark Harman, Yue Jia, Jens Krinke, William B Langdon, Justyna Petke, and Yuanyuan Zhang. Search based software engineering for software product line engineering: a survey and directions for future work. In *Proceedings of the 18th International Software Product Line Conference-Volume 1*, pages 5–18. ACM, 2014.

[70] Wolfgang Banzhaf. Some remarks on code evolution with genetic programming. In *Inspired by Nature*, pages 145–156. Springer, 2018.

[71] Alexander Lalejini and Charles Ofria. Tag-accessed memory for genetic programming. *Memory*, 101:0011, 2019.

[72] Jen-Yuan Yeh, Jung-Yi Lin, Hao-Ren Ke, and Wei-Pang Yang. Learning to rank for information retrieval using genetic programming. In *Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (LR4IR 2007)*, 2007.

[73] Jung Yi Lin, Jen-Yuan Yeh, and Chao Chung Liu. Learning to rank for information retrieval using layered multi-population genetic programming. In *2012 IEEE International Conference on Computational Intelligence and Cybernetics (CyberneticsCom)*, pages 45–49. IEEE, 2012.

[74] Ricardo Baeza-Yates, Alfredo Cuzzocrea, Domenico Crea, and Giovanni Lo Bianco. Learning ranking functions by genetic programming revisited. In *International Conference on Database and Expert Systems Applications*, pages 378–386. Springer, 2018.

[75] Ricardo Baeza-Yates, Alfredo Cuzzocrea, Domenico Crea, and Giovanni Lo Bianco. An effective and efficient algorithm for ranking web documents via genetic programming. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1065–1072. ACM, 2019.

[76] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[77] Bioinformatics Solutions Inc. PEAKS Studio (Bioinformatics Solutions Inc., Waterloo, ON, Canada), Version 8.0., 2016.

[78] Hans JCT Wessels, Tom G Bloemberg, Maurice van Dael, Ron Wehrens, Lutgarde Buydens, Lambert P van den Heuvel, and Jolein Gloerich. A comprehensive full factorial LC-MS/MS proteomics benchmark data set. *Proteomics*, 12(14):2276–2281, 2012.

[79] Pavel Sinitcyn, Jan Daniel Rudolph, and Jürgen Cox. Computational methods for understanding mass spectrometry–based shotgun proteomics data. *Annual Review of Biomedical Data Science*, 1:207–234, 2018.

[80] Matthias Mann and Ole N Jensen. Proteomic analysis of post-translational modifications. *Nature Biotechnology*, 21(3):255–261, 2003.

[81] Erik Ahrné, Markus Müller, and Frederique Lisacek. Unrestricted identification of modified proteins using MS/MS. *Proteomics*, 10(4):671–686, 2010.

[82] Elena Ossipova. *Methods for mass spectrometric proteome analysis*, volume 2008. 2008.

[83] Donald L Pavia, Gary M Lampman, George S Kriz, and James A Vyvyan. *Introduction to spectroscopy*. Cengage Learning, 2008.

[84] Fred W McLafferty. Tandem mass spectrometry. *Science*, 214(4518):280–287, 1981.

[85] J Throck Watson and O David Sparkman. *Introduction to mass spectrometry: instrumentation, applications, and strategies for data interpretation*. John Wiley & Sons, 2007.

[86] Milam SB Munson and F-H_ Field. Chemical ionization mass spectrometry. i. general introduction. *Journal of the American Chemical Society*, 88(12):2621–2630, 1966.

[87] Alison E Ashcroft. *Ionization methods in organic mass spectrometry*, volume 5. Royal Society of Chemistry, 1997.

[88] Christine H Chung, Shawn Levy, Pierre Chaurand, and David P Carbone. Genomics and proteomics: emerging technologies in clinical cancer research. *Critical Reviews in Oncology/Hematology*, 61(1):1–25, 2007.

[89] Raymond E March and John F Todd. *Quadrupole ion trap mass spectrometry*, volume 165. John Wiley & Sons, 2005.

[90] Ioannis A Papayannopoulos. The interpretation of collision-induced dissociation tandem mass spectra of peptides. *Mass Spectrometry Reviews*, 14(1):49–73, 1995.

[91] F Dubois, R Knochenmuss, R Zenobi, A Brunelle, C Deprun, and Y Le Beyec. A comparison between ion-to-photon and microchannel plate detectors. *Rapid Communications in Mass Spectrometry*, 13(9):786–791, 1999.

[92] Jimmy K Eng, Ashley L McCormack, and John R Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, 5(11):976–989, 1994.

[93] John S Cottrell and U London. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–3567, 1999.

[94] Karl R Clauser, Peter Baker, and Alma L Burlingame. Role of accurate mass measurement (±10 ppm) in protein identification strategies employing ms or MS/MS and database searching. *Analytical Chemistry*, 71(14):2871–2882, 1999.

[95] Robertson Craig and Ronald C Beavis. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, 2004.

[96] Jacques Colinge, Alexandre Masselot, Marc Giron, Thierry Dessingy, and Jérôme Magnin. Olav: Towards high-throughput tandem mass spectrometry data identification. *Proteomics*, 3(8):1454–1463, 2003.

[97] David Fenyö and Ronald C Beavis. A method for assessing the statistical significance of mass spectrometry-based protein identifications

using general scoring schemes. *Analytical Chemistry*, 75(4):768–774, 2003.

[98] Clement Chung. *Machine Learning Approaches to Refining Post-translational Modification Predictions and Protein Identifications from Tandem Mass Spectrometry*. PhD thesis, University of Toronto, 2012.

[99] J Mitchell Wells and Scott A McLuckey. Collision-induced dissociation (cid) of peptides and proteins. *Methods in Enzymology*, 402:148–185, 2005.

[100] Richard S Johnson and J Alex Taylor. Searching sequence databases via *de novo* peptide sequencing by tandem mass spectrometry. *Molecular Biotechnology*, 22(3):301–315, 2002.

[101] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[102] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

[103] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[104] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[105] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[106] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2009.

[107] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[108] Vladimir Vapnik. *The nature of statistical learning theory.* Springer science & business media, 2013.

[109] Christopher M Bishop. *Neural networks for pattern recognition.* Oxford university press, 1995.

[110] George AF Seber and Alan J Lee. *Linear regression analysis*, volume 936. John Wiley & Sons, 2012.

[111] Terence C Mills. The classical linear regression model. In *Analysing Economic Data*, pages 166–187. Springer, 2014.

[112] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[113] JW Davidson, Dragan A Savic, and Godfrey A Walters. Symbolic and numerical regression: experiments and applications. *Information Sciences*, 150(1):95–117, 2003.

[114] Jerome H Friedman. Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67, 1991.

[115] Qi Chen. Improving the generalisation of genetic programming for symbolic regression. 2018.

[116] Lawrence J Fogel. *Intelligence through simulated evolution: forty years of evolutionary programming.* John Wiley & Sons, Inc., 1999.

[117] Holland John. Holland, adaptation in natural and artificial systems, 1992.

[118] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies–a comprehensive introduction. *Natural computing*, 1(1):3–52, 2002.

[119] William B Langdon, Riccardo Poli, Nicholas F McPhee, and John R Koza. Genetic programming: An introduction and tutorial, with a survey of techniques and applications. In *Computational intelligence: A compendium*, pages 927–1028. Springer, 2008.

[120] Wolfgang Banzhaf, Peter Nordin, Robert E Keller, and Frank D Francone. Genetic programming—an introduction: On the automatic evolution of computer programs and its applications, dpunkt. verlag and morgan kaufmann publishers. *Inc., San Francisco, California*, 1998.

[121] Markus F Brameier and Wolfgang Banzhaf. *Linear genetic programming*. Springer Science & Business Media, 2007.

[122] Julian F Miller. An empirical study of the efficiency of learning boolean functions using a cartesian genetic programming approach. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*, pages 1135–1142. Morgan Kaufmann Publishers Inc., 1999.

[123] Michael O'Neil and Conor Ryan. Grammatical evolution. In *Grammatical evolution*, pages 33–47. Springer, 2003.

[124] Riccardo Poli, William B Langdon, Nicholas F McPhee, and John R Koza. *A field guide to genetic programming*. Lulu. com, 2008.

[125] John R Koza, David Andre, Forrest H-Bennett, and Martin A Keane. Genetic programming iii: Darwinian invention & problem solving. 1999.

[126] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[127] Anupam Trivedi, Dipti Srinivasan, Krishnendu Sanyal, and Abhiroop Ghosh. A survey of multiobjective evolutionary algorithms based

on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462, 2016.

[128] Vikas Palakonda and Rammohan Mallipeddi. Pareto dominance-based algorithms with ranking methods for many-objective optimization. *IEEE Access*, 5:11043–11053, 2017.

[129] Urvesh Bhowan, Mengjie Zhang, and Mark Johnston. Multi-objective genetic programming for classification with unbalanced data. In *Australasian Joint Conference on Artificial Intelligence*, pages 370–380. Springer, 2009.

[130] Eckart Zitzler. Spea2: Improving the performance of the strength pareto evolutionary algorithm. *Computer Engineering and Communication Networks Lab (TIK)*, 2001.

[131] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.

[132] Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[133] Yanyan Tan, Xue Lu, Yan Liu, Qiang Wang, and Huaxiang Zhang. Decomposition-based multiobjective optimization with invasive weed colonies. *Mathematical Problems in Engineering*, 2019, 2019.

[134] Xiaoliang Ma, Qingfu Zhang, Guangdong Tian, Junshan Yang, and Zexuan Zhu. On tchebycheff decomposition approaches for multiobjective evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 22(2):226–244, 2017.

[135] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[136] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30(1):79–82, 2005.

[137] Marc Gentzel, Thomas Köcher, Saravanan Ponnusamy, and Matthias Wilm. Preprocessing of tandem mass spectrometric data to support automatic protein identification. *Proteomics*, 3(8):1597–1610, 2003.

[138] Jingfen Zhang, Simin He, Charles X Ling, Xingjun Cao, Rong Zeng, and Wen Gao. Peakselect: preprocessing tandem mass spectra for better peptide identification. *Rapid Communications in Mass Spectrometry*, 22(8):1203–1212, 2008.

[139] David M Horn, Roman A Zubarev, and Fred W McLafferty. Automated reduction and interpretation of high resolution electrospray mass spectra of large molecules. *Journal of the American Society for Mass Spectrometry*, 11(4):320–332, 2000.

[140] Michael R Hoopmann, Gregory L Finney, and Michael J MacCoss. High speed data reduction, feature detection, and MS/MS spectrum quality assessment of shotgun proteomics datasets using high resolution mass spectrometry. *Analytical Chemistry*, 79(15):5620, 2007.

[141] Deukwoo Kwon, Marina Vannucci, Joon Jin Song, Jaesik Jeong, and Ruth M Pfeiffer. A novel wavelet-based thresholding method for the pre-processing of mass spectrometry data that accounts for heterogeneous noise. *Proteomics*, 8(15):3019–3029, 2008.

[142] Cong Zhou, Lucas D Bowler, and Jianfeng Feng. A machine learning approach to explore the spectra intensity pattern of peptides using tandem mass spectrometry data. *BMC Bioinformatics*, 9(1):325, 2008.

[143] Shivani Tiwary, Roie Levy, Petra Gutenbrunner, Favio Salinas Soto, Krishnan K Palaniappan, Laura Deming, Marc Berndl, Arthur Brant, Peter Cimermancic, and Jürgen Cox. High-quality MS/MS spectrum

prediction for data-dependent and data-independent acquisition data analysis. *Nature Methods*, page 1, 2019.

[144] Alexandros Kalousis, Julien Prados, Elton Rexhepaj, and Melanie Hilario. Feature extraction from mass spectra for classification. In *6th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 536–543, 2005.

[145] Hao Chi, Rui-Xiang Sun, Bing Yang, Chun-Qing Song, Le-Heng Wang, Chao Liu, Yan Fu, Zuo-Fei Yuan, Hai-Peng Wang, Si-Min He, et al. pnovo: *de novo* peptide sequencing and identification using hcd spectra. *Journal of Proteome Research*, 9(5):2713–2724, 2010.

[146] Siegfried Gessulat, Tobias Schmidt, Daniel Paul Zolg, Patroklos Samaras, Karsten Schnatbaum, Johannes Zerweck, Tobias Knaute, Julia Rechenberger, Bernard Delanghe, Andreas Huhmer, et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature Methods*, 16(6):509, 2019.

[147] Hannes L Röst. Deep learning adds an extra dimension to peptide fragmentation. *Nature Methods*, 16(6):469, 2019.

[148] Wenzhou Li, Li Ji, Jonathan Goya, Guanhong Tan, and Vicki H Wysocki. Sqid: an intensity-incorporated protein identification algorithm for tandem mass spectrometry. *Journal of Proteome Research*, 10(4):1593–1602, 2011.

[149] Chandrasegaran Narasimhan, David L Tabb, Nathan C VerBerkmoes, Melissa R Thompson, Robert L Hettich, and Edward C Uberbacher. Maspic: intensity-based tandem mass spectrometry scoring scheme that improves peptide identification at high confidence. *Analytical Chemistry*, 77(23):7581–7593, 2005.

[150] Jens Allmer. Algorithms for the *de novo* sequencing of peptides from tandem mass spectra. *Expert Review of Proteomics*, 8(5):645–657, 2011.

[151] Tsuneaki Sakurai, T Matsuo, Hideo Matsuda, and Ituso Katakuse. Paas 3: A computer program to determine probable sequence of peptides from mass spectrometric data. *Biological Mass Spectrometry*, 11(8):396–399, 1984.

[152] CW Hamm, WE Wilson, and DJ Harvan. Peptide sequencing program. *Computer Applications in the Biosciences: CABIOS*, 2(2):115–118, 1986.

[153] Roman Zubarev and Matthias Mann. On the proper use of mass accuracy in proteomics. *Molecular & Cellular Proteomics*, 6(3):377–381, 2007.

[154] Vineet Bafna and Nathan Edwards. On *de novo* interpretation of tandem mass spectra for peptide identification. In *Proceedings of the seventh annual international conference on Research in computational molecular biology*, pages 9–18. ACM, 2003.

[155] Vlado Dancik, Theresa A Addona, Karl R Clauser, James E Vath, and Pavel A Pevzner. *De novo* peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 6(3-4):327–342, 1999.

[156] Christian Bartels. Fast algorithm for peptide sequencing by mass spectroscopy. *Biological Mass Spectrometry*, 19(6):363–368, 1990.

[157] David L Tabb, Ze-Qiang Ma, Daniel B Martin, Amy-Joan L Ham, and Matthew C Chambers. DirecTag: accurate sequence tags from peptide MS/MS through statistical scoring. *Journal of Proteome Research*, 7(9):3838–3846, 2008.

[158] Patricia Hernandez, Robin Gras, Julien Frey, and Ron D Appel. Popitam: towards new heuristic strategies to improve protein identification from tandem mass spectrometry data. *Proteomics*, 3(6):870–878, 2003.

[159] Ting Chen, Ming-Yang Kao, Matthew Tepel, John Rush, and George M Church. A dynamic programming approach to *de novo* peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 8(3):325–337, 2001.

[160] Vlado Dančík, Theresa A Addona, Karl R Clauser, James E Vath, and Pavel A Pevzner. *De novo* peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 6(3-4):327–342, 1999.

[161] Jonas Grossmann, Franz F Roos, Mark Cieliebak, Zsuzsanna Lipták, Lucas K Mathis, Matthias Müller, Wilhelm Gruissem, and Sacha Baginsky. AUDENS: a tool for automated peptide *de novo* sequencing. *Journal of Proteome Research*, 4(5):1768–1774, 2005.

[162] Marshall Bern and David Goldberg. *De novo* analysis of peptide tandem mass spectra by spectral graph partitioning. *Journal of Computational Biology*, 13(2):364–378, 2006.

[163] Hao Chi, Haifeng Chen, Kun He, Long Wu, Bing Yang, Rui-Xiang Sun, Jianyun Liu, Wen-Feng Zeng, Chun-Qing Song, Si-Min He, et al. pNovo+: *de novo* peptide sequencing using complementary hcd and etd tandem mass spectra. *Journal of Proteome Research*, 12(2):615–625, 2012.

[164] Chuang Li, Tao Chen, Qiang He, Yunping Zhu, and Kenli Li. MRUniNovo: an efficient tool for *de novo* peptide sequencing utilizing the hadoop distributed computing framework. *Bioinformatics*, 33(6):944–946, 2016.

[165] Kira Vyatkina. *De novo* sequencing of top-down tandem mass spectra: A next step towards retrieving a complete protein sequence. *Proteomes*, 5(1):6, 2017.

[166] Andrew Keller, Alexey I Nesvizhskii, Eugene Kolker, and Ruedi Aebersold. Empirical statistical model to estimate the accuracy of peptide

identifications made by MS/MS and database search. *Analytical Chemistry*, 74(20):5383–5392, 2002.

[167] DC Anderson, Weiqun Li, Donald G Payan, and William Stafford Noble. A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and sequest scores. *Journal of Proteome Research*, 2(2):137–146, 2003.

[168] Lukas Käll, Jesse D Canterbury, Jason Weston, William Stafford Noble, and Michael J MacCoss. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature Methods*, 4(11):923–925, 2007.

[169] Hao Yang, Hao Chi, Wen-Feng Zeng, Wen-Jing Zhou, and Si-Min He. pNovo 3: precise *de novo* peptide sequencing using a learning-to-rank framework. *Bioinformatics*, 35(14):i183–i190, 2019.

[170] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[171] Soha Ahmed. Genetic programming for biomarker detection in classification of mass spectrometry data. 2015.

[172] David C Wedge, Simon J Gaskell, Simon J Hubbard, Douglas B Kell, King Wai Lau, and Claire Eyers. Peptide detectability following esi mass spectrometry: prediction using genetic programming. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 2219–2225. ACM, 2007.

[173] Soha Ahmed, Mengjie Zhang, and Lifeng Peng. Prediction of detectable peptides in ms data using genetic programming. In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pages 37–38. ACM, 2014.

[174] Soha Ahmed, Mengjie Zhang, and Lifeng Peng. Feature selection and classification of high dimensional mass spectrometry data: A genetic programming approach. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, pages 43–55. Springer, 2013.

[175] Piotr S Gromski, Yun Xu, Elon Correa, David I Ellis, Michael L Turner, and Royston Goodacre. A comparative investigation of modern feature selection and classification approaches for the analysis of mass spectrometry data. *Analytica Chimica Acta*, 829:1–8, 2014.

[176] John Koza, Forrest Bennett, and David Andre. Using programmatic motifs and genetic programming to classify protein sequences as to cellular location. In *Evolutionary Programming VII*, pages 437–447. Springer, 1998.

[177] David Lennartsson and Peter Nordin. A genetic programming method for the identification of signal peptides and prediction of their cleavage sites. *EURASIP Journal on Applied Signal Processing*, 2004:138–145, 2004.

[178] Abdolhossein Fathi and Rasool Sadeghi. A genetic programming method for feature mapping to improve prediction of hiv-1 protease cleavage site. *Applied Soft Computing*, 72:56–64, 2018.

[179] Viktoria Dorfer, Sergey Maltsev, Stephan Dreiseitl, Karl Mechtler, and Stephan M Winkler. A symbolic regression based scoring system improving peptide identifications for ms amanda. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1335–1341. ACM, 2015.

[180] Viktoria Dorfer, Peter Pichler, Thomas Stranzl, Johannes Stadlmann, Thomas Taus, Stephan Winkler, and Karl Mechtler. Ms amanda, a

universal identification algorithm optimized for high accuracy tandem mass spectra. *Journal of Proteome Research*, 13(8):3679–3684, 2014.

[181] David D Stranz and Leroy B Martin III. Derivation of peptide sequence from mass spectral data using the genetic algorithm. *Journal of Biomolecular Techniques*, 1998.

[182] Alejandro Heredia-Langner, William R Cannon, Kenneth D Jarman, and Kristin H Jarman. Sequence optimization as an alternative to *de novo* analysis of tandem mass spectrometry data. *Bioinformatics*, 20(14):2296–2304, 2004.

[183] Joël M Malard, Alejandro Heredia-Langner, Douglas J Baxter, Kristin H Jarman, and William R Cannon. Constrained *de novo* peptide identification via multi-objective optimization. In *Proceedings of the 18th International on Parallel and Distributed Processing Symposium*. IEEE, 2004.

[184] Joël M Malard, Alejandro Heredia-Langner, William R Cannon, R Mooney, and Douglas J Baxter. Peptide identification via constrained multi-objective optimization: Pareto-based genetic algorithms. *Concurrency and Computation: Practice and Experience*, 17(14):1687–1704, 2005.

[185] Michał Kistowski and Anna Gambin. Optimization algorithm for *de novo* analysis of tandem mass spectrometry data. *BioTechnologia. Journal of Biotechnology Computational Biology and Bionanotechnology*, 92(3), 2011.

[186] Soha Ahmed, Mengjie Zhang, Lifeng Peng, and Bing Xue. A multi-objective genetic programming biomarker detection approach in mass spectrometry data. In *European Conference on the Applications of Evolutionary Computation*, pages 106–122. Springer, 2016.

[187] Mohashin Pathan, Monisha Samuel, Shivakumar Keerthikumar, and Suresh Mathivanan. Unassigned MS/MS spectra: Who Am I? In *Proteome Bioinformatics*, pages 67–74. Springer, 2017.

[188] Soha Ahmed, Mengjie Zhang, and Lifeng Peng. Improving feature ranking for biomarker discovery in proteomics mass spectrometry data using genetic programming. *Connection Science*, 26(3):215–243, 2014.

[189] Bioinformatics Solutions Inc. `http://www.bioinfor.com/faq/`. Accessed: 2019-09-03.

[190] Richard L Beardsley Herrmann and Amye Hilderbrand. Peptide fragmentation overview. *Principles of Mass Spectrometry Applied to Biomolecules*, 10:279, 2006.

[191] David R White. Software review: the ecj toolkit. *Genetic Programming and Evolvable Machines*, 13(1):65–67, 2012.

[192] Anthony J Alberg, Ji Wan Park, Brant W Hager, Malcolm V Brock, and Marie Diener-West. The use of "overall accuracy" to evaluate the validity of screening or diagnostic tests. *Journal of General Internal Medicine*, 19(5p1):460–465, 2004.

[193] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[194] Yonghua Han, Bin Ma, and Kaizhong Zhang. SPIDER: protein identification from sequence tags with *de novo* sequencing error. *Journal of Bioinformatics and Computational Biology*, 3(03):697–716, 2005.

[195] Frederick R Blattner, Guy Plunkett, Craig A Bloch, Nicole T Perna, Valerie Burland, Monica Riley, Julio Collado-Vides, Jeremy D Glasner, Christopher K Rode, George F Mayhew, et al. The complete genome sequence of escherichia coli K-12. *Science*, 277(5331):1453–1462, 1997.

[196] Lewis Y Geer, Sanford P Markey, Jeffrey A Kowalak, Lukas Wagner, Ming Xu, Dawn M Maynard, Xiaoyu Yang, Wenyao Shi, and Stephen H Bryant. Open mass spectrometry search algorithm. *Journal of Proteome Research*, 3(5):958–964, 2004.

[197] Atiya Masood, Gang Chen, Yi Mei, and Mengjie Zhang. Reference point adaption method for genetic programming hyper-heuristic in many-objective job shop scheduling. In *Proceedings of the European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 116–131. Springer, apr 2018.

[198] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[199] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, 2012.

[200] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[201] Nery Riquelme, Christian Von Lücken, and Benjamin Baran. Performance metrics in multi-objective optimization. In *2015 Latin American Computing Conference (CLEI)*, pages 1–11. IEEE, 2015.

[202] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. Reference point specification in inverted generational distance for triangular linear pareto front. *IEEE Transactions on Evolutionary Computation*, 22(6):961–975, 2018.

[203] Sujoy Paul and Swagatam Das. Simultaneous feature selection and weighting–an evolutionary multi-objective optimization approach. *Pattern Recognition Letters*, 65:51–59, 2015.

[204] Agoston E Eiben, P-E Raue, and Zs Ruttkay. Genetic algorithms with multi-parent recombination. In *International Conference on Parallel Problem Solving from Nature*, pages 78–87. Springer, 1994.

[205] Sayak Roychowdhury, Theodore T Allen, and Nicholas B Allen. A genetic algorithm with an earliest due date encoding for scheduling automotive stamping operations. *Computers & Industrial Engineering*, 105:201–209, 2017.

[206] Anas Arram and Masri Ayob. A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems. *Computers & Industrial Engineering*, 133:267–274, 2019.

[207] Slawomir Koziel and Xin-She Yang. *Computational optimization, methods and algorithms*, volume 356. Springer, 2011.

[208] Jürgen Cox and Matthias Mann. Maxquant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, 26(12):1367–1372, 2008.