Learning to Rank Peptide-Spectrum Matches Using Genetic Programming

Samaneh Azari[1], Mengjie Zhang[1], Bing Xue[1], and Lifeng Peng[2]
[1]School of Engineering and Computer Science;
[2]Centre for Biodiscovery and School of Biological Sciences
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand,
Emails: (samaneh.azari, mengjie.zhang, bing.xue)@ecs.vuw.ac.nz and lifeng.peng@vuw.ac.nz

***Abstract*—The analysis of tandem mass spectrometry (MS/MS) proteomics data relies on automated methods that assign peptides to observed MS/MS spectra. Typically these methods return a list of candidate peptide-spectrum matches (PSMs), ranked according to a scoring function. Normally the highest-scoring candidate peptide is considered as the best match for each spectrum. However, these best matches do not necessary always indicate the true matches. Identifying a full-length correct peptide by peptide identification tools is crucial, and we do not want to assign a spectrum to the peptide which is not expressed in the given biological sample. Therefore in this paper, we present a new approach to improving the previous ordering/ranking of the PSMs, aiming at bringing the correct PSM for spectrum ahead of all the incorrect ones for the same spectrum. We develop a new method called GP-PSM-rank, which employs genetic programming (GP) to learn a ranking function by combining different feature functions that measure the quality of PSMs from different perspectives.**

We compare GP-PSM-rank with SVM-rank. The results show that GP-PSM-rank outperforms SVM-rank in terms of the number of identified peptides which are true matches. On a validation dataset with 120 spectra, the proposed method is used as the post processing step on the results of peptide identifications by two *de novo* sequencing algorithms. GP-PSM-rank improves the results of both *de novo* methods in terms of identifying the true matches.

***Index Terms*—Genetic Programming, ranking function, peptide-spectrum match, tandem mass spectrometry.**

I. Introduction

Proteins are principal parts of organisms and perform a vast array of functions inside cells. Proteins as macro-molecules can be digested by proteases into short peptide fragments. *Peptides* are generally considered to be short chains of amino acids (from 2 to 50 amino acids). 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. Proteins in their primary structure are made up of a long chain of amino acids linked together in a linear sequence and they can have 50 to 2000 amino acid residues.

The common method for identifying proteins and characterising their amino acid sequences in proteomics is to digest the proteins into peptides, analyse the peptides using mass spectrometry, assign the resulting mass spectra to peptides, and match the assigned peptides to proteins.

Mass spectrometry (MS) is currently the most commonly used technology in proteomic analysis for identifying proteins in complex biological samples. The mass spectrometer measures the masses within a sample by ionising the sample and sorting the ions based on their mass-to-charge ratio (m/z). This results in producing thousands mass spectra, where each mass spectrum is an intensity vs. m/z plot. These masses can be used later for identifying the proteins or peptides in the samples using peptide identification tools. Intensities indicate the abundance of ions or m/z values. As mentioned above, the spectra can be used for identification of the proteins or peptides in the samples using peptide identification tools.

Peptide identification using MS/MS is traditionally accomplished by two major approaches namely database search and *de novo* peptide sequencing [1]. Given a spectrum, a database search algorithm tries to compare the experimental spectrum with an *in silico* (computer simulated) digested protein database and match the spectrum with highly similar sequences in the database to produce a list of peptide spectrum matches. The top-scored candidate is selected as the best matched peptide being regarded as the identification result [2], [3]. However, database search algorithms are highly dependent on a reference protein database, therefore they cannot identify peptides and proteins not present in the database, therefore *de novo* sequencing algorithms are used to infer a spectrum'*s* sequence without the assistance of a sequence database [4]. They select pairs of peaks and labels them if their mass differences are within the tolerance ranges of the amino acid'*s* masses. The labelled peak pairs are joined together to make paths. The algorithms score the path and the high-scored path is considered as the best candidate peptide sequence.

However, the highest-scoring candidates in the results of peptide identification by *de novo* sequencing do not necessary always indicate the true matches. Therefore, it still remains challenging to control the false discovery rate (FDR) in *de novo* peptide sequencing algorithms. The precision of identifying full-length peptides by existing *de novo* sequencing algorithms cannot reach 70% even with identified high-scored PSMs [5]. The lack of a reference protein database results in such low accuracy in *de novo* sequencing methods. Therefore, after performing *de novo* peptide identification, a discriminating step (PSM validation algorithm) to distinguish true matches from many close false ones (such as homeometric peptides that are different peptides with similar theoretical MS/MS spectra) is essential in proteomic data analysis [6].

978-1-7281-2153-6/19/$31.00

The existing PSM-scoring functions to rank the PSMs have the following limitations:

- A number of scoring schemes (for example, TANDEM [7], OMSSA [8]) are based on the shared peak count (SPC), which is the number of peaks matched between experimental and theoretical spectra. However, in practice, SPC does not perform well since all shared peaks have equal weights, although some are more informative than others.
- There are other match scoring schemes such as simple dot-product [1], cross correlation score [2], [3] or more advanced statistical measures like the expectation value [9]. However, these scores cannot serve as the primary discriminating parameter for separating correct from incorrect identifications.
- PeptideProphet [10] built linear scoring functions using the combination of various weighted (sub)scores. The weights/ coefficients have been learnt using a genetic algorithm (GA) on a training set of already identified MS/MS spectra [11]. However, all the scoring scheme imposes a prior assumption (the linear combination of (sub) scores) and only tries to optimise the parameters (weights) for the pre-defined model structure.

However, none of the above scoring function on its own can work well enough on indicating if a PSM is correct or not. Therefore, we need a model to produce accurate prediction rules by combining these weak rules into a powerful discriminatory scoring function for PSMs. PSM scoring is quite similar to scoring query-document pairs in information retrieval (IR) where the degree of relevance of each document (the web page) to a user query is defined by a score. Practically, for effectively scoring the query-document pairs, a ranking function which defines an order among documents according to their degree of relevance to the user query, is generated.

GP has been successfully applied to automatically generate an effective ranking function for IR [12]. In such frameworks, during the training process, GP attempts to learn a ranking function and aims at optimising a performance measure (for example classification accuracy, error rate, etc.). As the potential of GP to automatically produce an effective ranking function for PSM ranking has not been investigated, it is worth discovering how GP builds a ranking function that can be used to improve the previous ordering of PSMs from *de novo* results.

A. Research Goals

The main goal of this paper is to generate an effective ranking function which will be used to re-rank a collection of candidate PSMs which are the output of *de novo* sequencing algorithms. GP is used to automatically produce a new ranking function which aims at improving rate of true matches from output of *de novo* sequencing methods. Specifically, the following objectives are investigated:

1) Design appropriate terminal sets by using a set of features to measure the quality of matches between the experimental spectra and the candidate peptides.
2) Design a fitness function to guide GP towards increasing the number of true matches during the learning process.
3) Evaluate the effectiveness of the new ranking function based on the improvement in identification rate of *de novo* sequencing algorithms.

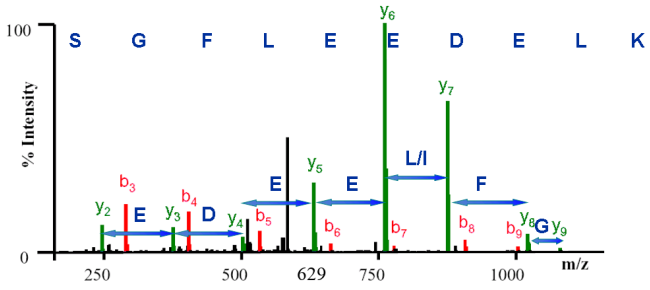

Fig. 1. *De novo* sequencing on an ideally fragmented MS/MS Spectrum 'SGFLEEDELK' with two b-/y-ions.

II. Background

A. Assigning MS/MS Spectra to Peptide Sequences

One of the suitable techniques for the identification of peptide sequences is Collision-induced dissociation (CID) [13]. In this technique, fragmentation happens at the peptide bonds, producing b-/y-ions. The amino acid sequence of an MS/MS spectrum can be determined by the mass differences between b-/y-ions. A *de novo* sequencing algorithm selects pairs of peaks and labels them if their mass differences are within the tolerant ranges of the amino acids masses. Fig. 1 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. 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. If the selected pairs of peaks are b-/y-ions, the correct peptide sequence is obtained. To match a spectrum against a peptide, a peptide is simulated to a theoretical spectrum which only has b-/y-ions. These ions are matched with the peaks in the experimental spectrum. The quality of the match is defined based on the number of matched and un-matched peaks between the two spectra.

B. Genetic Programming and MS/MS Data Analysis

Genetic Programming (GP) is an evolutionary algorithm that has been successfully applied to different kinds of real-world problems with complex search spaces [14], [15]. GP uses a variable-length individual representation to evolve a population of computer programs to automatically build or evolve a model to tackle the problem. GP randomly generates an initial population of individuals to search for the solution. During

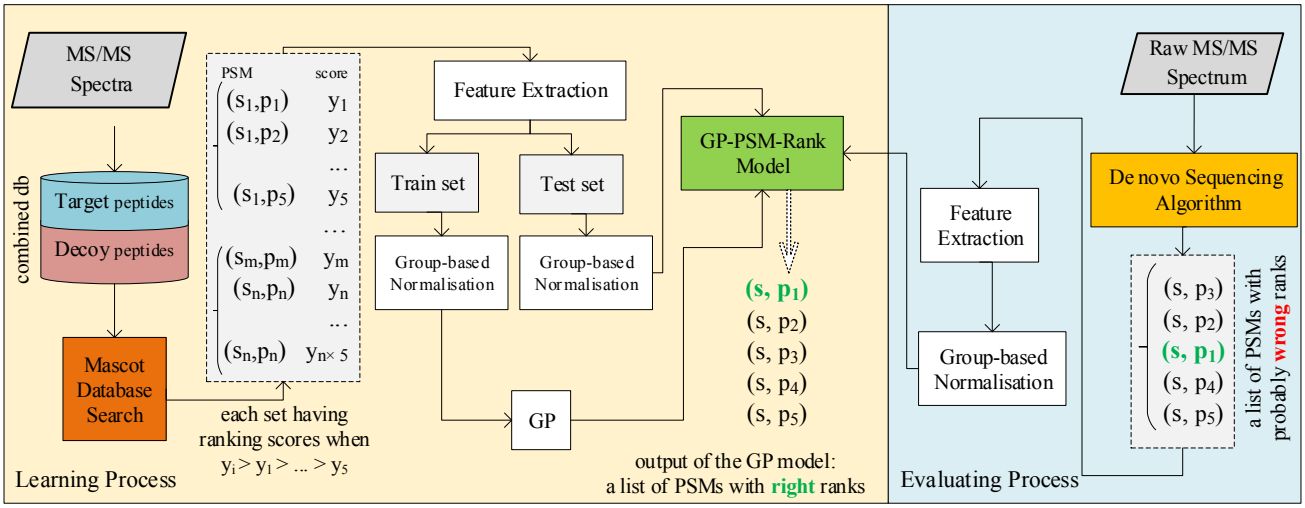

Fig. 2. The workflow of the proposed GP-PSM-rank method consisting of learning and evaluating phases.

the evolutionary search process, individuals are modified by the set of genetic operators [16] where the fitter solutions are more likely to be chosen to generate offsprings for the next generation of population. GP has shown a great potential to deal effectively with the challenges in MS and MS/MS data [17], [18]. It is worth investigating how GP can be used to build PSM-ranking functions.

III. The Proposed Ranking GP Method

Fig. 2 shows the proposed GP-PSM-rank workflow designed for re-ranking the PSMs. The workflow consists of two major parts: learning to rank PSMs by GP followed by an evaluating step to post-process the results of *de novo* sequencing using the new ranking function. Normally in the output of *de novo* sequencing algorithms true matches and other close false ones can be found among the top 5 candidate peptides for each spectrum. Therefore here for each spectrum, we consider five candidate peptides to cover the potential possibilities of finding the correct matches.

In our ground truth, which is a set of high confident PSMs from the results of peptide identification by Mascot database search tool, for each spectrum its corresponding peptide is provided. These sets of correct peptide-spectrum matches are called target PSMs. For each spectrum, we also need four incorrect peptides which are very close to the correct peptides and typically are responsible for most of the ranking errors in *de novo* sequencing algorithms. These incorrect peptides are produced by the decoy method where the correct peptides are randomly shuffled and creates a decoy sequence database. The existing database search tool, Mascot [3], is searched against the decoy sequences from the combined protein database. For each spectrum, we store four top-scoring decoy PSMs. Therefore, for each spectrum in the training set we have a group of five candidate peptides containing one target PSM and four decoy PSMs. The GP method needs to discriminate between the target and decoy PSMs in each group. GP is supposed to learn a ranking function that gives the highest score (the largest value, a real number) to the target PSM other than decoy PSMs in each group.

In summary, given a set of MS/MS spectra, S = $\{s_1, ..., s_n\}$, and a set of peptides, P = $\{p_1, ..., p_{n\times 5}\}$, the training set is created as a set of peptide-spectrum matches, each $(s_i, p_j) \in S \times P$, upon which a confidence score is assigned by the Mascot database search tool indicating the degree of reliability of match between the spectrum and the peptide to each PSM. For each instance, a set of features that describe the match between s_i and p_j, summarised in Table I, is extracted. A set of five PSMs (instances) belonging to the same spectrum is considered as one group where the instances in each group are sorted based on their Mascot score. Before applying GP to the training set, a group-based normalisation is applied on feature values and confidence scores of each group in order to normalise all values into a range of [0,1]. The inputs to the learning algorithm comprise training instances, i.e. the normalised feature vectors, and the corresponding Mascot confidence scores. The output is a ranking function, R, where $R(s_i, p_j)$ is supposed to associate a real number with (s_i, p_j) as its match score. For testing, the learned ranking function, the model is applied on new peptide p_i and a new spectrum s from the test set to determine their corresponding ranking score.

A. Feature Extraction

A set of 22 features shown in Table I are extracted from each PSM. The first feature, $Sim(s,p)$ which previously used in [19] as the fitness function of the GA algorithm, calculates the quality of match between the experimental spectrum s and theoretical (simulated) spectrum of peptide p

TABLE I
FEATURES USED TO REPRESENT PSMS.

	Feature name	Description
f_1	Sim score	Linear combination of different match scores
f_2	Int_Matched	sum of intensities of those peaks in the spectrum *s* which are matched with theoretical spectrum of peptide p
f_3	deltaMass	The mass difference between the spectrum *s* and peptide p
f_4	#NotMatched	# of not matched peaks in theoretical spectrum of peptide p
f_5	#Matched	# of matched peaks in theoretical spectrum of peptide p
f_6	Nterm	# of matched b-/y-ions from N-terminus (left to right) of peptide p
f_7	Cterm	# of matched b-/y-ions from C-terminus (right to left)
f_8	Cos	Fixed length Normalised Dot product
f_9	Euc	Fixed length normalised Euclidean distance
f_{10}	Hamming	Hamming distance between two vectorised
f_{11}	SeqVar	Variable length SEQUEST-like scoring function
f_{12}	SeqFix	Fixed length SEQUEST-like scoring function
f_{13}-f_{22}	Δscores	For each feature Cos, Euc, Hamming, SeqVar, and SeqFix two more features are computed. Each feature is calculated based on the fractional difference between the current PSM and the 2nd best and the 5th best PSM from the same group

using Equation (1).

$$Sim(s,p) = \frac{\sum I_{matched}}{\sum_{i=1}^{n} I_i} - \frac{|\Delta mass|}{M(s)} + \frac{Nterm + Cterm - \sum N_{not-matched}}{length(p)} \quad (1)$$

where $I_{matched}$ is the sum of intensities of those peaks in the experimental spectrum *s*, which are matched against the peaks in theoretical spectrum *t* corresponding to the peptide *p*. $I_{matched}$ is normalised by dividing it by the total intensities of spectrum *s*. Δmass is the mass difference between the spectrum *s* and peptide *p*, and is normalised by dividing it by the mass of spectrum *s*. The two terms $Nterm$ and $Cterm$ are the number of sequential b-ion matches from N-terminus (left to right) and from C-terminus of the theoretical spectrum *t*, respectively. $N_{not-matched}$ equals to the number of b-/y-ions in the theoretical spectrum *t*, which are not a match against the spectrum *s*. All the last three terms are normalised by dividing to the length of peptide *p*.

As each term in Equation (1) is normalised and all terms are linearly combined with equal weights of 1, it is worth considering each term as a seperate feature and let GP to find the non-linear relationship between them. Therefore, features $\{f_2, f_3, f_4, f_6, f_7\}$ are the non-normalised forms of each term in Equation (1). f_5 counts the number of matched peaks between two spectra.

Features $\{f_8, f_9, f_{10}, f_{11}, f_{12}\}$ each convert the experimental spectrum *s* and the theoretical spectrum *t* of peptide *p* into two binned vectors. SeqFix and SeqVar, (f_{11}, f_{12}), apply a preprocessing step which is inspired of SEQUEST, a benchmark database search engine [2]. Both features remove all the peaks in the 10-u window around the m/z-value of the precursor ion and then keep only the most intense 200 peaks on the spectrum. The intensities of peaks are normalised as follows. The whole spectrum is divided into ten intervals. For each interval, the most intense peak is set to an intensity of 50 and the intensity of the other peaks are divided by the maximum intensity in the interval and multiplied by 50. To vectorise the spectra, both the processed spectrum and the theoretical spectrum are split into *n* 1u-bins with *n* either a fix value or based on fragment ion tolerance. SeqVar, f_{11} , has a variable length which is determined based on dividing the precursor mass into fragment ion tolerance. Features $\{f_8, f_9, f_{10}, f_{12}\}$ have fix length of 4,000. Each bin in a vector of experimental spectrum is weighted as the sum of the intensities of the peaks within a corresponding bin. All bins for the vectorised theoretical spectrum have weights of one. The four features $\{f_8, f_{11}, f_{12}\}$ after converting the experimental and theoretical spectra into two vectors, use Equation (2) to calculate the normalised dot product between two vectors x, y.

$$cos\ \theta = \frac{x.y}{||x|| \times ||y||} \quad (2)$$

where x and y are the vectorised experimental and theoretical spectra, respectively.

The normalised dot product varies in a range of [0,1]. The output of Equation (2), indicates the matching between the two vectors/spectra. While $cos\ \theta = 0$ presents two orthogonal vectors, it indicates that two spectra have no peak matched between each other. On the other hand, $cos\ \theta = 1$ presents two identical vectors and indicates that every peak is matched between the experimental and the theoretical spectra. f_9 and f_{10} calculate the normalised Euclidean (based on Equation (3)) and hamming distance between the two binned spectra, respectively.

$$euc\ (x,y) = \frac{\sqrt{\Sigma(x_i - y_i)^2}}{||x|| \times ||y||} \quad (3)$$

For each feature value of $\{f_8, f_9, f_{10}, f_{11}, f_{12}\}$, the set of features $\{f_{13}, ..., f_{22}\}$ calculate the fractional difference between the current PSM and 2nd best and fifth best PSMs belonging to the same spectrum group.

B. Normalisation

Our learning set, L, is a group of PSMs, each with a vector of features describing the quality of match between the spectrum and peptide followed by a Mascot match score. For a set of MS/MS spectra, $S = \{s_1, ..., s_n\}$, and a set of peptides, $P = \{p_1, ..., p_{n\times5}\}$, a feature set, $F = \{f_1, ..., f_{|F|}\}$, and a set of Mascot scores corresponding to each (s_i, p_j), $Y = \{y_1, ..., y_{n\times5}\}$, the learning set L is formulated by Equation (4). The Mascot scores are used to sort the instances

in each group. So the top rank PSM (the first PSM in the group with the highest Mascot score) is the target PSM and the other ranks belong to the decoy sequences from the same group.

$$L = \{((s_i, p_j), (f_1(s_i, p_j), ..., f_{|F|}(s_i, p_j)), y_{ij})\} \quad (4)$$

After feature extraction and before applying the ranking function (here are GP and SVM), the learning set L is divided into two training and test sets. On each set a group-normalisation is applied to the feature values belonging to each group of PSMs in order to normalise all the values into a range of [0,1]. For a spectrum, and a peptide in (s_i, p_j), their corresponding feature value $f_k(s_i, p_j)$ is calculated based on Equation (5):

$$f_k(s_i, p_j) = \frac{f_k(s_i, p_j) - min\{f_k(s_i, p_l)\}}{max\{f_k(s_i, p_l)\} - min\{f_k(s_i, p_l)\}} \quad (5)$$

where $max\{f_k(s_i, p_l)\}$ and $min\{f_k(s_i, p_l)\}$ are the maximum and minimum value of $f_k(s_i, p_l)$ respectively for all $p_l \in P$. It is worth mentioning that calculating the max/min f_k of instances in the test set is completely separate from training set. As already the PSMs in the learning set are categorised into groups of five PSMs, the group normalisation calculates the min/max f_k based on the feature values of the PSMs belonging to the same group.

C. GP Program Representation

In GP-PSM-rank, an individual is a potential ranking function that assigns a real number to a spectrum and a peptide as their match score. A tree based GP structure is considered to represent each GP individual. The terminal set of the GP method includes the extracted features (S_f) and a set of predefined real numbers, S_c, ranging from 0 to 1 which are known as constants. The function set consists of at set of arithmetic operators (S_{arith}), where:

S_f = $\{f_k | f_k \in F, 1 \leq k \leq 22\}$

S_c = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}

S_{arith} = $\{+, -, \times, /$(protected division)$\}$

where F is the set of features in Table I. The protected division indicates usual division except that a division by zero gives a result of zero). Each arithmetic operator gets two arguments and returns one argument. For each instance, the output of the GP program is a single floating point number indicating the match score for PSM of (s_i, p_j). The following section, fitness function, explains the ranking strategy. Table II displays the GP parameters used in this work. GP-PSM-rank is implemented in Python 3.6 and uses DEAP (Distributed Evolutionary Algorithms in Python) package [20].

D. An Effective Fitness Function for PSMs scoring

The purpose of the GP method is to find a ranking function that gives real value scores to a collection of candidate PSMs and tries to maximise the number of target PSMs (true matches) which have the highest scores compared to the other decoy (incorrect) peptides corresponding to the same spectrum.

TABLE II
GENETIC PROGRAMMING PARAMETERS

Parameter	Value
Function Set	$\{+, -, \times, /$(protected)$\}$
Terminal Set	{Features from dataset, random Constants}
Initial Population	Ramped Half-and Half
Population Size	600
Generations	100
Mutation Rate	0.1
Elitism	The most fit ind. in each gen.
Crossover Rate	0.9
Selection	Tournament, Size = 5

For calculating the fitness score in the training set, the problem converts from a ranking task to a binary classification task, where in each group the target PSM is the positive instance and all the decoy PSMs are negative instances. After applying the GP ranking function, for each group of PSMs, if the target PSM gets the highest score, it is considered as a true positive (TP) or a hit and other PSMs are true negatives (TN). On the other hand, if the target PSM does not get the highest score among the other candidates in the same group, then it is considered as a false negative (FN) and the decoy PSM which gets the highest score is a false positive (FP) and other decoys are TNs.

In our GP method, we do not care about correctly ranking the decoys. Therefore, our fitness function only considers correctly ranking target PSMs. If the target PSM in each group gets a score higher than other PSMs in the same group, then the fitness function increases the number of TPs by one. Equation (6) shows the fitness function used for GP to measure the performance of the ranking function, i.e. GP individuals.

$$\text{true match}_{rate} = \frac{\sum_{i=1}^{N} hit(i)}{N} \quad (6)$$

$$hit = \begin{cases} 1, & \text{if } \textit{score} \text{ (target PSM)} > \textit{score} \text{ (decoy PSMs)} \\ 0, & \text{otherwise} \end{cases}$$

where N is the total number of MS/MS spectra, and sigma hit counts the number of TPs. The *score* function is the output of GP (a real number) for the given PSM. Therefore, true match$_{rate}$ is the rate of the total number of first-ranked target PSMs divided by total number of MS/MS spectra.

IV. Experiment Design

A. MS/MS Datasets

The MS/MS used in this work are selected from the comprehensive full factorial LC-MS/MS benchmark dataset. This dataset is particularly designed for evaluating MS/MS analysis tools and contains 50 protein samples extracted from Escherichia coli K12 [21]. The MS/MS spectra in this dataset were acquired from the linear ion trap Fourier-transform (LTQ-FT, Thermo Fisher Scientific) with the collision-induced dissociation (CID) technique and they have been already searched

TABLE III
THE MS/MS SPECTRA USED IN THIS STUDY.

dataset	# of Spectra	# of Target PSMs	# of Decoy PSMs	Total # of PSMs
Learning set	1,000	1,000	4,000	5,000
Validation set	120	120	480	600

against a curated Refseq [22] release 33 *Escherichia coli* (strain K12) database by using Mascot v2.2 [3]. As reported in [21], the dataset has the following parameters: the minimum precursor mass used in this dataset is 350 Dalton (Da) with maximum tryptic cleavage of one. The MS tolerance and MS/MS tolerance are 10 ppm and 0.8 Da, respectively. The cutoff q-value of 0.01 was considered to validate the results of database search. Since a peptide's charge and precursor mass can greatly influence the nature of its observed spectrum, from the peptide identification results provided by this dataset, a set of 1,000 doubly charged unique peptide-spectrum matches (PSMs) with maximum precursor mass of 1150 Da, peptide length between 7 and 12, and minimum Mascot peptide identification scores of 45 to ensure a high confident peptide identification were selected.

The learning set is composed of 1,000 spectra with known correct identifications of 1,000 target PSMs and known incorrect matches of 4,000 decoy PSMs. The total number of 5,000 PSMs in the learning set are divided into 1,000 groups, where for each group the target PSM has the highest Mascot score compared to other four decoy PSMs in the same group.

Also from full factorial dataset, a set of 120 MS/MS spectra which previously were used to evaluate GA-Novo [19], a *de novo* sequencing method by genetic algorithm, and PEAKS [4], the most common *de novo* sequencing software, is used to evaluate the effectiveness of GP-PSM-rank in terms of improving the peptide identification rates of these two *de novo* sequencing algorithms.

The set of 120 spectra in the validation set is given to these *de novo* sequencing tools. From the *de novo* results, for each spectrum, the top 5 candidate PSMs is taken as the results of *de novo* sequencing. As the *de novo* sequencing algorithm reports a confidence score for each PSM, the PSMs belonging to the same group are ordered based on their scores. Then, all PSMs (here for 120 MS/MS spectra, there are $120 \times 5 = 600$ PSMs) are given to GP-PSM-rank and SVM-rank models. They re-rank the PSMs, aiming at putting the correct peptide (target PSM) at first rank for each group of PSMs belonging to the same spectrum. Table III provides more details about the datasets.

B. Benchmark Algorithm

As GP is used to learn a ranking function, the proposed method is compared with SVM^{rank}, which is a benchmark algorithm in ranking tasks [23]. SVM-rank employs a support vector machine (SVM) to classify object pairs in consideration of large margin rank boundaries.

SVM^{rank} is free and consisting of a learning module and a module for making predictions. The training and test files should be prepared in a specific order where feature/value pairs must be ordered by increasing feature number. More details about SVM^{rank} can be found in [23].

To evaluate the effectiveness of the proposed GP method and comparing it with SVM-rank, the output of two *de novo* sequencing algorithms, PEAKS [4] and GA-Novo [19] are used to be re-ranked by GP-PSM-rank and SVM-rank, separately. Given an MS/MS spectrum to any of these *de novo* sequencing algorithms, the output is a set of peptide sequences each having a confidence score between 0 and 100 [4]. The scores indicate the reliability of identifications. For each spectrum, the top five score sequences are taken as the output of the *de novo* sequencing. Both GP-PSM-rank and SVM-rank get the list of candidate peptides, re-rank and return new confidence scores for each PSM.

C. Experiments

1) Experiment I: Learning the Rank Function: This experiment lets GP and SVM-rank to build and evaluate the ranking functions using the learning set in Table III. The 1,000 groups of PSMs in the data set are split into two sets of training and test each having 70% and 30% of PSMs, respectively. Based on the flowchart in Figure 2, the group-based normalisation is applied separately on both training and test sets. GP and SVM-rank use the training set to learn the ranking function and apply the models on the test set to evaluate the models based on Equation (6).

2) Experiment II: Evaluating the Effectiveness of GP-PSM-rank and SVM-rank: This experiment investigates the effectiveness of both ranking functions in terms of increasing the identification rate of target PSMs and minimising the missed identified target PSMs rate of the results of peptide identifications by two *de novo* sequencing methods using the validation set from Table III. The results of identifications are measured based on Equation 6 and Equation 7 before and after the post-processing.

$$\text{missed target PSMs}_{rate} = \frac{\text{FN}}{\text{total number of MS/MS spectra}} \quad (7)$$

where missed target PSMs$_{rate}$ is the rate of total number of those target PSMs that are not first-ranked divided by total number of MS/MS spectra.

V. RESULTS AND DISCUSSIONS

A. Results of Experiment I:

Table IV presents the results of the proposed GP method and SVM-rank in terms of true match rate (i.e. Equation (6)). For GP, the experiments are repeated for 30 individual runs using 30 different random seeds.

To compare the results of GP in 30 independent runs with SVM-rank, one sample statistical t-test with 95% confidence interval is used. (+) in the Table IV indicates the difference between the results of GP and SVM-rank is considered to be statistically significant. The results show that in average,

TABLE IV
THE RESULTS OF 30 INDEPENDENT RUNS OF GP METHOD AND SVM-RANK ON TRAINING AND TEST SET OF LEARNING SET CONTAINING 1,000 MS/MS SPECTRA IN TERMS OF IDENTIFIED TARGET PSMS RATE.

Algorithm	train	test
GP-PSM-rank	**0.83 ± 0.01 (+)**	**0.76 ± 0.02 (+)**
SVM-rank	0.74	0.70

GP outperformed SVM-rank by almost 9% on the training set and 6% on the test set of the learning set. One possible reason of successful performance of GP is its ability to learn from a relatively small training set. While success of most of the other ranking algorithms could possibly rely on learning from a large dataset.

B. Results of Experiment II:

To investigate the effectiveness of the ranking functions generated by GP and SVM, the best GP program among the 30 independent runs is selected as the post-processing method. The best GP ranking function is selected based on the best performance of GP on training set of the learning set. Both GP and SVM-rank models separately are applied on the results of PEAKS and GA-Novo using the validation set. As already the ground truth of the validation set is available, the target PSM identification rates before and after re-ranking the PSMs by both methods are calculated and presented in Table V. From this table it can be seen that both methods improve the identification rate of both *de novo* sequencing methods in terms of increasing the identification of target PSMs. However, GP outperformed SVM-rank by 5% and 3% increase in true match rate on the results of GA-Novo and PEAKS, respectively. Therefore, on the results of GA-Novo, the GP ranking function increased the true match$_{rate}$ by 10% and decreased the missed target PSMs$_{rate}$ by 10%. Also the best GP ranking function, improved the results of PEAKS by 15% and 15% in terms of increasing true match rate and decreasing the rate of those target PSMs not locating as first-ranked, respectively.

As the ground truth for the set of spectra in validation set is available, we further analysed the candidate peptide lists generated by the *de novo* sequencing algorithms to discover why after post processing a high target PSMs identification of 99% was not achieved. The analysis shows that for almost 10-15% of spectra, none of the 5 peptide sequences produced by the *de novo* sequencing method was correct. Basically in the dataset 3 situations may happen to each group of peptides belonging to the same spectrum: (1) the target PSM is the 1st-ranked, (2) the target PSM is not 1st-ranked, (3) the target PSM is not included in the peptide candidate list. Therefore, 99% of target PSMs identification was not achieved on the evaluating dataset, because the target PSMs of 10% to 15% of MS/MS spectra did not exist in the list of *de novo* candidate peptides. So this makes sense if the GP ranking function could not find them.

TABLE V
THE RESULTS OF PEPTIDE IDENTIFICATION RATES BEFORE AND AFTER POST-PROCESSING BY SVM-RANK AND THE BEST GP-PSM-RANK MODEL ON VALIDATION SET.

		true match rate		missed target PSMs rate	
De novo Method	Post-processing Model	before	after	before	after
GA-Novo	**GP-PSM-rank**	0.7	**0.8**	0.2	**0.1**
	SVM-rank		0.75		0.15
PEAKS	**GP-PSM-rank**	0.56	**0.71**	0.26	**0.11**
	SVM-rank		0.68		0.14

TABLE VI
THE AVERAGE RESULTS OF PEPTIDE IDENTIFICATION RATES BEFORE AND AFTER POST-PROCESSING BY GP-PSM-RANK FROM 30 INDEPENDENT RUNS ON VALIDATION SET CONTAINING 120 DOUBLY CHARGED MS/MS SPECTRA.

	true match$_{rate}$		missed target PSMs$_{rate}$	
Algorithm	before	after	before	after
GA-Novo	0.7	**0.75 ± 0.03**	0.2	**0.15 ± 0.03**
PEAKS	0.56	**0.66 ± 0.04**	0.26	**0.16 ± 0.04**

Table VI presents the average results of peptide identifications before and after refining the PSM ranks by the 30 GP ranking functions from 30 independent runs of GP. It can be seen that on average GP improve the target PSMs and missed target PSMs identification rates of *de novo* outputs.

C. Analysis on the best GP evolved program

Fig. 3 shows the best GP-evolved program among the 30 independent runs. The Tree includes 59 nodes. As GP performing implicit feature selection, out of the 22 given features, GP selected 12 informative features. The features $\{f_3, f_8, f_{10}, f_{14}, f_{15}, f_{16}, f_{18}, f_{19}, f_{20}, f_{21}\}$ are not used by the best evolved GP program. In this GP program, GP did not select the redundant features. For example the feature Hamming distance f_{10} is discarded and instead Euclidean distance f_9 is selected but f_8, cosine similarity, is not selected, ΔCos, f_{13}, seems to be more informative to GP and is selected.

GP has found a non-linear relationship between the features $\{f_1, f_2, f_5, f_7\}$ at its the most left sub-tree. It is worth investigating how this combination can improve the results of peptide identifications by GA-Novo if it is used as the fitness function of GA.

VI. CONCLUSIONS AND FUTURE WORK

This paper developed a method using GP to automatically produce an effective ranking function for PSM ranking, which combines different types of match scores that each measures the quality of match from different perspective. An effective fitness function was proposed to help GP to rank a collection of candidate PSMs, aiming at maximising the number of target PSMs identified at the top rank in each group of candidates for each spectrum. The results show that GP-PSM-rank outperformed SVM-rank by 9% and 6% in terms of

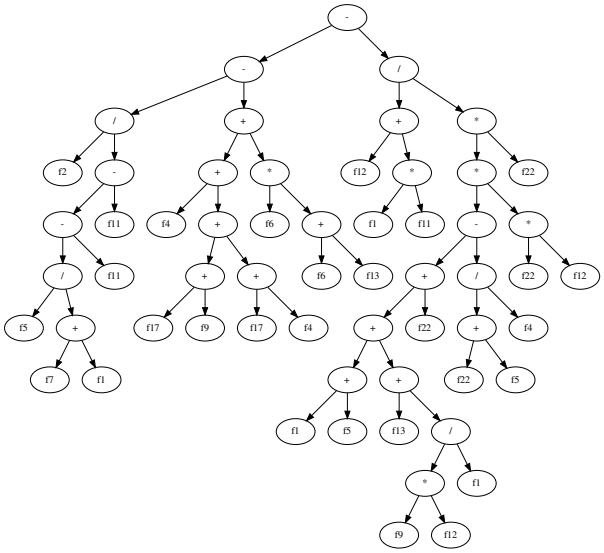

Fig. 3. The best GP evolved program (the PSM ranking function).

correctly identifying target PSMs on a training set and a test set containing 700 and 300 spectra, respectively.

On a test set of 120 spectra, our method was effective for two *de novo* sequencing algorithms including GA-Novo and PEAKS. When it was used to re-rank PSMs, the results of both methods significantly were improved in terms of true match$_{rate}$ and missed target PSMs$_{rate}$. The best evolved GP ranking function improved the results of peptide identification by GA-Novo and PEAKS in terms of true match rate by 10% and 15%, respectively. The best GP ranking function also decreased the rate of target PSMs that were ranked wrongly the *de novo* sequencing tools by 10% and 15% for GA-Novo and PEAKS, respectively.

The analysis of the best GP ranking functions revealed the important features for the task of ranking PSMs. The best evolved GP program only used 12 features out of the 22 available features in the datasets.

As for future work, we will investigate generating a generic model or a set of individual models that can handle different spectra with different charge numbers and precursor masses. We will also investigate more effective function sets for the GP method. The error of mis-ordering decoy PSMs will be also taken into consideration to come up with more accurate PSM ranking models.

REFERENCES

[1] Fengchao Yu, Ning Li, and Weichuan Yu. Pipi: Ptm-invariant peptide identification using coding method. *bioRxiv*, page 055806, 2016.

[2] 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.

[3] John S Cottrell and U London. Probability-based protein identification by searching sequence databases using mass spectrometry data. *electrophoresis*, 20(18):3551–3567, 1999.

[4] 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.

[5] 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.

[6] Alexey I Nesvizhskii, Olga Vitek, and Ruedi Aebersold. Analysis and validation of proteomic data generated by tandem mass spectrometry. *Nature methods*, 4(10):787, 2007.

[7] Robertson Craig and Ronald C Beavis. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, 2004.

[8] 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.

[9] 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.

[10] 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.

[11] Robin Gras, Markus Müller, Elisabeth Gasteiger, Steven Gay, Pierre-Alain Binz, William Bienvenut, Christine Hoogland, Jean-Charles Sanchez, Amos Bairoch, Denis F Hochstrasser, et al. Improving protein identification from peptide mass fingerprinting through a parameterized multi-level scoring algorithm and an optimized peak detection. *Electrophoresis*, 20(18):3535–3550, 1999.

[12] 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.

[13] Ioannis A Papayannopoulos. The interpretation of collision-induced dissociation tandem mass spectra of peptides. *Mass Spectrometry Reviews*, 14(1):49–73, 1995.

[14] Michael L Raymer, William F Punch, Erik D Goodman, and Leslie A Kuhn. Genetic programming for improved data mining: application to the biochemistry of protein interactions. In *Proceedings of the 1st annual conference on genetic programming*, pages 375–380. MIT Press, 1996.

[15] Richard A Davis, Adrian J Charlton, Sarah Oehlschlager, and Julie C Wilson. Novel feature selection method for genetic programming using metabolomic 1 h nmr data. *Chemometrics and Intelligent Laboratory Systems*, 81(1):50–59, 2006.

[16] 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.

[17] Soha Ahmed, Mengjie Zhang, and Lifeng Peng. Genetic programming for biomarker detection in mass spectrometry data. In *Australasian Joint Conference on Artificial Intelligence*, pages 266–278. Springer, 2012.

[18] Richard J Gilbert, Royston Goodacre, Andrew M Woodward, and Douglas B Kell. Genetic programming: A novel method for the quantitative analysis of pyrolysis mass spectral data. *Analytical Chemistry*, 69(21):4381–4389, 1997.

[19] S. Azari, X. Bing, M. Zhang, and L. Peng. GA-Novo: *De Novo* Peptide Sequencing via Tandem Mass Spectrometry using Genetic Algorithm. arXiv:1902.00845.

[20] 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.

[21] 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.

[22] Donna R Maglott, Kenneth S Katz, Hugues Sicotte, and Kim D Pruitt. Ncbis locuslink and refseq. *Nucleic acids research*, 28(1):126–128, 2000.

[23] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.