Discovery of New Features for Peptide Sequencing with Mass Spectrometry

by

Tiancong Wang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2017

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Bioinformaticians have been working on peptide sequencing with tandem mass spectrometry (MS/MS) for decades. However, the results are still not perfect. A lot of research have been carried on two peptide sequencing methods, database search and de novo sequencing. However, due to the quality of spectra and the inherent difficulty of this problem itself, both methods are having problem improving their results further better.

The publishing of the NIST peptide library in May 2014 brought fresh ideas into this long lasting problem. This peptide library contains a large amount of MS/MS spectra and their corresponding peptide sequences. Taking advantage of this high-quality dataset, more and more researches have started to find internal patterns in MS/MS spectra since then.

In this thesis, we are going to look more into this peptide library and use statistical and machine learning ideas to find new features to help improve peptide sequencing results. Two main contributions have been made.

First, a general scoring feature is presented that can be incorporated in the scoring functions of other peptide sequencing software. The scoring feature is based on the intensity ratios between two adjacent y-ions in the spectrum. A method is proposed to obtain the probability distributions of such ratios, and to calculate the scoring feature based on the distributions. To demonstrate the performance of the method, this new feature is incorporated with X!Tandem [1][2] and Novor [3] and significantly improved their performances on testing data, respectively.

Second, a machine learning model to predict the appearances of internal fragment ions in MS/MS spectra is presented. Even though this is the first model on this topic to the best of our knowledge, it achieves fairly good results. Several possible applications of this model are also discussed to show that this topic is valuable for peptide sequencing and thus worth further research.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Bin Ma. You are my supervisor not only on bioinformatics but also on my life. You gave me a lot of advice from how to debug to how to program in multi-thread, from how to make life decisions to even how to date girls. I have to say the luckiest thing in my life is having you as my supervisor.

I would also like to thanks my committee members, Dr. Lila Kari and Dr. Brendan McConkey. Your knowledge and insights in this area are very valuable and greatly enriched my work.

Also, I would like to thank Dr. Qixin Liu and Rapid Novor Inc. for providing me the data for experiments in section 5. Without your kind help, those experiments will never be carried out this easily.

Besides, I thank my fellow labmates in Bioinformatics Group: Chenyu Yao, Jianqiao Shen, Lian Yang, Rong Wang, and Qi Tang, for the heated discussions, and all the fun we have had in the last two years.

Last but not the least, I would like to thank my parents. You gave birth to me, you gave support to me, you gave everything you have to me. Hope I didn't let you down.

Dedication

This is dedicated to the my grandfather. You never left, you are always in my dreams.

Table of Contents

List of Tables viii

List of Figures ix

1 Introduction 1

1.1 Motivation . . . 1

1.2 Research Objectives and Contributions . . . 2

1.3 Thesis Overview . . . 3

2 Background 4

2.1 Protein, Peptide and Amino Acid . . . 4

2.2 Mass Spectrometry . . . 6

2.3 Y-ions and B-ions . . . 8

2.4 Peptide Sequencing . . . 10

2.5 Target Decoy and Decoy Fusion . . . 11

3 Related Works 14

3.1 X!Tandem . . . 14

3.2 Percolator . . . 15

3.3 Novor . . . 15

3.4 Spectrum Prediction . . . 17

4 Adjacent Y-ion Ratio Score Function **19**
4.1 Introduction . . . 19
4.2 Adjacent Y-ion Ratio Distributions . . . 21
4.2.1 Distributions Generation . . . 21
4.2.2 Different Distributions Comparison . . . 24
4.3 Y-ion Ratio Score Function . . . 24
4.3.1 Score Function . . . 26
4.3.2 X!Tandem Optimization . . . 27
4.3.3 Novor Optimization . . . 30
4.4 Experiments Results . . . 34
4.4.1 X!Tandem Optimization . . . 34
4.4.2 Novor Optimization . . . 37
4.5 Further revised model for Novor . . . 37

5 Internal Fragment Ions Prediction **45**
5.1 Introduction . . . 45
5.2 Internal Fragment Ions Statistics . . . 46
5.3 Prediction Model . . . 46
5.4 Experiment Results and Important Features . . . 48
5.4.1 Experiment Results . . . 48
5.4.2 Important Features . . . 48
5.5 Applications and Discussions . . . 50

6 Conclusions and Future Works **52**
6.1 Conclusions . . . 52
6.2 Future Works . . . 53

References **54**

List of Tables

2.1 Standard Amino Acids Table. 5

4.1 Features in Novor Optimization Model. 31
4.2 Percolator results without each single feature. 36
4.3 Features in Further Revised Novor Optimization Model. 42

5.1 Features in Internal Fragment Ions Prediction Model. 47

List of Figures

2.1 Basic instructions of a mass spectrometer (orbitrap). 6
2.2 A Sample of mass spectrum. 7
2.3 Y-ion series for example peptide MNSLQTDDTAK. 8
2.4 B-ion series for example peptide MNSLQTDDTAK. 9

3.1 A small portion of decision tree learned by Novor [3]. 16
3.2 MS-Simulator's work flow to generate a theoretical spectrum. [4]. 18

4.1 Y-ions for an example peptide. 21
4.2 Frequency of all data points and the distribution after smooth for 4-mer VPDL. 23
4.3 Different distributions for different 4-mers. 25
4.4 Different 0 and 90 degree probabilities different 4-mers. 25
4.5 X!Tandem score vs. y-ion ratio score scatter plot. 29
4.6 Optimization results for dataset C. elegans. 38
4.7 Optimization results for dataset Ubiquitin. 39
4.8 Optimization results for dataset UPS2. 40
4.9 Optimization results for dataset U2OS. 41
4.10 Optimization results with further revised Novor optimization model for dataset thermo. 44

5.1 Internal fragment ion prediction model precision-recall curve. 49

Chapter 1

Introduction

1.1 Motivation

Peptide sequencing from tandem mass spectrometry (MS/MS) data is a central task in proteomics. It is a problem to determine the amino acid sequence of a peptide. The accuracy and sensitivity of this task directly impacts the performance of protein identification, as well as other downstream protein analysis.

Two of the most common computational approaches for peptide sequencing are de novo sequencing [5][6][3][7] and database search [8][9][10][11][12]. De novo sequencing derives the peptide sequence directly from the MS/MS spectrum, whereas a database search queries a sequence database for the best peptide to explain the peaks in a MS/MS spectrum. Researchers have been working on these two methods for decades. However, due to the difficulty of the problem itself and the limitation of experiment equipments, peptide sequencing results could still be improved .

Recently, more and more novel approaches came up to look for a breakthrough. Some of them are following the hot topic, machine learning. One of the first efforts in this direction is the Percolator program[13]. It collects peptide spectrum matches (PSMs) outputted from database search softwares such as Mascot[9], SEQUEST[8] and X!Tandem[1][2] along with some additional information and then use a semi-supervised learning model (support vector machine model, to be more specific) to re-rank all those PSMs. Also for de novo sequencing, Frank et al. used a logistic regression model to combine several features together to estimate the correctness of PepNovo's de novo sequencing results [6].

One reason why peptide sequencing results are not perfect is because, due to the limitation of experiment equipments, spectra to be sequenced can never contain every necessary

information without any noises. This means there will always be some information missing for peptide sequencing software. However, because of the lack of quantitative understanding of the complex peptide fragmentation process in MS/MS, there are also a lot of other valuable information that are not being utilized. These information may also be helpful to improve peptide sequencing results. Sun et al. took a step in this area predicting the relateive intensity ratio of adjacent y-ions peaks in MS-Simulator software [4]. However, not many researches follow up on finding more such information due to the lack of high-quality data.

The publication of NIST peptide library in May 2014 brought new insight into this area [14]. This library contains a large amount of high-quality peptide spectrum matches (PSMs). More and more researchers are moving forward on finding internal patterns of MS/MS spectra taking advantage of this dataset. Novor was the first software to make full use of this library. It used this library to train a decision tree model combining 169 features, some of which are newly found, to give confidence scores to sequenced amino acids [3].

This peptide library provides us with a lot of valuable information of MS/MS spectra. Peptide sequencing tools can benefit a lot if more information about MS/MS is found and more features to help improve peptide identification results are proposed.

1.2 Research Objectives and Contributions

The purpose of this thesis is to find new features from existing peptide-spectrum match (PSM) database for peptide sequencing software, mainly by using machine learning ideas and techniques.

These newly found features from the spectrum library are preferred to be independent from any specific peptide sequencing software and algorithms. In this way, they can incorporate with all of them working as post processes. These new features can be used to build separate score functions, other than the confidence score given by either database search or de novo sequencing software. These scores can be combined together afterwards to re-rank all peptide sequences candidates for a spectrum and then find the best one. Being capable of working with all software makes these new features more flexible and valuable.

Our contribution contains two parts of work, both related to find internal patterns in spectra and use machine learning techniques to improve peptide sequencing results.

The first one focuses on the intensity ratio between two adjacent y-ions in the spectrum. A method is proposed to obtain the probability distributions of such ratios. Based on

these distributions, a general scoring feature is presented which can be incorporated in the scoring function of all other peptide sequencing software. This scoring feature is proved in this thesis to improve both database search and de novo sequencing results significantly. Further more, a logistic regression [15] model designed specifically for Novor [3] is provided. This model can improve the performance of Novor even further.

The second one shows that the appearances of internal fragment ions in spectra are predictable. A prediction model using random forest is provided along with its possible applications and some discussion. To the best of our knowledge, this is a pilot work on this specific topic, and it shows that it is worth further study.

1.3 Thesis Overview

This thesis is constructed in the following chapters.

In Chapter 2, some basic proteomics background is reviewed, such as the relationship between protein, peptide and amino acids, mass spectrometry and the peptide sequencing methods. In Chapter 3, preliminary works are introduced such as software that are being optimized. Chapter 4 focuses on the intensity ratio of adjacent y-ions, our first contribution, and how it used to improve peptide identification results. Chapter 5 talks about internal fragment ions prediction and its applications. In the end, we will have a conclusion in Chapter 6.

Part of the work in Chapter 4 is submitted to The Sixteenth Asia Pacific Bioinformatics Conference.

Chapter 2

Background

2.1 Protein, Peptide and Amino Acid

Proteins, from the Greek proteios, meaning first, are a class of organic compounds which are present in and vital to every living cell. In the form of skin, hair, callus, cartilage, muscles, tendons and ligaments, proteins hold together, protect, and provide structure to the body of a multi-celled organism. In the form of enzymes, hormones, antibodies, and globulins, they catalyze, regulate, and protect the body chemistry. In the form of hemoglobin, myoglobin and various lipoproteins, they effect the transport of oxygen and other substances within an organism.

Proteins are also called polypeptides. This is because proteins are either very long peptides or combinations of several peptides. Peptides are naturally occurring biological molecules. They are found in all living organisms and play a key role in all manner of biological activity. [16]

The hydrolysis of proteins yields several peptides and further hydrolysis of peptides yields a set of amino acids. Amino acids are organic compounds containing amine (-NH2) and carboxyl (-COOH) functional groups, along with a side chain (R group) specific to each amino acid. Twenty standard amino acids are used to make the body's proteins. Table. 2.1 shows all these 20 amino acids. This thesis will mostly use 1-letter code to present amino acids.

There are two termini on both ends of a peptide, which are N-terminus and C-terminus. The N-terminus is the start of a protein or peptide referring to the free amine group (-NH2). While the C-terminus is the end of an amino acid chain (protein or peptide), terminated

Amino Acid	1-letter code	3-letter code	Chemical formula(-H_2O)	Monoisotopic mass(-H_2O)	Average mass(-H_2O)
Alanine	A	Ala	C_3H_5OH	71.03711	71.0788
Arginine	R	Arg	$C_6H_{12}ON_4$	156.10111	156.1875
Asparagine	N	Asn	$C_4H_6O_2N_2$	114.04293	114.1038
Aspartic Acid	D	Asp	$C_4H_5O_3N$	115.02694	115.0886
Cysteine	C	Cys	C_3H_5ONS	103.00919	103.1388
Glutamic Acid	E	Glu	$C_5H_7O_3N$	129.04259	129.1155
Glutamine	Q	Gln	$C_5H_8O_2N_2$	128.05858	128.1307
Glycine	G	Gly	C_2H_3ON	57.02146	57.0519
Histidine	H	His	$C_6H_7ON_3$	137.05891	137.1411
IsoLeucine	I	Ile	$C_6H_{11}ON$	113.08406	113.1594
Leucine	L	Leu	$C_6H_{11}ON$	113.08406	113.1594
Lysine	K	Lys	$C_6H_{12}ON_2$	128.09496	128.1741
Methionine	M	Met	C_5H_9ONS	131.04049	131.1926
Phenylalanine	F	Phe	C_9H_9ON	147.06847	147.1766
Proline	P	Pro	C_5H_7ON	97.05276	97.1167
Serine	S	Ser	$C_3H_5O_2N$	87.03203	87.0782
Threonine	T	Thr	$C_4H_7O_2N$	101.04768	101.1051
Tryptophan	W	Trp	$C_{11}H_{10}ON_2$	186.07931	186.2132
Tyrosine	Y	Tyr	$C_9H_9O_2N$	163.06333	163.176
Valine	V	Val	C_5H_9ON	99.06841	99.1326

Table 2.1: Standard Amino Acids Table.

by a free carboxyl group (-COOH). By convention, peptide sequences are written from N-terminus to C-terminus [17].

Not all amino acids in proteins are listed in Fig. 2.1. Some of the missing ones are formed by post-translational modifications (PTMs), which are modifications on those listed amino acids after translation during protein synthesis. PTMs will change the mass of these amino acids as well as some of their behaviours [18].

2.2 Mass Spectrometry

Mass spectrometry (MS) is an analytical technique that sorts ions based on their mass-to-charge ratio. It works by ionizing chemical compounds to generate charged molecules and measuring their mass-to-charge ratios. The mass is usually measured in Dalton (Da), which is 1/12 of the mass of a carbon atom, and is approximately the mass of a hydrogen atom. A mass spectrum is a plot of the ion signal as a function of the mass-to-charge ratio, which is used for analyzing the elemental composition of a sample or molecule, and for elucidating the chemical structures of molecules, such as peptides and other chemical compounds. Among a lot of MS techniques, orbitrap is one of the most wildly used MS procedure in protein and peptide analysis [19]. The process using orbitrap to generate MS spectra is shown in Fig. 2.1 [20].

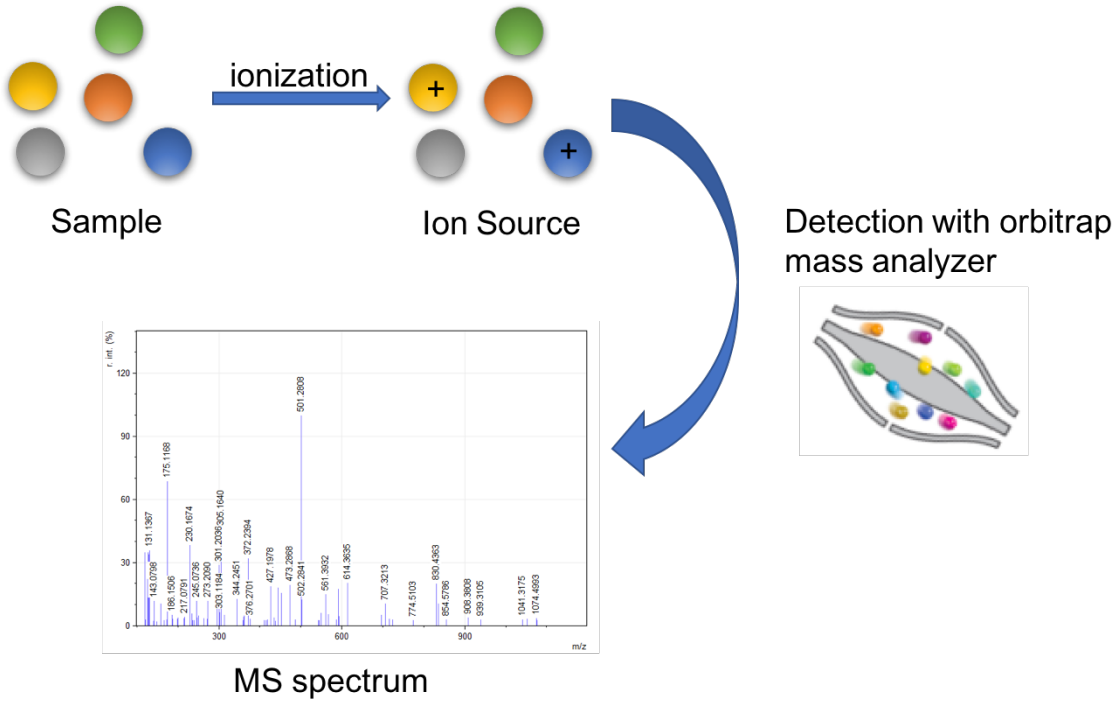

Figure 2.1: Basic instructions of a mass spectrometer (orbitrap).

Tandem mass spectrometry, also MS/MS, involves multiple steps of mass spectrometry selection, with some form of fragmentation happening in between the stages [21]. In a

tandem mass spectrometer, ions are generated in the ion source and separated by the mass-to-charge ratio in the first stage of mass spectrometry (MS1). Ions of a particular mass-to-charge ratio (precursor ions) are selected and fragment ions (product ions) are created by other physicochemical processes such as collision with argon atoms. The resulting ions are then separated and detected in the second stage of mass spectrometry (MS2). A simple sample of mass spectrum produced by such procedure is shown in Fig. 2.2.

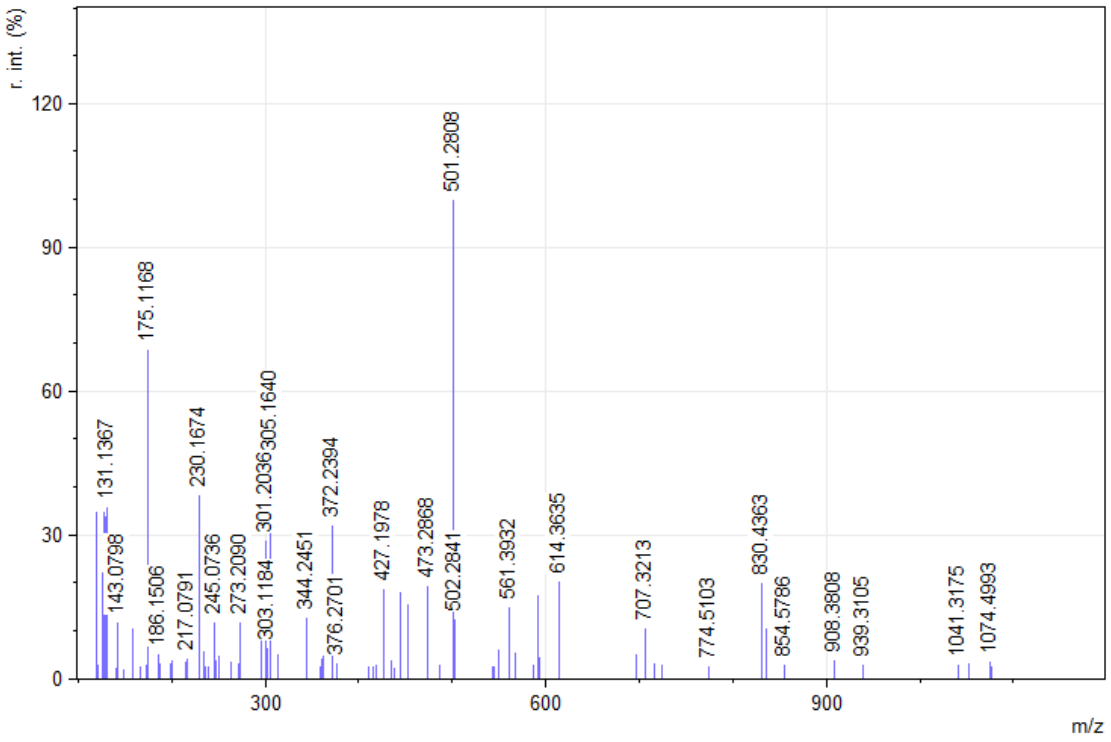

Figure 2.2: A Sample of mass spectrum.

Mass spectrometry is a very big topic to cover and details can be referred to [22]. However, only very limited knowledge is needed to understand this thesis. Getting rid of most bioinformatics background, a mass spectrum can be regarded as a set of key-value pairs, where key is a mass-to-charge ratio (m/z) showing which ion it presents and value is the abundance of this ion signal.

2.3 Y-ions and B-ions

The most common fragment ions observed in mass spectra are y-ions and b-ions.

Y-ions appear to extend from the carboxyl terminus, or C-terminus. The fragment containing only the carboxyl terminal amino acid is termed y_1. The fragment containing the first two amino acids from C-terminus is term y_2 and so forth [23]. The y-ion series for an example peptide MNSLQTDDTAK is illustrated below in Fig. 2.3.

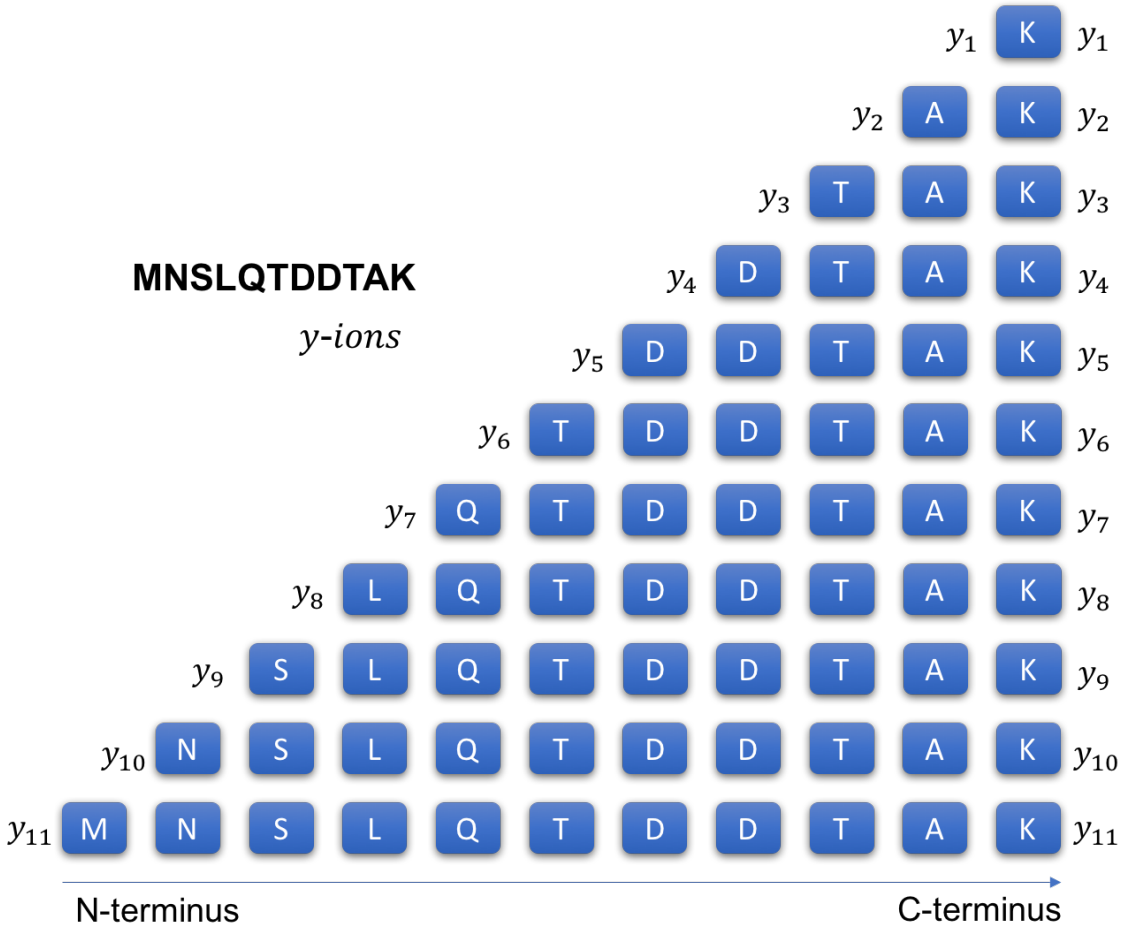

Figure 2.3: Y-ion series for example peptide MNSLQTDDTAK.

B-ions are similar with y-ions but in the reverse direction which appear to extend from

the amino terminus, or N-terminus (see Fig. 2.4) [23]. In most of the cases, y-ions have higher peak intensities (more abundance) in mass spectra than b-ions.

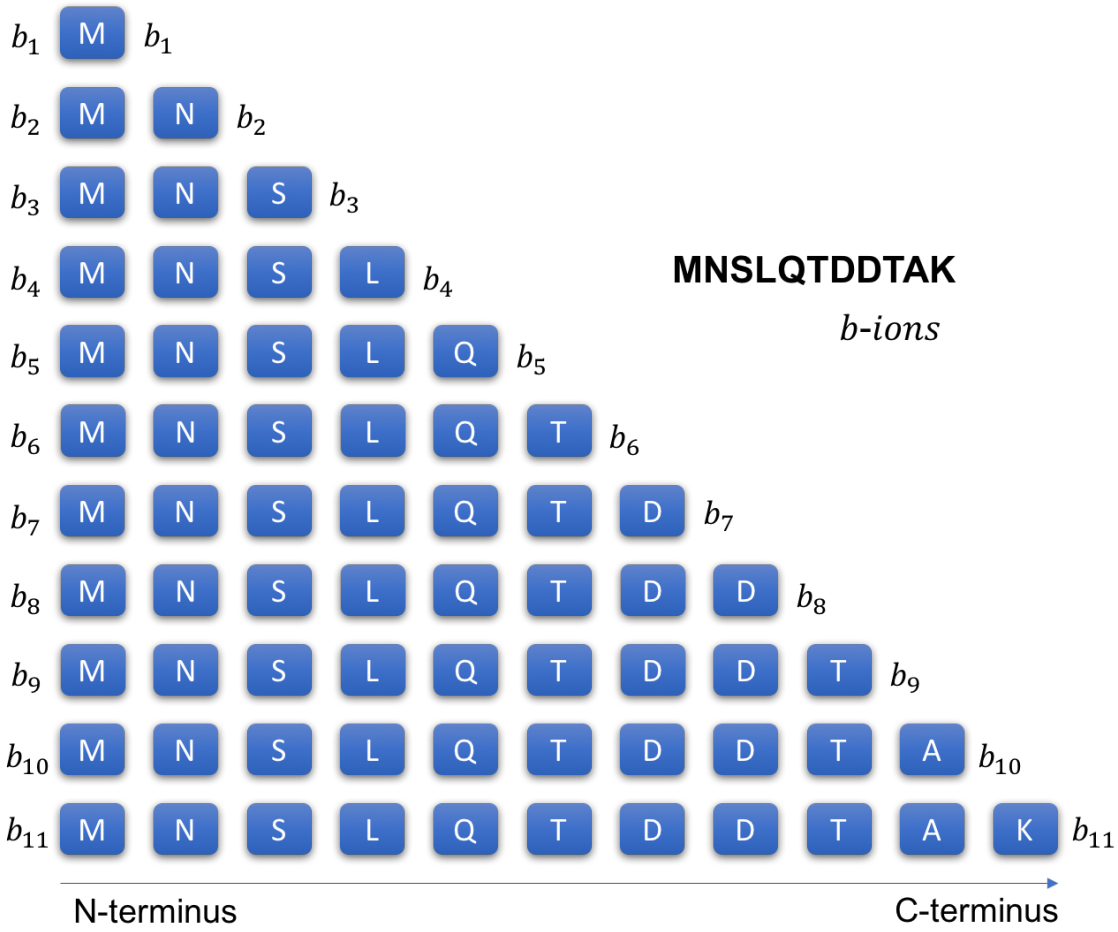

Figure 2.4: B-ion series for example peptide MNSLQTDDTAK.

Y-ions are b-ions are the two most significant and common fragment ions in MS/MS spectra, therefore, peptide sequencing algorithms are usually built based on them.

However, there all still some other types of ions in peptide mass spectra. One of them is internal fragment ion which refers to ions that contains neither the original C- or N-terminal residues. Ballard et al. first pointed out the existence of this type of ions back into 1991 [24]. However, no peptide sequencing software is taking advantage of this type

of ions yet.

2.4 Peptide Sequencing

Peptides are usually sequenced with tandem mass spectrometry. Researchers have been working on automatic algorithms to do this for decades.

Fig. 2.3 and 2.4 tell us that each peptide fragment in a series (either y-ions or b-ions) differs from its neighbour by one amino acid assuming the complete series is presented. In principle, it is therefore possible to determine the amino acid sequence by considering the mass difference between neighbouring peaks in a series of y-ions or b-ions. However, the difficulty lies in the fact that the information in MS/MS spectra is often not complete and that intervening peaks, which might or might not belong to the series, can confuse the analysis. This peptide sequencing method by looking solely at the spectra is called de novo sequencing. The name de novo comes from Latin which means from the very beginning. The success of de novo sequencing crucially depends on the quality of the data, in terms of both the mass accuracy and the presences and absences of the ions. Two well-known de novo sequencing software are PEAKS[5] and Nover[3].

At the beginning of the 1990s, researchers realized that the peptide sequencing problem could be converted to a database matching problem, which would be simpler to solve. The reason database searching is easier than de novo sequencing is that only an infinitesimal fraction of the possible peptide amino acid actually occur in nature. A peptide spectrum might therefore not contain sufficient information to unambiguously derive the complete amino acid sequence, but it might still have sufficient information to match it uniquely to a peptide sequence in the database on the basis of the observed and expected fragment ions. Some example software are X!Tandem[1][2], PEAKS DB[12], Mascot[9] and SEQUEST [8].

Database search methods usually have better results and are more tolerant on the quality of spectra. Meanwhile, de novo sequencing is more flexible as it does not rely on any previous results. Database search used to be faster than de novo sequencing, although this is being challenged with the development of the Novor software [3]. Due to their distinct capabilities, both approaches have traditionally been used in different circumstances. De novo sequencing has been mostly used to sequence endogenous peptides [25][26], characterize mutations in antibodies [25], perform proteomics analysis for organisms with no or incomplete protein databases [26][27][28], and to help sequence an entire protein [29][30][31][32]. While database search is widely used to study any organisms with a complete or partial protein sequence database. In real peptide sequencing processes, two

methods are usually combined together to fully sequence all spectra: de novo is applied to sequence the spectra where database search finds no or only low confidence matches.

For a very long time, these two approaches are developing separately. However, recently, the bioinformaticians have started to blur the boundaries of those approaches. The sequence tag approach uses de novo sequencing to identify confident partial sequences, and search a sequence database for identifying the exact or approximate sequence of the target peptide [12][33][34][35]. Also, de novo sequencing has been used to improve the speed of database search [12] and blind or unspecified PTM search ([33][34]). The PEAKS DB algorithm compares the de novo sequencing results with database search results, and uses the similarity to boost the confidence of the database search result [12]. The combination of multiple approaches within a single algorithm usually leads to more significant improvement than using different algorithms separately on different subsets of the data.

2.5 Target Decoy and Decoy Fusion

The output of database search programs indicates the best theoretical peptide matches to the input spectra, which are then used to infer the source peptide that was present in the biological sample. However, those matches are not always correct. Unfiltered sets of peptide identifications produced in this manner are necessarily imperfect for three reasons: (1) not all peptide species in a sample are represented in the search space; (2) spectra derived from background non-peptide species will often be given a peptide assignment; and (3) incorrect candidate peptide sequences occasionally may outscore correct sequences. For many search engines, nearly all input MS/MS spectra will be assigned a peptide match if there are any that lie within the supplied mass tolerance. Thus, the primary task of proteomics researchers is to distinguish incorrect from correct peptide assignments [36].

The "target-decoy" search strategy is a simple yet powerful way to deliver false positive estimations and can be applied to nearly any MS/MS workflow. Also, it is wildly used in proteomics to evaluate peptide sequencing performances. Here, we brief introduce some basic ideas about this well-known strategy.

One deceptively simple way to estimate false positives is to manufacture "decoy" sequences that do not exist in nature, and then allow the search engine to consider these alongside "target" sequences derived from the peptide database being studied. Necessarily, incorrect decoy hits should be similar to incorrect but unknown hits derived from target sequences in terms of length, amino acid composition, mass accuracy, and search engine-assigned scores. Therefore, knowing the proportion of decoy versus target sequences in

the search space allows one to estimate the number of incorrect target sequences in a reasonably large collection of peptide spectrum matches (PSMs).

Target-decoy searching strategy is used to evaluate peptide identification performances in this thesis in the following steps:

1. Construct decoy peptide sequences and put them into the target database, marking decoy sequences with a text flag in their annotation.

   Ideal decoy sequences should have the following characteristics:

   - Similar amino acid distributions as target protein sequences.
   - Similar protein length distribution as target protein sequence list.
   - Similar numbers of proteins as target protein list.
   - Similar numbers of predicted peptides as target protein list.
   - No predicted peptides in common between target and decoy sequence lists.

   Following these characteristics, several widely used decoy sequences are: reversed protein sequences, shuffled protein sequences and random protein sequences. They are theoretically fit the above requirements, although not perfectly, and very are easy to construct. To simplify calculation, the number of these decoy sequences are typically the same as the target ones and this is the approach that is used in this paper.

2. Use a search engine to identify input spectra using target-decoy sequence database.

   Once a target-decoy sequence database has been generated, the analysis of a set of spectra can begin. The generally accepted means to do this is to supply the search engine with a single protein sequence database consisting of both target and decoy sequences. For each spectrum, the search engine must then choose between target and decoy sequences. Correctly sequenced peptides will exclusively be selected from target protein sequences, while incorrect peptide matches will be randomly drawn from target and decoy sequences. In this thesis, the length of generated decoy proteins are the same with target proteins, so there should be a one-to-one correlation between target and decoy sequences among incorrect identifications.

3. Evaluate search engine's performance under specific false discovery rate (FDR).

   The number of correct sequenced peptide spectrum matches is the most important factor to evaluate search engine's performance. However, most of their sequencing

results are not 100% sure. Therefore, false discovery rate is needed to show the quality of results.

The false discovery rate is defined to be $\frac{\#\text{False Positive}}{\#\text{True Positive} + \#\text{False Positive}}$. This can be phrased as how many incorrect results can be tolerant in the outputs. Usually in proteomics, FDR is set to be 1%, or 0.1% if really high-quality results are needed.

In this way, different methods are compared according to how many PSMs they can output under such FDR. To be more specific, those methods will output PSMs in this following way,

(a) Sort all PSMs output by score, descending.

(b) Check PSMs one by one, record the last index satisfies that

$$\frac{\#Decoy}{\#Decoy + \#Target} \leq FDR$$

(c) Output all PSMs before this index, presuming they are correct.

According to the discussion before, the number of decoys should be the same as the number of false positive results in all discoveries. Therefore, only FDR portion of the output are incorrect. All those PSMs outputed using this method are called valid PSMs in this thesis. The number of valid PSMs is used to evaluate the performance of peptide sequencing methods.

There is a small problem when using target decoy during a multi-stage search procedure, where target decoy can make it biased toward underestimating the FDR [37][38][39]. It is mainly because in the protein shortlisting step, such search procedures may select more target proteins than the decoy proteins. This causes the false identifications in later steps to fall in the target proteins with a higher probability.

To solve this problem, Zhang et al. provides a new similar method called decoy fusion. Instead of adding decoy as separate protein sequences in the database, decoy fusion concatenates the target and decoy sequences of the same protein together as a single entry in the database [12]. While all other steps remain the same.

Chapter 3

Related Works

3.1 X!Tandem

Database search is a commonly used method to identify proteins in tandem mass spectrometry. It is a process that requires search algorithms to compare observed spectra against protein databases and identify potential matches. A number of programs exist for performing this search, examples can be found in 2.4. Among them the most popular one is an open source program, X!Tandem[1][2].

X!Tandem's major innovation is to conduct the search in two phases. In the first phase, a rapid survey identifies candidate proteins that are approximate matches to the input spectra. In this phase, perfect cleavage is assumed, and no post-translational modifications are allowed. In the second phase, a new search is conducted against only the candidates identified in the first phase, this time permitting refinements such as missed cleavages and post-translational modifications, which greatly increase the complexity of the search. Performing this refined search against the smaller population of candidates from the first phase significantly reduces search time.

To further improve its performance, X!Tandem has revised versions running on cluster computer systems, which are called Parallel X!Tandem [40] and X!!Tandem [41]. Even though X!Tandem is a open source software, it is under very good maintenance. All documentations can be found in http://www.thegpm.org/tandem/ and new versions are pushed very frequently. It has a version for MacOS where our most experiments run.

3.2 Percolator

The first step in analyzing a mass spectrometry dataset is to match the harvested spectra against a target database using database search engines such as SEQUEST[8] and Mascot[9] and X!Tandem[1][2], a process that returns a list of peptide-spectrum matches. However, it is not trivial to assess the accuracy of these identifications.

In order to filter out incorrect results, target-decoy strategy is used to find a threshold score under specific FDR. However, because most database search algorithms return multiple scores (for example, XCorr, Sp, and ΔCn for SEQUEST), most proteomics studies apply separate thresholds to each score. Using multiple orthogonal score criteria is useful for eliminating false discoveries that might exceed one threshold but not another. However, in most cases these orthogonal scores are considered independently, ignoring the benefits that can be obtained if the features are considered jointly.

Percolator[13] provides a solution to this problem. It uses a software post-processor that can be appended to any existing database search algorithm. Percolator uses a semi-supervised learning method that eliminates the need to construct a manually curated training set. The PSMs derived from searching a decoy database consisting of shuffled protein sequences are used as negative examples for the classifier, and a subset of the high-scoring PSMs derived from searching the target database are used as positive examples. Percolator trains a machine learning algorithm called a support vector machine (SVM) [42] to discriminate between positive and negative PSMs. One benefit of the semi-supervised learning paradigm is that the classifier is free to exploit a variety of specific features of the data, without overfitting to a particular type of spectrum. Percolator represents each PSM using a rich vector of 20 features.

In their original paper, percolator was used to optimize results from SEQUEST and Mascot. It increased correctly sequenced PSMs (or valid PSMs) by 17% in their experiments. In later versions, it supports more software such as X!Tandem, which was used in our later experiments.

3.3 Novor

Before the publishing of Novor [3], database search method were more widely used than de novo sequencing due to the consideration of both speed and accuracy. De novo sequencing is mostly used when there is no protein database to search against.

When a protein database is available, de novo sequencing can be used to assist database search to increase its sensitivity and accuracy by confirming its results [12], and to speed up database search by using de novo sequence tags as a filter [12][33][34][35]. However, these uses are often diminished by the relatively slow speed of previous de novo sequencing software. Besides the speed, the accuracy of existing de novo sequencing is not ideal. This is primarily due to the inherent difficulty of de novo sequencing but there is still room to improve.

Novor attempts to address these challenges and develop new software to achieve a real-time de novo sequencing speed with much improved accuracy over the previous ones.

In Novor, a large scale machine learning experiment was conducted using a decision tree model. Up to 169 features were used, and the decision tress with thousands of branching nodes were automatically generated from the training data automatically. Part of the decision tree Novor trained is shown in Fig. 3.1.

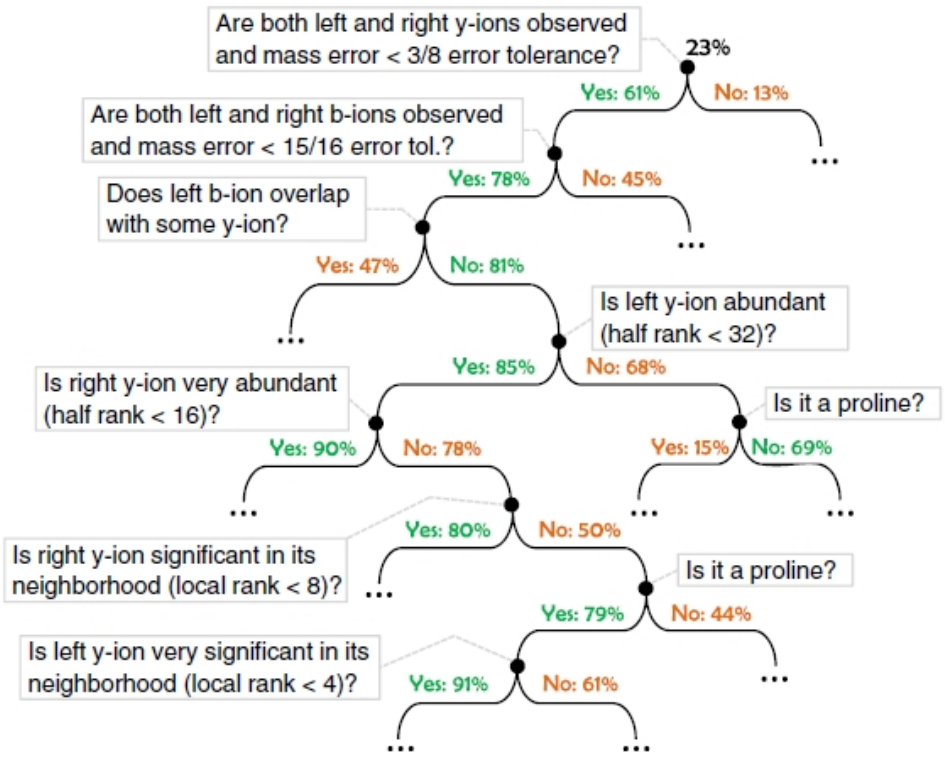

Figure 3.1: A small portion of decision tree learned by Novor [3].

In their experiments, Novor significantly improved the de novo sequencing accuracy and is more than an order of magnitude faster. More than 300 MS/MS spectra can be sequenced by Novor on a laptop computer per second, which opens the possibility to develop a fast speed protein identification method using de novo results.

3.4 Spectrum Prediction

To sequence peptides with tandem mass spectrometry, many software tools rely on the comparison between an experimental spectrum and a theoretically predicted spectrum. Consequently, the accurate prediction of the theoretical spectrum from a peptide sequence can potentially improve the peptide identification performance and is an important problem for mass spectrometry based proteomics.

Spectrum prediction is proven to be theoretically possible in [43]. However, the accurate spectrum prediction is still a challenging problem due to the lack of quantitative understanding of the complex peptide fragmentation process in MS/MS. To circumvent this difficulty, most of available database searching tools employ certain oversimplified models. For example, the widely used SEQUEST software assigns artificially determined fixed intensities to different types of ions [8]. This oversimplification usually incurs a significant deviation between the experimental spectrum and the prediction and may lead to compromised peptide identification performance [43].

A pioneer research in the computational prediction of a peptides spectrum was the MassAnalyzer program [43][44]. On the basis of the mobile proton hypothesis, a kinetic model was proposed to simulate the peptide fragmentation process. The model includes most fragmentation pathways listed in the literatures and additional pathways observed by the author. A total of 236 parameters were trained from the spectra of known peptides and used in the computer simulation algorithm for predicting the spectra of new peptides. The model was first developed for peptides with a net charge of +2 [43] and then extended to spectra with more than 2 charges [44]. Besides MassAnalyzer, there have been other works for studying the relationship between peak intensities and peptide sequences, without explicitly predicting the spectrum [45][46][47][48][49][50]. Different from the kinetic model, another program, PeptideART[51], uses a machine learning approach to predict the probability that a specific ion is observed.

Recently, a new software called MS-Simulator is proposed by Sun et al. [4]. Instead of predicting all the peak intensities simultaneously in MassAnalyzer, the new model focuses on the accurate prediction of the relative intensity ratio between every two adjacent y-

ion peaks. Also in this paper, they showed that the prediction of this ratio is a closed-form equation that involves up to five consecutive amino acids nearby the two y-ions and the two peptide termini. Compared with another existing spectrum prediction tool MassAnalyzer, MS-Simulator not only simplifies the computation, but also improves the prediction accuracy. The ratios predicted by MS-Simulator can also be used to derive a theoretical spectrum. The work flow briefly shows in Fig. 3.2.

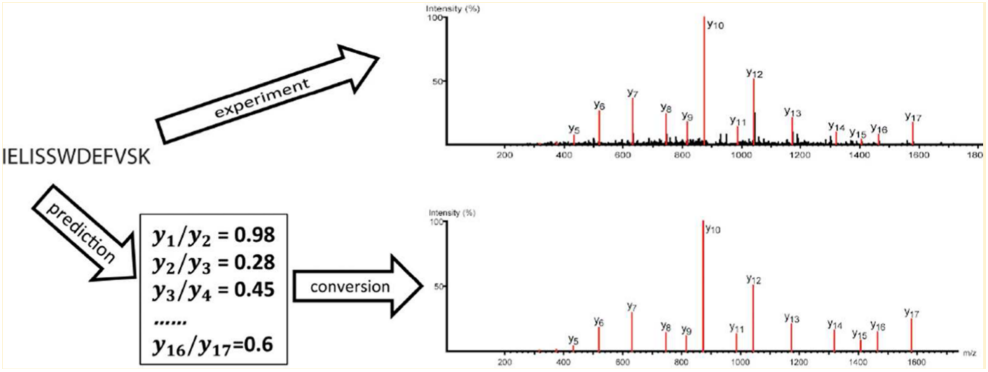

Figure 3.2: MS-Simulator's work flow to generate a theoretical spectrum. [4].

Chapter 4

Adjacent Y-ion Ratio Score Function

4.1 Introduction

In today's proteomics research, bioinformatics software has been routinely used to identify peptides from tandem mass spectrometry data. The software programs in use can be classified in three categories: database search [8][9][10][11][12], de novo sequencing [5][6][7][3], and spectral library search [52]. In each of these software programs, a good scoring function is critical in order to rank the correct peptide on top of the incorrect ones.

Developing a scoring function is a nontrivial task and has a profound impact to the performance of the software program. Some of the earlier software programs used empirically designed scoring functions. For example, SEQUEST used an $XCorr$ score [8]. For a given pair of peptide and experimental spectrum, SEQUEST computes the theoretical spectrum of the peptide with a simple empirical rule. Then the experimental spectrum is displaced by a mass shift and the dot product of the theoretical spectrum and the displaced experimental spectra is denoted with R_τ. The difference between R_0 and the average of R_τ for $-75 \leq \tau \leq 75$ is used as the $XCorr$ score. The de novo sequencing tool PEAKS used an empirical scoring function to evaluate the significance of the matched fragmentation peaks at each fragmentation site of the peptide. Then the sum of the fragmentation sites is used as the peptide score [5].

Other programs relied on simple statistics to weigh different matching events. For example, in the Sherenga program [53], two probabilities, p and q, are first counted with spectra of known peptides. The probability p is of that a fragment peak with certain ion type appears in the spectrum. And the probability q is of that a peak appears in a

random m/z value of the spectrum. The log likelihood ratio, $log\frac{p}{q}$, is then used as the score contribution of a peak matching a fragment ion with the selected ion type. The total of all matching peaks with all concerned ion types is used as the de novo peptides score.

The database search program X!Tandem followed SEQUEST and Mascot, combining three important attributes from each through the calculation of an expectation value. The expectation value is a probabilistic assessment of the correlation of an experimental and theoretical spectrum (similar to $XCorr$ in SEQUEST) and how much better a peptide match is than a stochastic match [54].

More recent programs started to adopt the machine learning method in developing scoring functions. One of the first efforts in this direction is the Percolator program that used a Support Vector Machinery (SVM) method in machine learning to combine 20 features. Percolator significantly improved other database search engines results by re-ranking their results with Percolators new scoring function. In another work, Frank et al. [55] used a logistic regression model to combine several features together to estimate the correctness of the de novo sequencing results of PepNovo [6]. Novor used a decision tree method to combine 169 scoring features in determining the correctness of an amino acid in a de novo peptide [3]. Machine learning has demonstrated great success in designing good scoring functions due to its ability to combine scoring features that are independently designed. In this chapter, we study the use of the intensity ratio between two adjacent y-ions as the scoring feature.

As we mentioned before, in an earlier study, the relative ratio of two adjacent y-ions has been used to predict the spectrum of a peptide from the amino acid sequence [4]. In that paper, Sun et al. demonstrated that the ratio between two adjacent y-ions is mostly determined by the few surrounding amino acids. A closed-form formula was proposed to predict this ratio from four surrounding amino acids. Then the ratios of all adjacent y-ions are used together to predict the y-ion intensities in the spectrum. Compared to the original spectrum prediction method proposed by Zhang [43], Suns method is significantly simpler and achieved satisfactory performance in the prediction. In the Novor software, the adjacent y-ion ratio is also used for scoring an amino acid's confidence.

In the previous studies, the adjacent y-ion ratio is treated as a constant once the surrounding amino acids are given. However, in reality, the ratio forms a probability distribution and may change from one peptide to another. Therefore, in this chapter we present a way to obtain such probability distributions through non-parametric statistics [56], and the use of such distributions in scoring feature to improve the scoring functions of database search and de novo sequencing, respectively.

The remainder of this chapter is presented as follows: Section 2 introduces how to generate the probability distributions mentioned above and show that they are valuable for peptide identification methods. Section 3 introduces y-ion ratio score function and how to apply it as a score feature to database search (X!Tandem) and de novo sequencing (Novor). Sections 4 presents the results of these experiments and these results show that this score function can be easily applied to improve both performances.

4.2 Adjacent Y-ion Ratio Distributions

Spectrum libraries contain a lot of valuable information. This chapter will focus on the intensity ratios of y-ion peaks on both sides of a residue. These y-ion peaks are also called adjacent y-ions in this chapter. For example, in Fig. 4.1, y_1 and y_2 are on both sides of amino acid S, so they are adjacent y-ions.

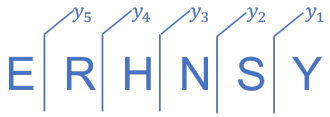

Figure 4.1: Y-ions for an example peptide.

Sun et al. [4] found that these ratios can be mostly determined by several more consecutive amino acids near this residue itself. Rather than use a fixed ratio value in [4], a probability distribution is generated to describe this pattern for every specific consecutive amino acid combination in this chapter.

Furthermore, as these distributions for different consecutive amino acids vary a lot, peptide identification algorithm can benefit from a scoring feature based on this.

4.2.1 Distributions Generation

The spectrum library released on May 29, 2014 from NISTs website (chemdata.nist.gov) was used to generate those distributions. In the spectrum library, there are data of spectra and those corresponding peptides. It consists of $340,357$ spectra measured with Ion-trap and all of them are used for generation.

As we mentioned before, adjacent y-ion peak intensity ratios can be mostly determined by several surrounding residues. In this chapter, four consecutive amino acids (4-mer) are used to present the pattern of these ratios. It is because, first, four residues can already mostly determine this ratio, and second, there are only $20^4 = 160,000$ combinations which can still be easily stored and accessed.

Furthermore, if a 4-mer is using to determine the ratio, it is better to have two more amino acid on the left and one more amino acid on the right [43][44]. This is slightly better than the opposite way, having two more amino acids on the right.

Here are the steps for generating an adjacent y-ion ratio distribution for a given 4-mer.

1. All y-ions from those spectra are traversed. Every time we met this specific 4-mer, let's say $A_1A_2A_3A_4$, y-ion intensity at the right-hand side of $A_3(y_i)$ and y-ion intensity at the left-hand side of $A_3(y_{i+1})$ are taken for further calculation. Intensities of y-ions are computed as the sum of peaks intensities for both charge 1+ and 2+ in the corresponding spectrum. If any peak intensity is less than 2% of the highest y-ion peak intensity in this spectrum, it will be regarded as noise and this peak intensity will be 0.

2. All those adjacent y-ion pairs (y_i and y_{i+1}) are used to calculate in formula

$$\frac{180}{\pi} \cdot arctan(\frac{y_i}{y_{i+1}}) \tag{4.1}$$

   as a data point (or a y-ion ratio in this chapter) to generate the distribution for this 4-mer. It would be a degree between 0 and 90 (both included). The use of arctan function converts the ratio from range 0 to ∞ to a constant interval, making it more convenient to carry out the statistics.

3. All those data points with value 0 or 90, which means one of the adjacent y-ions is missing, will be processed separately as outliers.

   For each different 4-mer, the probabilities of the y-ion ratio being 0 or 90 degrees are calculated using a classical probability model. More specifically, the probability of the y-ion ratio being 0 degree is the total number of 0 degree out of the number of all y-ion ratios. And the same for those ratios of 90 degree.

   After this step, the distributions of other data points are drawn. Take 4-mer VPDL as an example, the distribution of adjacent y-ion ratios looks like the bar chart in Fig. 4.2. This frequency distribution cannot fit into any typical probability distribution model such as normal distribution.

4. In order to easily present this distribution for further use, smoothing is a necessary step. In this case, non-parametric statistics[56][57] is a used as follows. For each 4-mer, each data point (y-ion ratio) is used to generate one normal distribution $N(\mu, \sigma)$, where μ equals to this ratio and σ equals to 10 degrees. To aggregate those distributions, a new distribution is generated by averaging all the normal distributions. The smoothed distribution is shown as the line chart in Fig. 4.2. Probabilities for 0 and 90 degrees calculated in step 3 will be attached to this distribution after this step.

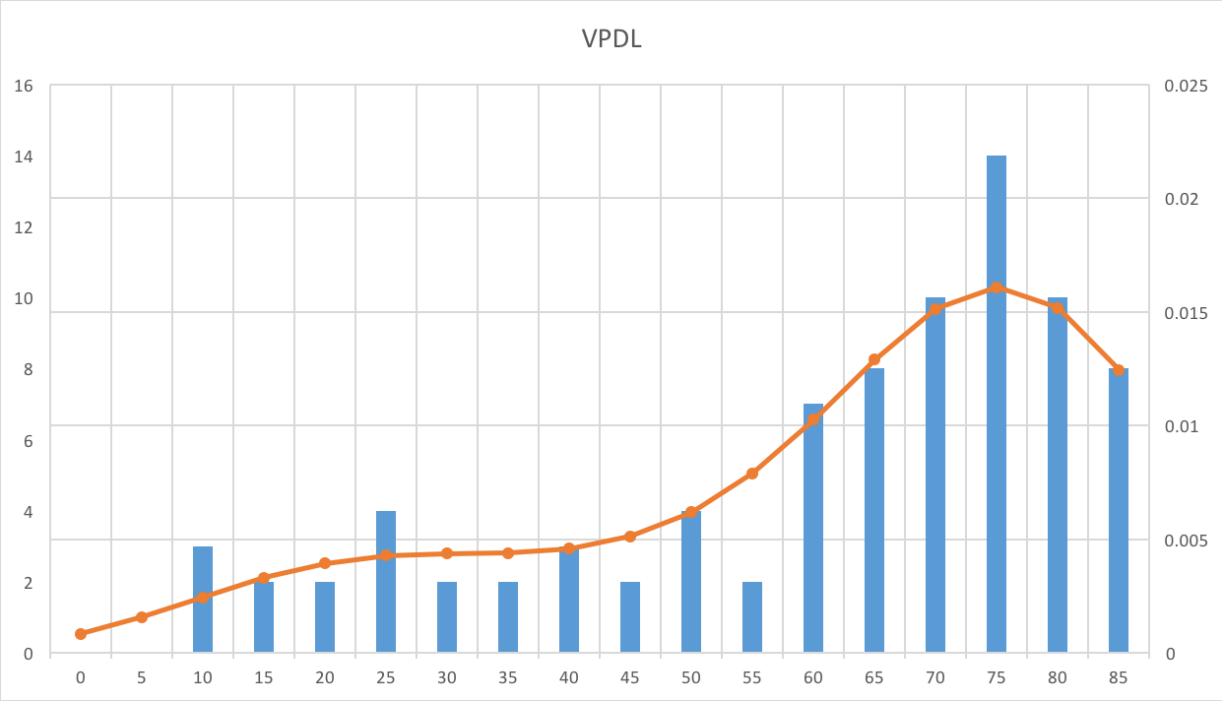

Figure 4.2: Frequency of all data points and the distribution after smooth for 4-mer VPDL.

After all those steps, a distribution for this 4-mer is generated presenting the probabilities of any potential y-ion ratio values. Also, to simplify this probability distribution, only probabilities for integer degrees are stored. Probabilities between integers will be calculated by linear interpolation.

Here the algorithm is used to generate distributions for 4-mers. However, for some 4-mers there are too few data points to show a pattern. Therefore, besides all 4-mers ($A_4A_3A_2A_1$), distributions for 3-mers ($A_3A_2A_1$) and 2-mers (A_2A_1) are also generated and

stored. In all circumstances, y-ion intensity on both sides of A_2 are always used to calculate the ratios.

When looking up a specific 4-mer (say $A_4A_3A_2A_1$) for a probability, its 4-mer distribution will be checked first. If there are only less than 50 y-ion ratios, which we think may not be sufficient to show the pattern, the 3-mer distribution (for $A_3A_2A_1$) will be checked. If the number of y-ion ratio entries is still less than 50, the 2-mer distribution (for A_2A_1) will be checked. Once data points are sufficient to show the pattern, probability of the degree in this distribution will be returned.

Besides all these distributions, we have a distribution named "global". This "global distribution" put all y-ion ratios in the whole spectrum library into one distribution regardless of the 4-mers. This can be regarded as a global setting for this human library and will be used to calculate likelihood which will be introduced in section 4.3.

4.2.2 Different Distributions Comparison

Four probability distributions for four different 4-mers (AAAA, EEEE, VPDL and SNPS) are presented in Fig. 4.3 and 4.4. Fig. 4.3 shows the smoothed distribution without probabilities of 0 or 90 degree while Fig. 4.4 shows those probabilities for 0 or 90 degree alone.

These distributions are very different from each other. Therefore, it is valuable to take y-ion ratio into consideration when doing peptide identification.

This procedure is also run using other libraries from other species. However, they all generated almost the same distributions. Thus, even though all the distributions are generated from human spectrum library, they can be easily applied to sequence peptides of all species. This means our score function discussed below can be commonly used regardless of the species.

4.3 Y-ion Ratio Score Function

In this section, our y-ion ratio score function is introduced as well as experiment details carried on both X!Tandem [1][2] and Novor [3].

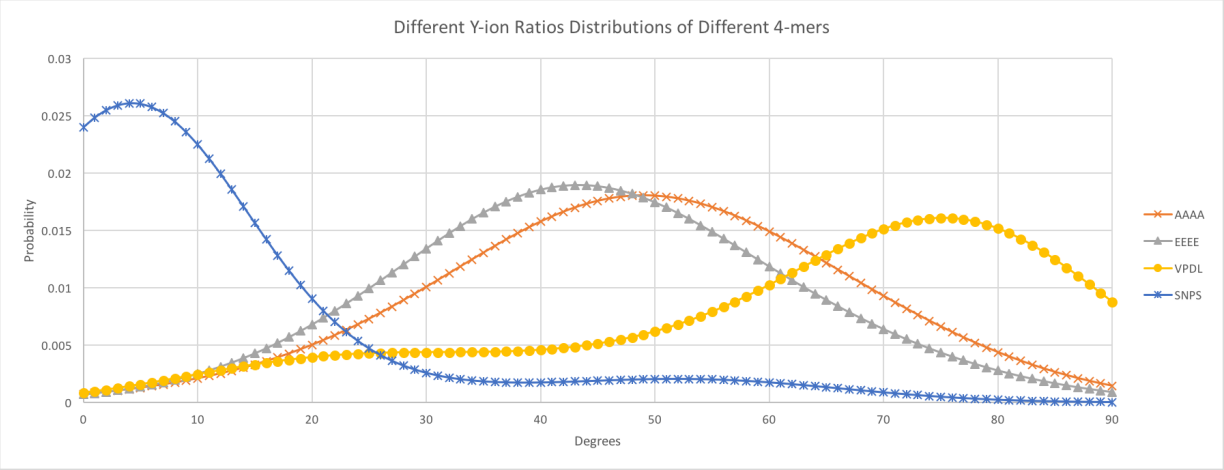

Figure 4.3: Different distributions for different 4-mers.

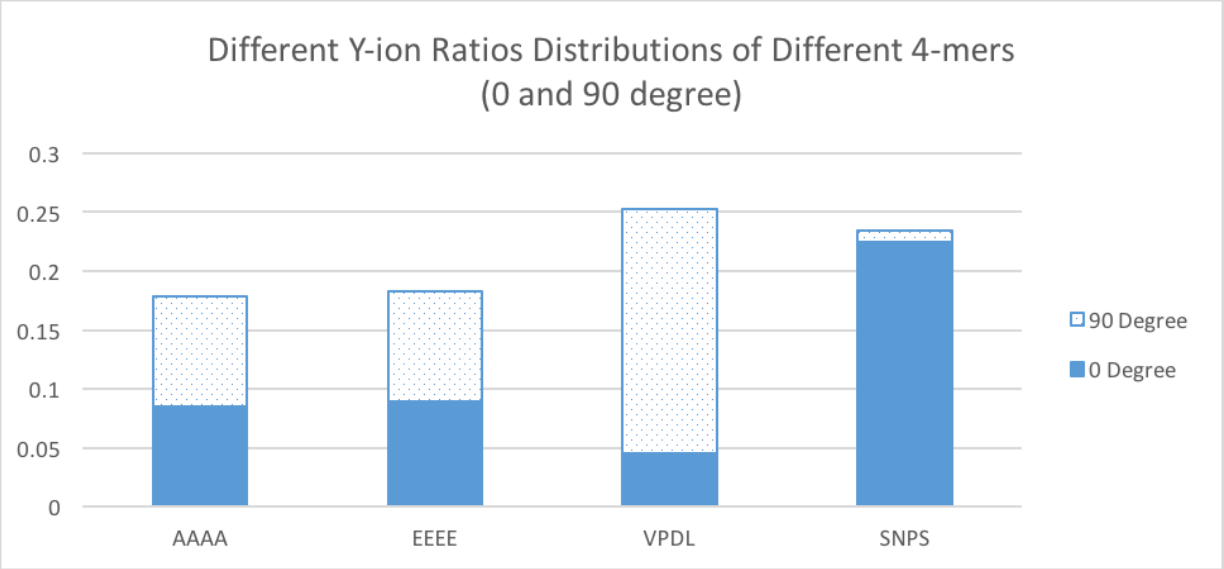

Figure 4.4: Different 0 and 90 degree probabilities different 4-mers.

4.3.1 Score Function

For each single residue in a peptide, a score called y-ion ratio score is proposed in this chapter. This score is calculated based on both peptide sequence and its corresponding spectrum (i.e. peptide sequence match, or PSM). These PSMs can come from any peptide identification software.

Given a residue A_l in a peptide, two residues before A_l and one residue after A_l are fetched to form a 4-mer, i.e. $A_{l-2}A_{l-1}A_lA_{l+1}$, and the y-ion ratio around this given amino acid A_l is calculated.

Then, the distribution of this 4-mer is checked for the probability of this y-ions ratio under this distribution, i.e.

$$P_1 = Pr[ratio|A_{l-2}A_{l-1}A_lA_{l+1}] \tag{4.2}$$

Also, the probability of this y-ions ratio under the global distribution is calculated, i.e.

$$P_2 = Pr[ratio|Global] \tag{4.3}$$

Finally, the likelihood of this y-ions ratio being under this specific 4-mers distribution rather than the global distribution is calculated as our y-ion ratio score.

$$\text{Y-ion Ratio Score} = log\frac{P_1}{P_2} \tag{4.4}$$

Note if two neighbor y-ions of this amino acid are both missing in the spectrum, the y-ion ratio score will be 0. Also, y-ion ratio score will always be 0 when it comes to N terminus or C terminus.

We have two experiments showing our scoring function has a significant improvement for both database search approach (X!Tandem [1][2]) and de novo sequencing (Novor[3]). In the first experiment, y-ion ratio scores of every amino acid in the peptide are summed up as a score for the whole peptide spectrum match (PSM) and used in conjunction with X!Tandem score to improve its performance on peptide identification. In the second experiment, scores are given to each single residues to better predict its correctness with Novor's confidence scores.

4.3.2 X!Tandem Optimization

After we had all these distributions, we first tried it to optimize database search based peptide sequencing. X!Tandem [1][2] is chosen because it is one of the most famous free software in this area. It is under good maintenance and it also has a version for MacOS.

Briefly, by having a new y-ion ratio score along with X!Tandem score, two different scores are given for one single PSM. A new score is calculated by simply linear combining those two scores and this new score is used to re-rank all those PSMs outputted by X!Tandem.

Datasets

Two public data sets downloaded from proteomeXchange Datasets are used to search against a human peptide database. They are:

- U2OS

  This dataset was downloaded from the proteomeXchange data repository (ID: PXD001220). The data was produced by Kirkwood et al. in their study of native protein complexes and protein isoform variation in human osteosarcoma (U2OS) cells[58]. We only used one file, PT1541S1F16.raw, which contains 36169 MS/MS spectra, for the experiment.

- Iris

  This dataset was downloaded from the proteomeXchange data repository (ID: PXD002194). The data was produced by Zhang et al. in their study of human iris, ciliary body, retinal pigment epithelium, and choroid [59]. We combined from 0orHDJ8h-2509-01.mgf to 0orHDJ8h-2509-12.mgf as a whole spectrum file to run the experiment which containing 87,489 MS/MS spectra.

The database we searched against is downloaded from UniProt. The proteome ID is UP000005640. There are 70,225 proteins in the database.

Decoy-fusion

Before running X!Tandem, in order to evaluate its performance, a decoy database is needed. As X!Tandem is using a multi-stage database search algorithm, decoy-fusion [12] works better than target-decoy.

In order to do decoy-fusion on the target database, reverse of every protein sequence are appended to this protein itself. So, the newly generated database contains the same number of protein entries, while the length of each protein is doubled. This new generated database are used for X!Tandem to search against. Target and decoy identifications form X!Tandem are separated by checking whether they are from the first half or the second half of that new protein sequence. Note X!Tandem will output some same peptide sequences from several different proteins in the database. In this situation, if one of those peptides are from the first half of that protein, we will treat it as a target, if none, decoy.

If the C-terminal amino acid of the target protein is not an enzyme cleavage site, then appending a decoy sequence to its end may prevent the search engine from considering the C-terminal peptide of the target protein. A normal way to solve this problem is to add a special letter J between target and decoy sequences as the separator [12]. However, X!Tandem cannot deal with invalid amino acid letter like J, which it will change to other valid letter. So, instead of adding J, we added XXXXXXXXXX (10 Xs) between target and decoy. Also, we changed our way to identify target or decoy a little bit. If one peptide falls into the area of 10 Xs, it will be treated as neither target nor decoy.

Benchmark and Comparison Baselines

How many PSMs can be outputted under 1% false discovery rate (FDR) is used as a benchmark for the performances of different methods. In this chapter, number of valid PSMs is defined to be #target_PSM − #decoy_PSM when FDR equals to 1%.

First comparison was between using X!Tandem score only and the combination of X!Tandem score and y-ion ratio score. The version we used for X!Tandem is PILEDRIVER, released on 2015.04.01. The hyperscore X!Tandem outputted for each PSM is used as Tandem score.

In order to avoid over-fitting, a linear function is used to combine X!Tandem score and y-ion ratio score. To find the best parameter, we first run this process on a test dataset and drew a scatter plot for both scores.

According to Fig. 4.5,

$$\text{Tandem score} + 2 \times \text{Y ion Ratio Score}$$

is used as our final score after some observation.

This final score will be used to re-rank all the PSMs outputted by X!Tandem and the number of valid PSMs is calculated again based on this score.

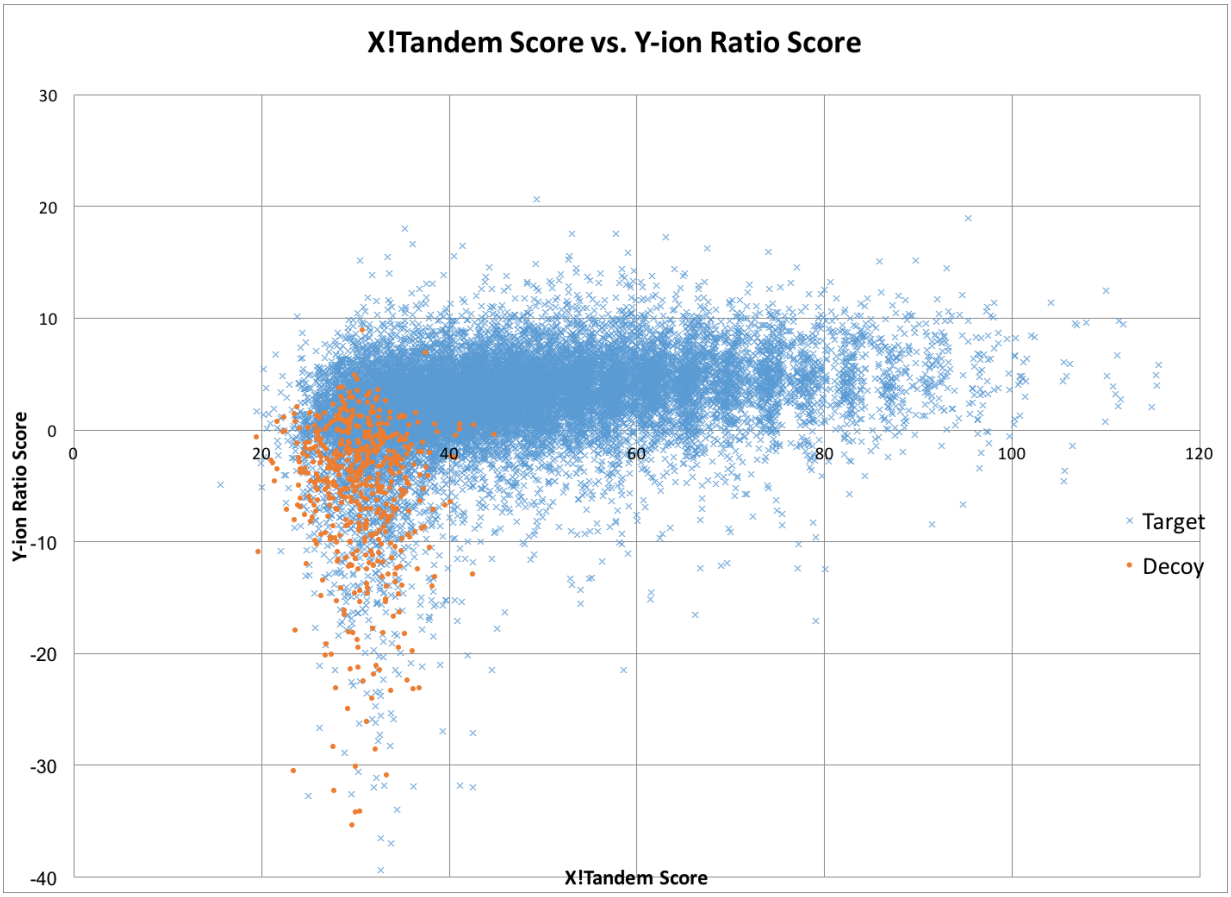

Figure 4.5: X!Tandem score vs. y-ion ratio score scatter plot.

Second comparison will be our revised X!Tandem score against percolator. Percolator uses a semi-supervised machine learning to discriminate correct from incorrect peptide-spectrum matches, and calculates accurate statistics such as q-value (FDR) and posterior error probabilities [13]. Percolator was downloaded from Github and used with its plugin command to transfer X!Tandem results to tab-delimited file format which is the input format for percolator. Percolator also needs another input which contains X!Tandem results against a totally decoy database as the negative labeled data points to do semi-supervised learning. For this decoy database, we shuffled every peptide entry in the decoy-fusion database we made and put them in another file. In this way, percolator interpreted every peptide in our previous decoy-fusion database is target and it outputted a percolator score for each PSM X!Tandem got. However, whether a peptide is a target or decoy can still be told by searching this peptide in the original database before we did decoy-fusion. Still, the number of valid PSMs is used as benchmark here.

Percolator has a very good interface where a feature could be easily added into its SVM model by simply revising its input file. Since features in percolator and y-ion ratio score are trying to do the same thing, y-ion ratio score can be treated as a feature in percolator as well. After those two comparisons, y-ion ratio score is added into percolator as a new feature and compared which feature contributed the most to the result. This was done by deleting features one by one and see which made the number of valid PSMs drop the most.

4.3.3 Novor Optimization

Y-ion ratio score function can not only be used to optimize database search results but also de novo sequencing results. Novor is a state-of-art de novo sequencing software and it is free to use in academia. For every peptide spectrum match, Novor gives confidence scores to each residue it predicts.

In database search, even if we have some peaks missing, there will still be some adjacent y-ions to provide y-ion ratio scores so we can sum them up as a score for the whole peptide. However, it is not the case here in Novor. If the peaks around an amino acid are both missing, our y-ion score will be zero all the time. In order to distinguish different situations where our new method cannot give a score and make sure our y-ion ratio score can still contribute, a slightly complicated model is brought up.

A logistic regression model is proposed to solve this problem. Logistic regression is a simple classification model wildly used in both academia and industry nowadays [15]. It was first developed by statistician David Cox in 1958 [60]. The binary logistic model we

were using here is used to estimate the probability of a binary response based on one or more predictor features. Our model contains features in table 4.1.

Feature Name	Explanation
Novor Score	The score Novor gave on an amino acid
Y-ion Ratio Score	Score we are introducing in this chapter
Missing_1	If at least one of the y-ions is missing around this amino acid
Missing_2	If both y-ions are missing around this amino acid
Is_N_term	If this amino acid is at N terminus
Is_C_term	If this amino acid is at C terminus

Table 4.1: Features in Novor Optimization Model.

Missing_1 and Missing_2 are two features showing that how many y-ion peaks are missing around this residue. Due to the property of logistic regression model, setting this into two features with each can be 0 or 1 is better than setting this into only one feature with possible value 0,1,2. These two features helped us to make up the model when y-ion ratio score cannot provide enough information (or y-ion ratio score is set to 0.0 by default).

Also, in more specific cases, N terminus and C terminus should be treated differently since our y-ion ratio score will always give 0 to termini.

This model has two labels, 0 and 1, where 0 means this residue is not correctly sequenced and 1 means it is correct. As it is logistic regression, this model will output the probability of the prediction to be 0 and 1 separately. The probability of the prediction to be 1 is treated to be the confidence score of our model.

Our logistic regression model was implemented using scikit-learn package (version 0.17.0) in Python [61].

A normal way to make full use of de novo sequencing is to run both database search software and de novo sequencing software. If a spectrum can be sequenced by database search software under 1% FDR, we will output this peptide as sequencing results. If it cannot, output de novo result as the final sequenced peptide.

Therefore, one workflow to use our new method is

1. Run a database search based peptide sequencing software (like PEAKS[5] and X!Tandem) to get PSMs they outputted under 1% FDR (or less) and regard them as ground truth

2. Run de novo sequencing software (like Novor) and get peptide for every spectrum

3. For those outputted both in (1) and (2), compare those peptides to distinguish residues that are correctly and incorrectly sequenced in de novo. Then use them to train our logistic regression model.

4. For those peptides only outputted in (2), use our new model based on y-ion ratio score is used to predict the result. If the prediction is true, we will also output this residue.

In this way, both outputs can be fully used and the residues that are correctly sequenced from the spectrum can be maximized. However, since all our features in the logistic regression model are instrument independent, the model can still be trained in advance and used to predict for all de novo results. Our experiments are also carried out in this way.

Datasets

Four datasets in [3] are used to replay the experiments. They are:

- C. elegans

  Similar to NIST human library, this dataset is the C. elegans ion trap peptide library (released date May 24, 2011), downloaded from the NIST website. It consists of 67,470 spectra and was produced with the same procedure as the human peptide library.

- Ubiquitin

  This dataset was extracted from a larger dataset recently published at the MassIVE database (ID: MSV000078991). The dataset was produced by Coyaud et at. in their study for E3 ubiquitin ligase [62]. Out of the 80 experiment (Control_BioID_no_bait_A_v1) was chosen in this study. The peptide identification results submitted together with the data were also downloaded, and the ones with a probability score of 95% or above were extracted and used as the ground truth. If a peptide was identified by multiple MS/MS spectra with the same charge state, only the spectrum with the highest score was kept. After this filtration process, 3398 non-redundant PSMs remained in the final list for benchmarking.

- UPS2

  This dataset was the data file MSups_15ul.RAW.gz in dataset 13 of the MS/MS data repository (www.marcottelab.org/MSData/) at Marocttes lab at the University of Texas, Austin. The data were generated by Vogel et al. for confirmation purposes in their previous study of mRNA and protein concentration [63]. To produce the data, the standard UPS2 sample was digested with trypsin, and measured with a LTQ Orbitrap. There are 9466 MS/MS spectra in the data file.

- U2OS

  This dataset is the same one we described in section 2.2.

As we mentioned before, these experiments are mainly redoing the experiments in [3], more details can be found in their original paper.

Benchmark and Comparison Baseline

As the experiments in [3], we firstly retrieved the results from both Novor and PEAKS[5] (ver. 7.0, Bioinformatics Solutions Inc.).

A residue x in the real peptide is considered as correctly sequenced if the de novo sequence reports a residue y with the similar residue mass at approximately the same prefix mass position. More specifically, both of the following two conditions need to be satisfied:

1. $|mass(x) - mass(y)| \leq 0.1Da$
2. the total mass before x and before y differ by at most 0.5 Da

The reason to only require approximate match of the mass is because the mass accuracy in low resolution mass spectrometers is not sufficient to distinguish residue pairs such as I versus L, K versus Q, and Oxidized M versus F.

Peptides sequenced by PEAKS DB [12] are used as ground truth. For each dataset, 80% of the PSMs outputted by PEAKS DB are randomly picked and used to train the logistic regression model and then those remaining data are used to test the model.

Novor outputs a confidence score between 0 and 100, and meanwhile our logistic regression outputs a confidence score (i.e. the probability of the prediction to be 1) between 0 to 1. To compare these two, we draw two precision-recall curves, one for Novor and the other

for logistic regression model, for each dataset. Basically, the higher the precision-recall curve is, the better this score function is.

Precision-recall curves are draw as following way, take Novor score as an example. Let N be the total number of residues in the real peptide sequences. For any given score threshold t, let denovo(t) be the number of residues with scores of at least t in the de novo sequences; and correct(t) be the number of residues that are correctly sequenced with score at least t. Then, the precisions and recalls of the algorithm at score threshold t are defined as,

$$recall(t) = \frac{correct(t)}{N} \tag{4.5}$$

$$precision(t) = \frac{correct(t)}{denovo(t)} \tag{4.6}$$

By traverse all possible thresholds t, we will get several points at a 2D space. The plot drew from those points will be treated as the precision-recall curve.

4.4 Experiments Results

In this section, the results of two experiments mentioned above are presented as well as some analysis on the results.

4.4.1 X!Tandem Optimization

U2OS Dataset

In U2OS dataset, there are 36,169 spectra. X!Tamdem outputted 18,191 PSMs, 17,655 of them were target in the meanwhile 536 of them were decoy. Under 1% FDR, 15,030 target PSMs were outputted with minimum score 28.3. Since 1% of them could be decoy according to the definition of target-decoy method, the number of valid PSMs was 14,880, which was 41.14% of the total number of spectra.

However, if our y-ion ratio scores and X!Tandem are combined together and used to re-rank all PSMs, 16,472 target PSMs were outputted. The number of valid PSMs should be $16472 \times 0.99 = 16307$ PSMs, which counted for 45.06% of the total number of spectra.

Furthermore, we tested this dataset with percolator. Under 1% FDR, 16360 target PSMs were outputted with minimum score -0.667. The number of valid PSMs were 16196, which counted for 44.77% of the total number of spectra.

In conclusion, for this dataset, our new score function can help X!Tandem output 1,427 more valid PSMs than X!Tandem itself. This number means it improves X!Tandem by 9.59%. Also, our new score with only two features (X!Tandem score and y-ion ratio score) beats percolator, which run SVM on 12 features, by 111 valid PSMs.

Iris Dataset

We combined all 12 different spectra files together so it is large enough to do our experiment. However, it also resulted in the low quality of the whole data comparing to selected ones from the middle of an experiment.

There are 87,489 spectra in our file totally. Using the same way as before, X!Tandem outputted 26,186 valid PSMs, which counted for 29.93% of the number of all spectra.

If our y-ion ratio score and X!Tandem score are combined together, the number of valid PSMs went to 30,044, which counted for 34.34%. So our new score function can help X!Tandem output 3,858 valid PSMs more. This means we improved X!Tandem on this dataset by 14.74%.

By using 12 scoring features, Percolator outputted 37.26% of all the spectra for this dataset. This is better than the linear combination of the two scoring features Hyper_score and Y-ion ratio score. However, by adding the Y-ion ratio score as an additional scoring feature to Percolator, Percolator was able to identify 38.19% of all the spectra for this dataset. This indicates that our y-ion ratio score feature can be used to further improve Percolator, despite that it already uses so many features.

It is informative to compare the relative contribution of each of the 13 features used by Percolator in order to achieve this 38.19% identification rate. For such purpose, we run Percolator 13 times. In each time, one of the 13 features are not used. The identification rate of each time is given in Table 4.2. Note that the line "None" indicates the inclusion of all 13 features. The relative performance drop of deleting a feature is a good indication on how important a feature is.

For the features in percolator, length means the length of the sequenced peptide; enzInt, enzC, enzN are enzymatic information about this PSM; Charge 3 is the charge state; dM and absdM are the differences in calculated and observed mass; Frac_ion_y and Frac_ion_b are realted to ions we got in the spectrum; Delta_score is p-value X!Tandem used to rule

Features Missing	Percentage of valid PSM
None	38.19%
Y-ion Ratio Score	37.26%
Length	38.21%
enzInt	37.83%
enzC	38.09%
enzN	38.02%
charge3	38.19%
absdM	37.84%
dM	37.88%
mass	38.06%
frac_ion_y	38.21%
frac_ion_b	38.24%
Delta_score	37.78%
Hyper_score	38.23%

Table 4.2: Percolator results without each single feature.

out unlikely PSMs; Hyper_score is the score X!Tandem give. It is still unknown why the percentage increases when some of the features are missing, one possible guess is that there is some redundancy in those features.

From the chart, the percentage dropped the most when the SVM model did not have y-ion ratio score as a feature. This means that y-ion ratio score plays one of the most important roles in this model.

In conclusion of two datasets, y-ion ratio score has the ability to improve X!Tandem results by about 10%. It can even beat percolator in some cases. In all cases, percolator can benefit a lot from adding y-ion ratio score as a new feature.

We tried to look into why y-ion ratio score did relatively better in U2OS than Iris. We found that the results of U2OS contains less missing y-ions than the other one. Recall that if two adjacent y-ions are both missing, y-ion ratio score of this residue cannot be provided. The more this case exists, the less information that y-ion ratio score of this peptide can provide. Therefore, our new score function performed relatively worse. In conclusion, even if our new score function performed well on both dataset, sequencing results with less missing y-ions can still benefit more from it.

4.4.2 Novor Optimization

The logistic regression model we proposed was run against 4 different datasets. Four precision-recall curves were drawn to compare the results between Novor and our new logistic regression model. Those curves are shown in Fig 4.6 4.7 4.8 4.9.

In those figures, our logistic regression model is presented by red connected triangles while the result of Novor is presented by blue connected dots.

Recall that this precision-recall curve was draw by setting a series of threshold t and calculate the recall and precision function mentioned before. In general, two curves in one figure both comes from point (0,1) to a same point. This is because even if we brought up a new model, this model still used the same results from Novor, so when drawing the point with threshold t approaches to 0, they were all considering the same results.

These curves show that our logistic regression model has a significant better precision under the same recall value almost all the time. This means no matter what trade-off people are using between recall and precision, our model is always a better one to use.

These precision-recall curves may be slightly different from the ones in [3]. This is because only 20% of the whole dataset are randomly picked to test and draw these curves. Our recall function was calculated against those 20% dataset, so it is not a contradiction if those recall value went beyond 0.2.

Also, it takes linear time for logistic regression to do both training and predicting [15]. Therefore, this logistic regression model is not time consuming at all in the whole peptide sequencing workflow. We tested it on a MacBook Pro laptop computer (Retina, Mid-2015, 2.2 GHz Intel Core i7, 16 GB RAM) against U2OS, where PEAKS DB gave 9857 PSMs. It took 75 seconds to both train 80% of dataset and test on the remaining 20%.

In conclusion, this logistic regression model provides a significant improvement for Novor almost all the time under those 4 datasets. Also, it is not time consuming comparing to the whole de novo sequencing method. Even though we only test our model in Novor, since the only thing this model used is the results outputted, our model can be applied to any de novo sequencing software.

4.5 Further revised model for Novor

Since a logistic regression model is already deployed and it achieved a very good result, it is straightforward to do one more step: adding several more features related to Novor itself

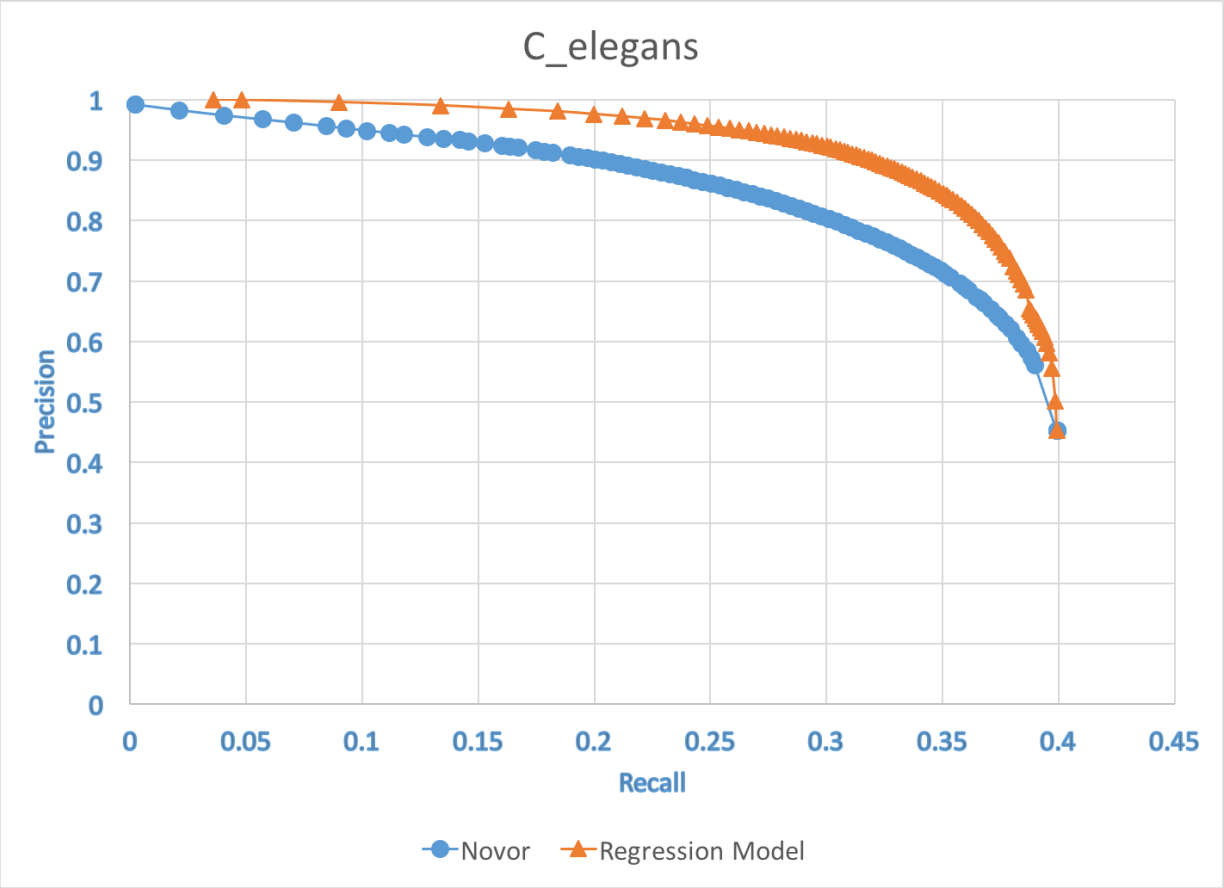

Figure 4.6: Optimization results for dataset C. elegans.

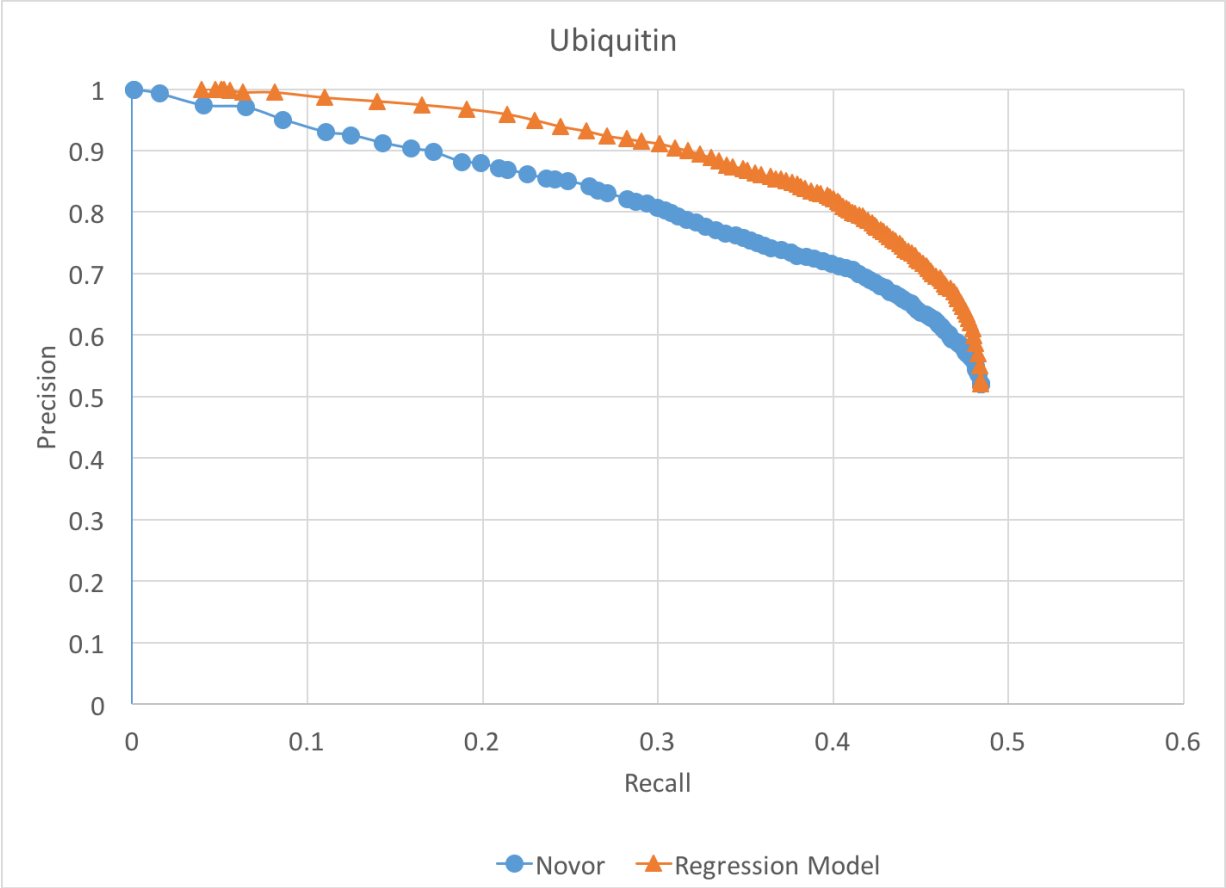

Figure 4.7: Optimization results for dataset Ubiquitin.

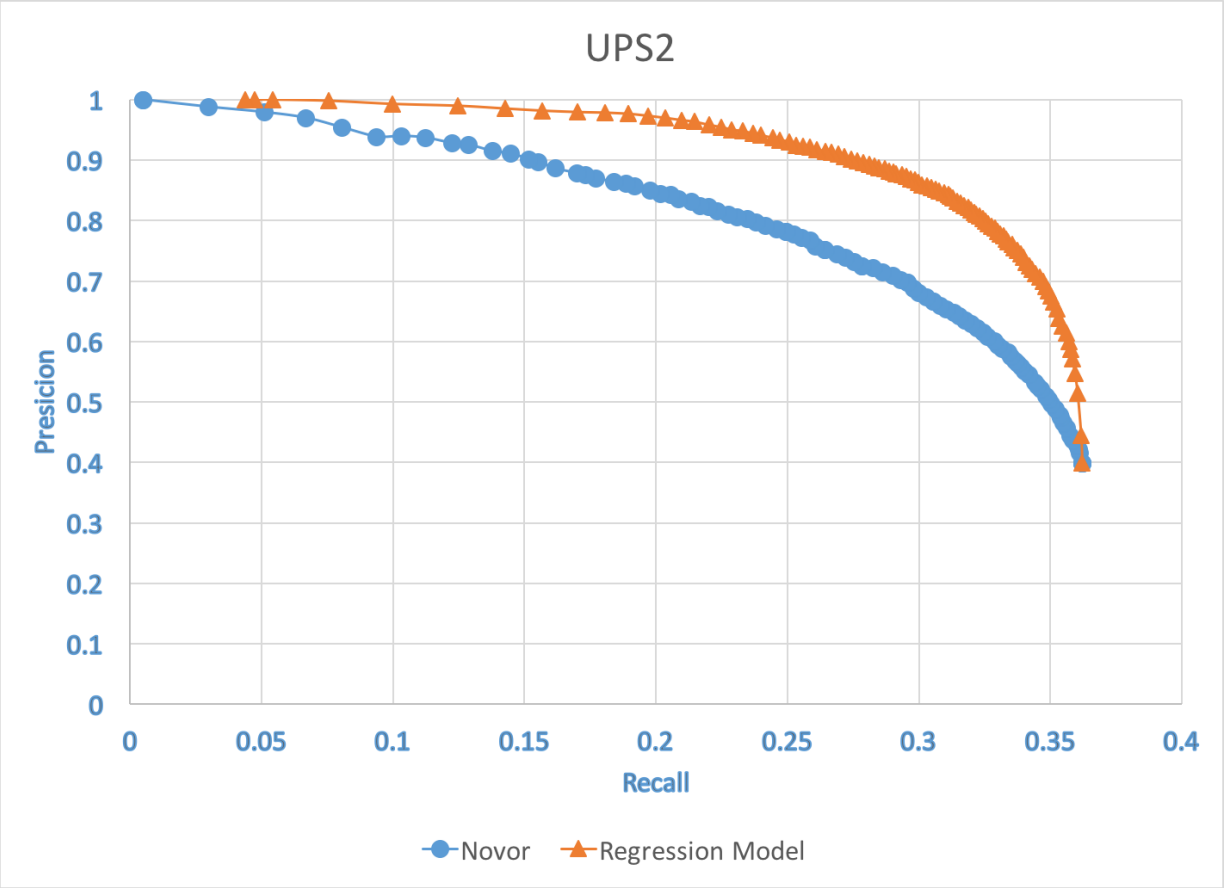

Figure 4.8: Optimization results for dataset UPS2.

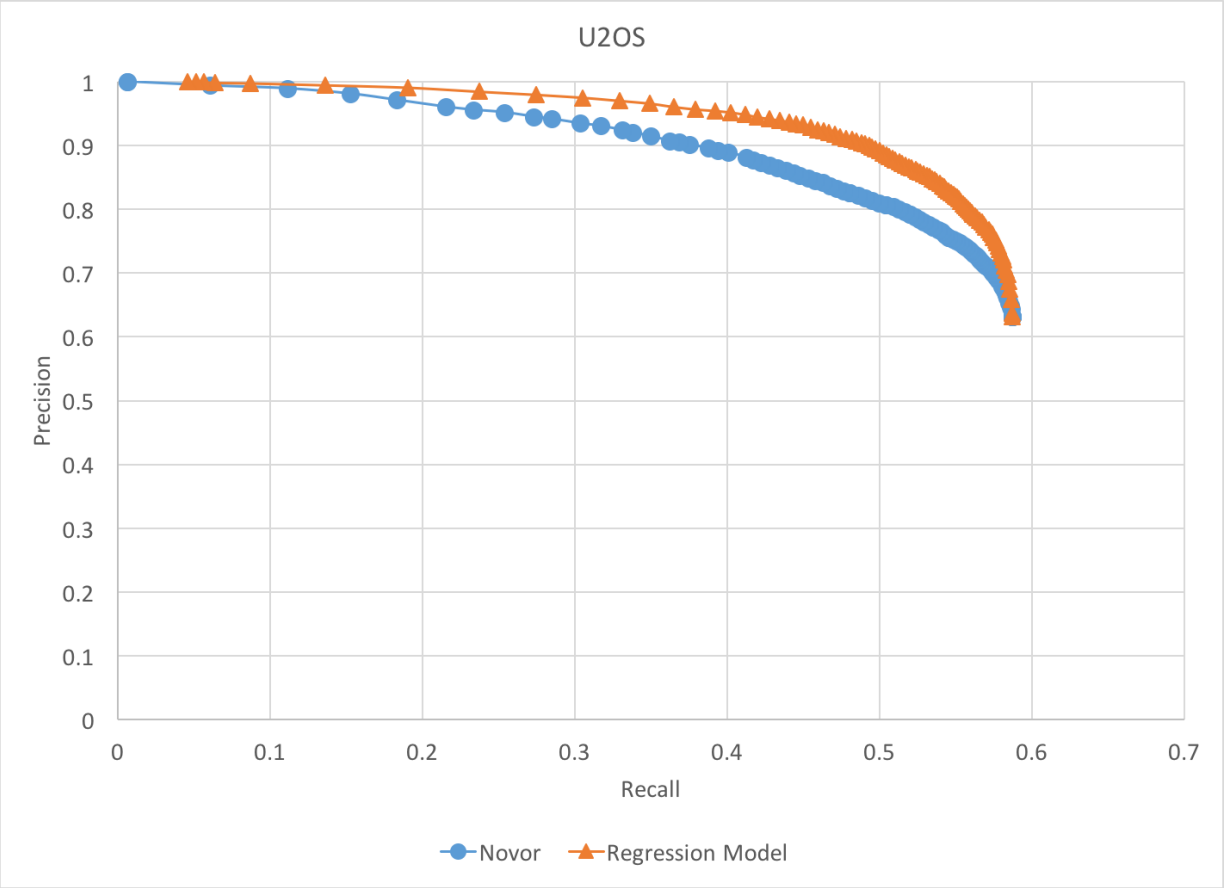

Figure 4.9: Optimization results for dataset U2OS.

into a new logistic regression model to achieve better results. Unlike the previous one, this model is specifically designed to Novor and is instrument dependent. Therefore, the best way to use this model is following the four steps mentioned in 4.3.3.

Features in this further revised novor optimization model shows in table Fig. 4.3.

Feature Name	Explanation
Novor Score	The score Novor gave on an amino acid
Y-ion Ratio Score	Score we are introducing in this chapter
Missing_1	If at least one of the y-ions is missing around this amino acid
Missing_2	If both y-ions are missing around this amino acid
Is_N_term	If this amino acid is at N terminus
Is_C_term	If this amino acid is at C terminus
Peptide_average_score	The average Novor confidences of all residues in this peptide
Max_neighbour_score	The maximal Novor confidence of two neighbours of this residue
Min_neighbour_score	The minimal Novor confidence of two neighbours of this residue
Residue_mass_diff	Mass difference between the mass of two neighbor y-ions and theoretical residue mass
Yions_diff	Minimal mass difference of two neighbor y-ions between theoretical mass and observed mass

Table 4.3: Features in Further Revised Novor Optimization Model.

The first six features are the same with the previous model.

Recall that Novor gives confidence scores to each residue. Peptide_average_score is the average of confidence scores for all residues in this predicted peptide. This shows how confident Novor feels about this whole peptide spectrum match. If Novor gave a high average confidence for the whole peptide, residues in this peptide are more likely to be correctly sequenced.

Max_neighbour_score and Min_neighbour_score are the confidence scores on both neighbour residues. If the neighbours of this residue are correctly sequenced, this residue will have a high probability to be correct. Minimum and maximum scores are used instead of

left and right because scores themselves are more important than the relative position. For example, confidence 20 at left and confidence 80 at right contributes the same with 80 at left and 20 at right.

The remaining two features are showing the mass differences between theoretical and experiment observed values. These two features will be set to 0.0 is there are at least one neighbour y-ion is missing (i.e. Missing_1 is True). Also, these two features only take y-ions with charge 1 into consideration.

Residue_mass_diff is the difference between the theoretical mass of this residue and the observed mass (i.e. the difference between two observed y-ions around this residue). For these two adjacent y-ions, their observed m/z values will also have some deviations from their theoritical m/z values. So there are two deviations for two y-ions on both sides of this residue, and Yions_diff is the minimal one of them. Note here only one deviation is recorded in the model because the other one can be calculated by Residue_mass_diff and this Yions_diff.

Still, logistic regression model is used because it is simple and fast. This model is implemented by Scikit-learn (sklearn, version 0.17) [61] and tested on a dataset called thermo collected from a different (more precise) instrument other than all datasets mentioned above. The experiment is carried out the same way in 4.3.3 and precision-recall curves were drew to compare. As this instrument is more precise than others, the requirements for a residue x is correctly sequenced to y changed to:

1. $|mass(x) - mass(y)| \leq 0.05$ Da, and
2. the total mass before x and before y differ by at most 0.05 Da

Two precision-recall curves are shown in Fig. 4.10, red curve with triangles presents our new logistic regression model while blue curve with dots presents Novor results. The precision-recall curve of the previous model's optimized results was also drawn which almost coincided with Novor's original curve. Therefore, for this dataset, the previous model can only slightly improve the results because they are already very good. However, this further revised model can still improve it significantly which shows the value of this model.

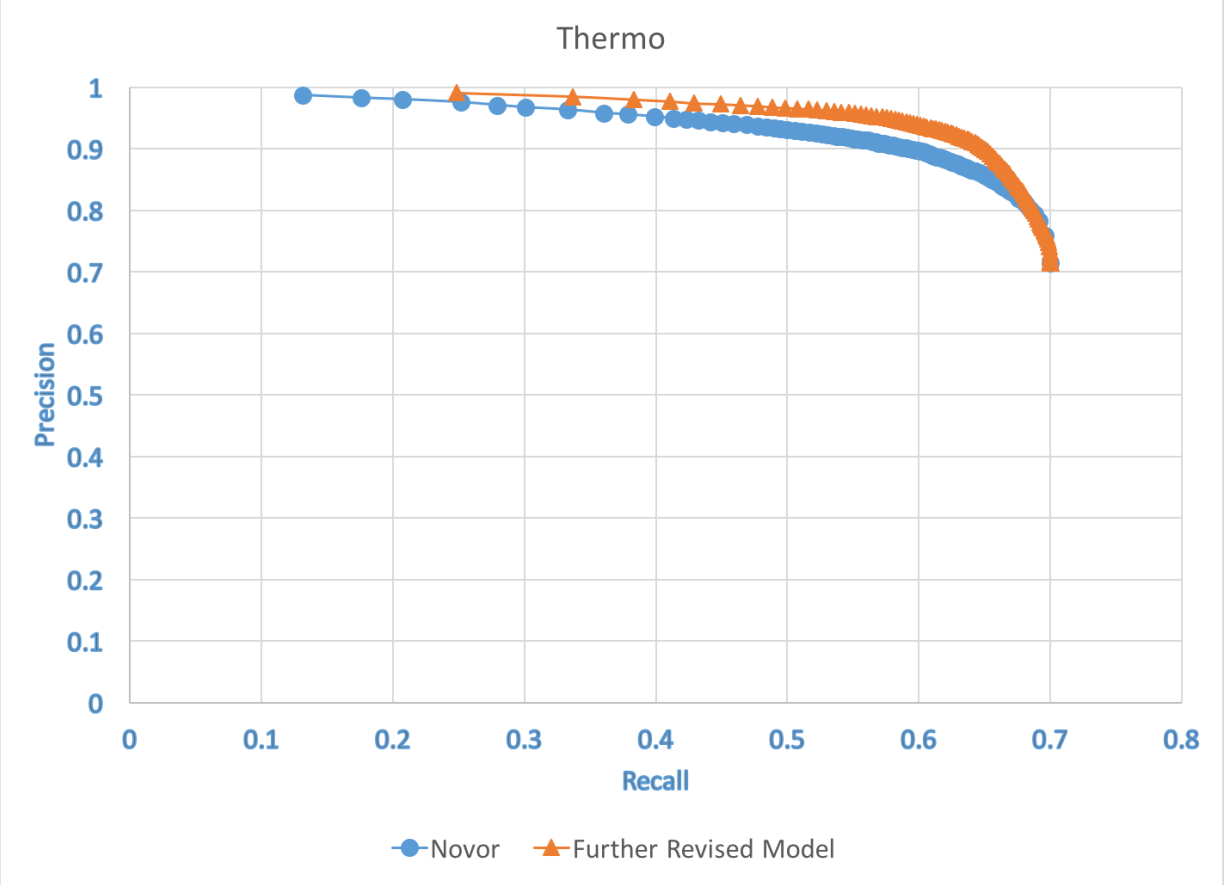

Figure 4.10: Optimization results with further revised Novor optimization model for dataset thermo.

Chapter 5

Internal Fragment Ions Prediction

5.1 Introduction

Most of the significant ions separated and detected by MS/MS are y-ions and b-ions, however, there are still other ones.

Recall that y-ions always end in the rightmost side of peptides (C-terminus) and b-ions always start from the leftmost side of the peptides (N-terminus). All other ions that do not include either the N-terminus or the C-terminus will be called internal fragment ions [24]. These internal fragment ions of protonated peptides arise by charge retention on a portion of the structure excised from the peptide chain. MS/MS can also detect some of these internal fragment ions.

Even though Ballard found internal fragment ions in MS/MS back into 1990s [24], there is no research working on prediction of fragment ions' appearances to the best of our knowledge. This is possibly because internal fragment ions are usually not as significant as y-ions and b-ions, so due to the limitation of MS/MS equipment, it's hard to distinguish internal fragment ions from random noises.

As MS/MS equipment develops fast recently, it is easier today to distinguish valuable ion peaks from random noises. Therefore, it is time to bring internal fragment ions into consideration to help peptide identification.

This chapter will focus on analysing the pattern whether a internal fragment ion will appear in spectrum. Further more, a prediction model will be presented with more than 60% precision under 60% recall. This model performs much better than random guess

which achieves 8% precision under 8% recall. Also, discussion is followed about how peptide identification can benefit from this prediction model.

5.2 Internal Fragment Ions Statistics

In this chapter, WATERS antibody light chain database is used for our analysis. Corresponding peptide prediction results are provided by Novor. In order to keep the data clean, only hand-picked high confident results are kept for further use.

Among all these hand-picked peptide spectrum matches, all possible internal fragments are traversed to see if there is corresponding peak in the spectrum. To be more specific, for every peptide sequences, a start point and a end point are enumerated to form a possible internal fragment. This start point begins its loop from the second left residue and the end point finishes its loop at the second right residue to avoid both y-ions and b-ions. Also, only fragments with length equal or greater than 2 are considered.

In this way, we found 53,109 internal fragment ions detected while other 605,291 fragments are not detected. By these numbers, best random guess strategy is predicting internal fragment ions appear with $Pr = \frac{53109}{53109 + 605291} \approx \frac{1}{13}$. This strategy can provide $\frac{1}{13}$ precision under $\frac{1}{13}$ recall. Since there are not many research carried on this topic, this strategy is the best available and will be used as benchmark in this chapter.

5.3 Prediction Model

$\frac{1}{13}$ precision under $\frac{1}{13}$ recall is far away from satisfaction. In order to better predict whether an internal fragment will appear, a machine learning model will be presented in this section.

An internal fragment can be treated as two cuts on a peptide. After some trial and error, it is found that residues next to both cuts have influence on the appearance of this internal fragment. To be more specific, in peptide $\cdots X_1X_2A\ \ BY_1Y_2\cdots Y_tC\ \ DZ_1Z_2\cdots$, if the fragment was obtained from cuts between A and B, C and D, all A,B,C,D are important for prediction. Besides these residues, N-terminus and C-terminus also have influence on the result, so they are included as well.

Also, it is believed that charge is a very important factor in this prediction. Therefore, the charge of this peptide spectrum match is taken as a feature to train the model. There are three amino acids that have basic side chains at neutral pH which are tending to gain a positive charge. They are arginine (Arg, R), lysine (Lys, K), and histidine (His, H). The numbers of these amino acids are also used as feature in our model.

1	PSM charge	the charge of this peptide spectrum match
2	#K in fragment	number of K in this fragment
3	#R in fragment	number of R in this fragment
4	#H in fragment	number of H in this fragment
5	#K in peptide	number of K in the whole peptide
6	#R in peptide	number of R in the whole peptide
7	#H in peptide	number of H in the whole peptide
8	length	length of the fragment
9	N-term	amino acid at the N-term of this peptide
10	C-term	amino acid at the C-term of this peptide
11	A	amino acid at position A as mentioned before
12	B	amino acid at position B as mentioned before
13	C	amino acid at position C as mentioned before
14	D	amino acid at position D as mentioned before

Table 5.1: Features in Internal Fragment Ions Prediction Model.

There are 14 features in total in our machine learning model as listed in Table 5.1. To simplify our model and optimize our results, all post-translational modifications (PTMs) on amino acids (or termini) will be treated as new (completely different) amino acids. PTM on amino acids usually change some behaviours of amino acids themselves. Therefore, considering them as different amino acids can address these changes and thus improve the prediction results.

A random forest regression model [64] is applied as our prediction model. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

In our specific situation, regression models are needed because we want flexible precision and recall trade off and regressions models can provide a confident score for each

prediction. Comparing to decision tree regression model, random forest provides more possible confident score values. This means using random forest model will provide more trade-off options and make the precision recall curve more smooth. Also, random forest models usually have better prediction results than a simple decision tree model.

5.4 Experiment Results and Important Features

Recall that there are 53,109 positive (internal fragment ions detected) samples and 605,291 negative (fragments not detected) samples in the WATERS antibody light chain database. 90% of them are randomly picked as training data, while the other 10% are used to test and draw a precision recall curve.

Python is used to implement this experiment including data processing, training and testing. Scikit-learn (sklearn, version 0.17) is as the random forest regression model [61].

Training and testing takes less than 5 minutes on a MacBook Pro laptop computer (Retina, Mid-2015, 2.2GHz Intel Core i7, 16 GB RAM).

5.4.1 Experiment Results

Fig. 5.1 shows the precision recall curve of this random forest model. Recall that the area under the precision recall curve can be used as a measurement for performance of the model. Our model can reach nearly 60% precision under 60% recall. This curve covers about 5 times of areas against the best random guess strategy, whose precision-recall curve is a constant line with precision equals to $\frac{1}{13}$.

Even though it is not perfect, this result, which achieved by such simple model and in such short time, can still show us that the appearances of internal fragment ions are predictable.

5.4.2 Important Features

In addition to the test results, feature importances of the model are also recorded to provide valuable to researchers.

- Among all those features, length of the fragment is the most important one, the longer the fragment is, the more likely it tends to be detected.

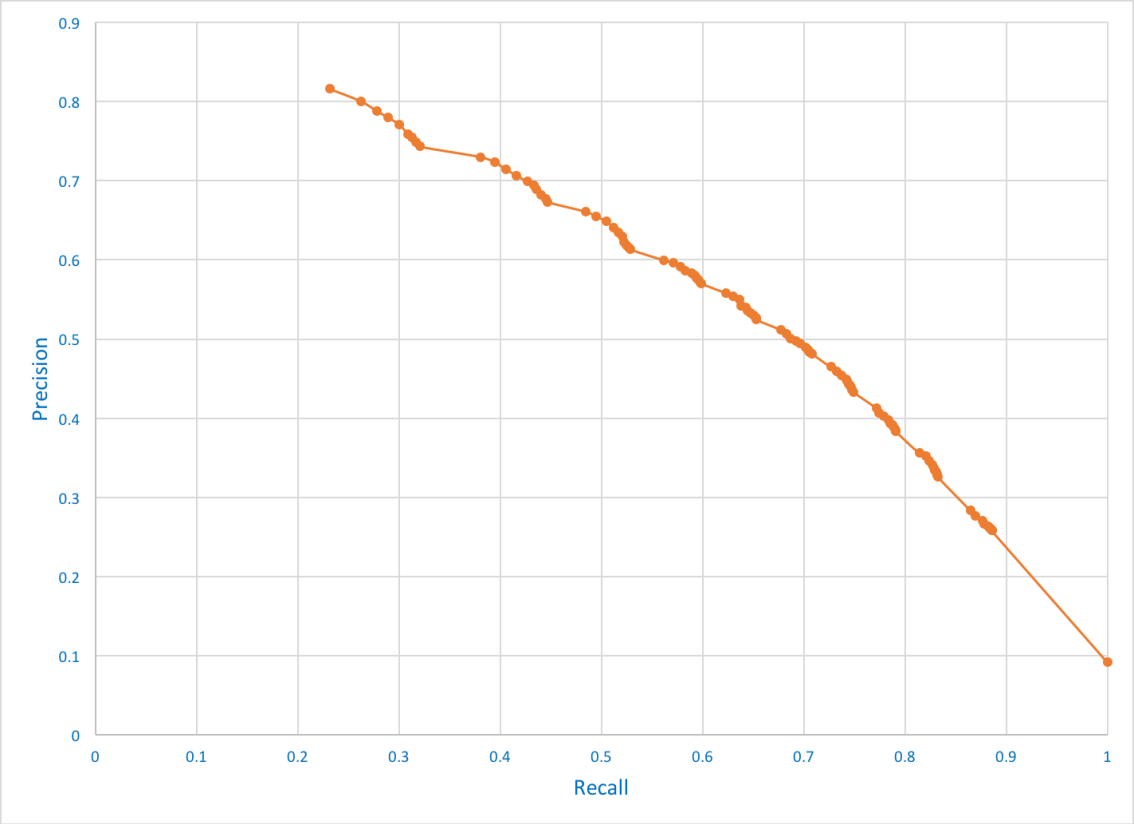

Figure 5.1: Internal fragment ion prediction model precision-recall curve.

- Also, the charge of the spectrum and the number of amino acid KRH in the fragment are very important.
- If the N-terminus amino acid is S or D, or the C-terminus amino acid is K, this peptide will have higher probability to generate internal fragment ions.
- An amino acid P near any one of the cuts causes more internal fragment ions, especially at the position B we mentioned before.
- Generally speaking, residues at position A and B are more important than residues at C and D. This means the start of the internal fragment ions are more important for their appearances.

Not all of these knowledge has research papers to support it by far. However, they can still provide a brief idea to researchers about the patterns of internal fragments ions.

5.5 Applications and Discussions

As this model performs much better than random guesses, it can already contribute to peptide identification. A score feature can be built using this model, and it can work similar as y-ion ratio score to re-rank both database search and de novo sequencing results.

For peptide sequencing results from database search methods, given a peptide spectrum match, a new score feature base on whether internal fragment ions are presented can be provided. Basically, for each single internal fragment, this score will increase if our prediction matches the appearance of this internal fragment, while it will decrease if they do not match. Since we have $O(n^2)$ of internal fragments where n is the length of peptide, this score function can ignore some less confident predictions and only take highly confident ones into consideration.

De novo sequencing usually have some candidates for one residue. After de novo algorithms determine some more confident residues, our new score function can help to determine those less confident ones. Since there are already some residues, inserting a new one will composite more internal fragments. Therefore, candidates can be re-ranked according to the appearances of internal fragments composited by themselves. If our prediction model match most of the appearances of new composited internal fragments, this corresponding candidate is more possible to be the correct one.

However, there are some limitations on this model. First of all, the precision recall curve is not perfect and there is still room to improve. Also, this model only predict internal fragment ions' appearances, so it is only a binary results. A prediction of ion peak abundances might be more helpful. However, both absolute abundances and relative abundances can be influenced by a lot of factors, so they are extremely hard to predict.

Also, appearances of internal fragment ion depend on MS/MS equipments. So this machine learning model only works when predicting spectrums obtained from the same (or similar) experiment equipments. Each time predictions are needed for a new experiment environment, previous peptide sequencing results are needed to train a new model.

In conclusion, despite some limitations, we provided a machine learning model to predict whether an internal fragment ion will appear in a corresponding spectrum.This model can be used to improve peptide sequencing results, for both database search and de novo algorithms. By providing such a model, we showed that internal fragment ions' appearances are predictable. As a pilot work, this also shows that this problem is valuable and is worth further research.

Chapter 6

Conclusions and Future Works

6.1 Conclusions

Peptide sequencing is a long lasting problem In this thesis, we are trying to improve peptide sequencing results using machine learning ideas and techniques, taking advantage of the high-quality spectrum library. As main results, two new features that can help peptide identification are introduced.

Our first contribution is about adjacent y-ion ratios. We presented a method to obtain the adjacent y-ion ratio distributions and use it as a scoring feature to improve the scoring functions for peptide identification. The experimental results demonstrate that the inclusion of this scoring feature could significantly improve the performance of X!Tandem and Novor, for database search and de novo sequencing, respectively. The scoring feature is general enough to be incorporated in other peptide sequencing software. At the same time, based on this scoring feature, we proposed a new machine learning model for Novor and it achieves even better results.

The other contribution is about internal fragment ions. We proposed a prediction model for the appearances of internal fragment ions. To the best of our knowledge, this is a pilot work on this problem and it proves that they are indeed predictable. As nowadays most peptide sequencing software only take y-ions and b-ions into consideration, this work will provide them with more options and information. Similar as adjacent y-ion ratio score, this model can also be used to build a score function and help all peptide sequencing software, but the details are left for future work.

6.2 Future Works

In this thesis, only adjacent y-ion ratios are introduced. It is straightforward to apply this idea to b-ions. However, a larger peptide library may be needed to obtain sufficient b-ions data to run a similar process as adjacent y-ions.

Also, the qualify differences of spectra in the spectrum library is not considered when running the statistics. All of the spectra are treated equally even though some spectra might have significantly higher quality. For example, spectra with relatively high absolute ion abundances should be more important than other ones. A simple solution to this problem would be have a weighing function to all of these spectra, however, details about the function is still missing.

As the prediction of internal ions' appearances is only a pilot work, there are a lot of future work could be carried on this topic. Finding more important features and revising the model to obtain higher precision is definitely one of them. And as mentioned in Chapter 5, our model only predict the ions' appearances while the prediction of ion's abundances could be more helpful. Also, details about applying this prediction model to de novo sequencing are still missing. The method mentioned before is a depth-first search algorithm and may face exponential explosion if the peptide is too long. Greedy algorithm may be useful but its performance still needs to be tested.

References

[1] R. Craig and R. C. Beavis, "A method for reducing the time required to match protein sequences with mass spectra," *Rapid Communications in Mass Spectrometry*, vol. 17, no. 20, pp. 2310–2316, 2003.

[2] R. Craig and R. C. Beavis, "Tandem: matching proteins with tandem mass spectra," *Bioinformatics*, vol. 20, no. 9, pp. 1466–1467, 2004.

[3] B. Ma, "Novor: real-time peptide de novo sequencing software," *Journal of the American Society for Mass Spectrometry*, vol. 26, no. 11, pp. 1885–1894, 2015.

[4] S. Sun, F. Yang, Q. Yang, H. Zhang, Y. Wang, D. Bu, and B. Ma, "Ms-simulator: predicting y-ion intensities for peptides with two charges based on the intensity ratio of neighboring ions," *Journal of Proteome Research*, vol. 11, no. 9, pp. 4509–4516, 2012.

[5] B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby, and G. Lajoie, "Peaks: powerful software for peptide de novo sequencing by tandem mass spectrometry," *Rapid Communications in Mass Spectrometry*, vol. 17, no. 20, pp. 2337–2342, 2003.

[6] A. Frank and P. Pevzner, "Pepnovo: de novo peptide sequencing via probabilistic network modeling," *Analytical Chemistry*, vol. 77, no. 4, pp. 964–973, 2005.

[7] H. Chi, R.-X. Sun, B. Yang, C.-Q. Song, L.-H. Wang, C. Liu, Y. Fu, Z.-F. Yuan, H.-P. Wang, S.-M. He, *et al.*, "pnovo: de novo peptide sequencing and identification using hcd spectra," *Journal of Proteome Research*, vol. 9, no. 5, pp. 2713–2724, 2010.

[8] J. K. Eng, A. L. McCormack, and J. 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*, vol. 5, no. 11, pp. 976–989, 1994.

[9] J. S. Cottrell and U. London, "Probability-based protein identification by searching sequence databases using mass spectrometry data," *Electrophoresis*, vol. 20, no. 18, pp. 3551–3567, 1999.

[10] S. Kim, N. Mischerikow, N. Bandeira, J. D. Navarro, L. Wich, S. Mohammed, A. J. Heck, and P. A. Pevzner, "The generating function of cid, etd, and cid/etd pairs of tandem mass spectra: applications to database search," *Molecular & Cellular Proteomics*, vol. 9, no. 12, pp. 2840–2852, 2010.

[11] D. Li, Y. Fu, R. Sun, C. X. Ling, Y. Wei, H. Zhou, R. Zeng, Q. Yang, S. He, and W. Gao, "pfind: a novel database-searching software system for automated peptide and protein identification via tandem mass spectrometry," *Bioinformatics*, vol. 21, no. 13, pp. 3049–3050, 2005.

[12] J. Zhang, L. Xin, B. Shan, W. Chen, M. Xie, D. Yuen, W. Zhang, Z. Zhang, G. A. Lajoie, and B. Ma, "Peaks db: de novo sequencing assisted database search for sensitive and accurate peptide identification," *Molecular & Cellular Proteomics*, vol. 11, no. 4, pp. M111–010587, 2012.

[13] L. Käll, J. D. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss, "Semi-supervised learning for peptide identification from shotgun proteomics datasets," *Nature Methods*, vol. 4, no. 11, p. 923, 2007.

[14] N. I. of Standards and Technology, "NIST Libraries of Peptide Tandem Mass Spectra." http://chemdata.nist.gov/dokuwiki/doku.php?id=peptidew:start, 2014.

[15] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, vol. 398. John Wiley & Sons, 2013.

[16] R. V. Gold, K. Loening, A. McNaught, and P. Sehmi, *Compendium of Chemical Terminology. IUPAC*. JSTOR, 1987.

[17] W. Reusch, "Peptides & Proteins." https://www2.chemistry.msu.edu/faculty/reusch/virttxtjml/protein2.htm, 2013.

[18] O. N. Jensen, "Modification-specific proteomics: characterization of post-translational modifications by mass spectrometry," *Current Opinion in Chemical Biology*, vol. 8, no. 1, pp. 33–41, 2004.

[19] Q. Hu, R. J. Noll, H. Li, A. Makarov, M. Hardman, and R. Graham Cooks, "The orbitrap: a new mass spectrometer," *Journal of Mass Spectrometry*, vol. 40, no. 4, p. 430443, 2005.

[20] Wang, Rong, "Protein de novo sequencing," Master's thesis, 2016.

[21] A. D. McNaught and A. D. McNaught, *Compendium of Chemical Terminology*, vol. 1669. Blackwell Science Oxford, 1997.

[22] T. F. Scientific, "Overview of Mass Spectrometry for Protein Analysis." https://www.thermofisher.com/ca/en/home/life-science/protein-biology/protein-biology-learning-center/protein-biology-resource-library/pierce-protein-methods/overview-mass-spectrometry.html.html, 2016.

[23] "b and y ions." http://www.ionsource.com/tutorial/DeNovo/b_and_y.htm, 2016.

[24] K. D. Ballard and S. J. Gaskell, "Sequential mass spectrometry applied to the study of the formation of internal fragment ions of protonated peptides," *International Journal of Mass Spectrometry and Ion Processes*, vol. 111, pp. 173–189, 1991.

[25] N. Hatano and T. Hamada, "Proteome analysis of pitcher fluid of the carnivorous plant nepenthes alata," *The Journal of Proteome Research*, vol. 7, no. 2, pp. 809–816, 2008.

[26] V. L. Viala, D. Hildebrand, M. Trusch, R. K. Arni, D. C. Pimenta, H. Schlüter, C. Betzel, and P. J. Spencer, "Pseudechis guttatus venom proteome: insights into evolution and toxin clustering," *Journal of Proteomics*, vol. 110, pp. 32–44, 2014.

[27] J. Catusse, J.-M. Strub, C. Job, A. Van Dorsselaer, and D. Job, "Proteome-wide characterization of sugarbeet seed vigor and its tissue specific expression," *Proceedings of the National Academy of Sciences*, vol. 105, no. 29, pp. 10262–10267, 2008.

[28] J. V. Jorrín-Novo, J. Pascual, R. Sánchez-Lucas, M. C. Romero-Rodríguez, M. J. Rodríguez-Ortega, C. Lenz, and L. Valledor, "Fourteen years of plant proteomics reflected in proteomics: Moving from model species and 2de-based approaches to orphan species and gel-free platforms," *Proteomics*, vol. 15, no. 5-6, pp. 1089–1112, 2015.

[29] R. S. Johnson and K. Biemann, "The primary structure of thioredoxin from chromatium vinosum determined by high-performance tandem mass spectrometry," *Biochemistry*, vol. 26, no. 5, pp. 1209–1214, 1987.

[30] L. A. Martin-Visscher, M. J. van Belkum, S. Garneau-Tsodikova, R. M. Whittal, J. Zheng, L. M. McMullen, and J. C. Vederas, "Isolation and characterization of

carnocyclin a, a novel circular bacteriocin produced by carnobacterium maltaromaticum ual307," *Applied and Environmental Microbiology*, vol. 74, no. 15, pp. 4756–4763, 2008.

[31] X. Liu, Y. Han, D. Yuen, and B. Ma, "Automated protein (re) sequencing with ms/ms and a homologous database yields almost full coverage and accuracy," *Bioinformatics*, vol. 25, no. 17, pp. 2174–2180, 2009.

[32] X. Liu, L. J. Dekker, S. Wu, M. M. Vanduijn, T. M. Luider, N. Tolic, Q. Kou, M. Dvorkin, S. Alexandrova, K. Vyatkina, *et al.*, "De novo protein sequencing by combining top-down and bottom-up tandem mass spectra," *Journal of Proteome Research*, vol. 13, no. 7, pp. 3241–3248, 2014.

[33] C. Liu, B. Yan, Y. Song, Y. Xu, and L. Cai, "Peptide sequence tag-based blind identification of post-translational modifications with point process model," *Bioinformatics*, vol. 22, no. 14, pp. e307–e313, 2006.

[34] S. Tanner, H. Shu, A. Frank, L.-C. Wang, E. Zandi, M. Mumby, P. A. Pevzner, and V. Bafna, "Inspect: identification of posttranslationally modified peptides from tandem mass spectra," *Analytical Chemistry*, vol. 77, no. 14, pp. 4626–4639, 2005.

[35] X. Han, L. He, L. Xin, B. Shan, and B. Ma, "Peaksptm: mass spectrometry-based identification of peptides with unspecified modifications," *Journal of Proteome Research*, vol. 10, no. 7, pp. 2930–2936, 2011.

[36] J. E. Elias and S. P. Gygi, "Target-decoy search strategy for mass spectrometry-based proteomics," *Proteome Bioinformatics*, pp. 55–71, 2010.

[37] M. Bern, B. S. Phinney, and D. Goldberg, "Reanalysis of tyrannosaurus rex mass spectra," *Journal of Proteome Research*, vol. 8, no. 9, pp. 4328–4332, 2009.

[38] L. J. Everett, C. Bierl, and S. R. Master, "Unbiased statistical analysis for multi-stage proteomic search strategies," *Journal of Proteome Research*, vol. 9, no. 2, pp. 700–707, 2010.

[39] M. Bern and Y. J. Kil, "Comment on unbiased statistical analysis for multi-stage proteomic search strategies," *Journal of Proteome Research*, vol. 10, no. 4, pp. 2123–2127, 2011.

[40] D. T. Duncan, R. Craig, and A. J. Link, "Parallel tandem: a program for parallel processing of tandem mass spectra using pvm or mpi and x! tandem," *Journal of Proteome Research*, vol. 4, no. 5, pp. 1842–1847, 2005.

[41] R. D. Bjornson, N. J. Carriero, C. Colangelo, M. Shifman, K.-H. Cheung, P. L. Miller, and K. Williams, "X!! tandem, an improved method for running x! tandem in parallel on collections of commodity computers," *The Journal of Proteome Research*, vol. 7, no. 1, pp. 293–299, 2007.

[42] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, ACM, 1992.

[43] Z. Zhang, "Prediction of low-energy collision-induced dissociation spectra of peptides," *Analytical Chemistry*, vol. 76, no. 14, pp. 3908–3922, 2004.

[44] Z. Zhang, "Prediction of low-energy collision-induced dissociation spectra of peptides with three or more charges," *Analytical Chemistry*, vol. 77, no. 19, pp. 6364–6373, 2005.

[45] S. J. Barton, S. Richardson, D. N. Perkins, I. Bellahn, T. N. Bryant, and J. C. Whittaker, "Using statistical models to identify factors that have a role in defining the abundance of ions produced by tandem ms," *Analytical Chemistry*, vol. 79, no. 15, pp. 5601–5607, 2007.

[46] Y. Lin, Y. Qiao, S. Sun, C. Yu, G. Dong, and D. Bu, "A fragmentation event model for peptide identification by mass spectrometry," in *Research in Computational Molecular Biology*, pp. 154–166, Springer, 2008.

[47] B. Paizs and S. Suhai, "Towards understanding some ion intensity relationships for the tandem mass spectra of protonated peptides," *Rapid Communications in Mass Spectrometry*, vol. 16, no. 17, pp. 1699–1702, 2002.

[48] F. Schütz, E. Kapp, R. Simpson, and T. Speed, "Deriving statistical models for predicting peptide tandem ms product ion intensities," 2003.

[49] A. M. Frank, "Predicting intensity ranks of peptide fragment ions," *Journal of Proteome Research*, vol. 8, no. 5, pp. 2226–2240, 2009.

[50] S. Sun, C. Yu, Y. Qiao, Y. Lin, G. Dong, C. Liu, J. Zhang, Z. Zhang, J. Cai, H. Zhang, *et al.*, "Deriving the probabilities of water loss and ammonia loss for amino acids from tandem mass spectra," *Journal of Proteome Research*, vol. 7, no. 01, pp. 202–208, 2007.

[51] S. Li, R. J. Arnold, H. Tang, and P. Radivojac, "On the accuracy and limits of peptide fragmentation spectrum prediction," *Analytical Chemistry*, vol. 83, no. 3, pp. 790–796, 2010.

[52] H. Lam, E. Deutsch, J. Eddes, J. Eng, N. King, S. Yang, J. Roth, L. Kilpatrick, P. Neta, S. Stein, *et al.*, "Spectrast: An open-source ms/ms spectramatching library search tool for targeted proteomics," in *Poster at 54th ASMS Conference on Mass Spectrometry*, 2006.

[53] V. Dančík, T. A. Addona, K. R. Clauser, J. E. Vath, and P. A. Pevzner, "De novo peptide sequencing via tandem mass spectrometry," *Journal of Computational Biology*, vol. 6, no. 3-4, pp. 327–342, 1999.

[54] Y. Zhang, B. R. Fonslow, B. Shan, M.-C. Baek, and J. R. Yates III, "Protein analysis by shotgun/bottom-up proteomics," *Chemical Reviews*, vol. 113, no. 4, pp. 2343–2394, 2013.

[55] A. Frank, S. Tanner, V. Bafna, and P. Pevzner, "Peptide sequence tags for fast database search in mass-spectrometry," *Journal of Proteome Research*, vol. 4, no. 4, pp. 1287–1295, 2005.

[56] D. A. S. Fraser, "Nonparametric methods in statistics.," 1956.

[57] R. T. Clemen and R. L. Winkler, "Combining probability distributions from experts in risk analysis," *Risk Analysis*, vol. 19, no. 2, pp. 187–203, 1999.

[58] K. J. Kirkwood, Y. Ahmad, M. Larance, and A. I. Lamond, "Characterization of native protein complexes and protein isoform variation using size-fractionation-based quantitative proteomics," *Molecular & Cellular Proteomics*, vol. 12, no. 12, pp. 3851–3873, 2013.

[59] P. Zhang, D. Kirby, C. Dufresne, Y. Chen, R. Turner, S. Ferri, D. P. Edward, J. E. Eyk, and R. D. Semba, "Defining the proteome of human iris, ciliary body, retinal pigment epithelium, and choroid," *Proteomics*, vol. 16, no. 7, pp. 1146–1153, 2016.

[60] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 215–242, 1958.

[61] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[62] E. Coyaud, M. Mis, E. M. Laurent, W. H. Dunham, A. L. Couzens, M. Robitaille, A.-C. Gingras, S. Angers, and B. Raught, "Bioid-based identification of skp cullin f-box (scf) β-trcp1/2 e3 ligase substrates," *Molecular & Cellular Proteomics*, vol. 14, no. 7, pp. 1781–1795, 2015.

[63] C. Vogel, R. de Sousa Abreu, D. Ko, S.-Y. Le, B. A. Shapiro, S. C. Burns, D. Sandhu, D. R. Boutz, E. M. Marcotte, and L. O. Penalva, "Sequence signatures and mrna concentration can explain two-thirds of protein abundance variation in a human cell line," *Molecular Systems Biology*, vol. 6, no. 1, p. 400, 2010.

[64] T. K. Ho, "Random decision forests," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1, pp. 278–282, IEEE, 1995.