CHARACTERIZATION AND DE NOVO SEQUENCING OF MULTI-CHARGE MS/MS SPECTRA

CHONG KET FAH
M.Sc., NUS

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2010

CHARACTERIZATION AND DE NOVO SEQUENCING OF MULTI-CHARGE MS/MS SPECTRA

CHONG KET FAH
M.Sc., NUS

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2010

Acknowledgement

First of all, I would like to thank God for literally carrying me all the way through in this long and often frustrating journey in the pursuit of knowledge. Next I would like to thank my supervisor A/P Leong Hon Wai for having uncommon patience with me as he taught me what true research is, and as I struggled to understand and apply his instructions. To my parents, without you I wouldn't be here today. I'm sorry it took so long, but it's finally done. I would also like to thank my brother for sticking it out with me through these many years. He is the very definition of a true brother. A big thank you to Ning Kang, Max Tan, Melvin Zhang, and Sriganesh Srihari. The fruitful discussions I had with you were invaluable to my research. Last but not least, to all whom I have not named but have in some way or another helped me along the way, accept my heartfelt gratitude.

May God bless you all.

Table of Contents

Summary

This thesis addresses the problem of de novo peptide sequencing. Specifically, the issue addressed here is the sequencing of charge 3 and above spectra, called multi-charge spectra, on CID based mass spectrometer machines. We show in this thesis that integrating higher charge ion-types (charge 3 and above) for multi-charge spectra and introducing a novel algorithm for denovo sequencing can help in obtaining better sequencing results.

Current algorithms mainly focus on sequencing peptides for charge 1 and 2 data, but do not directly handle multi-charge spectra. This is because of the additional challenges posed by including them. These challenges includes the increase in problem size (number of pseudo-peaks to be considered), the increase in the noise level caused by these additional pseudo-peaks, and also the increase in the complexity of the resulting sequencing problem. These challenges to sequencing multi-charge spectra lead to two questions being posed by Pavel Pevzner. Namely, are there higher charged peaks and if so do they increase the percentage of recoverable peptides (portions of the peptides that are "supported" by peaks), and can we devise better sequencing algorithms that consider these higher charge peaks?

In this thesis, we answer both these questions. To answer the first question, we first did a characterization study that showed higher charge peaks either increases the upperbound on the percentage of recoverable peptides by explaining fragmentation points which are not explained by lower charge peaks, or by becoming supporting peaks for fragmentation points already explained by lower charge peaks.

In order to properly model higher charge peaks, we extend the notion of the extended spectrum to include pseudo-peaks of ion-types with higher charges. For a given spectrum, this step properly models the higher charge peaks, but it increases the number of pseudo-peaks to be considered and also increases the noise level. With this extended spectrum model, our characterization study of annotated spectra from the GPM-Amethyst dataset (charge 1-5) shows that there is an increase in the upperbound of the percentage of recoverable peptide by including higher charge peaks. Although the characterization study on ISB and Orbitrap data (both having charge 1-3 data) did not show much increase to the recoverable peptide when

using charge 3 ion-types, we cannot conclude that they are useless since they can still act as supporting ions. This has shown to be true from our sequencing result where using charge 3 ion-types for ISB/ISB2 data results in an improvement in recoverable amino acids of around 1-2% as compared to not using charge 3 data.

While the characterization study shows that considering higher charge peaks can potentially increase the percentage of recoverable peptide, the problem of actually recovering the peptide is still very challenging (the second question). To settle this question, we design a de novo peptide sequencing algorithm called MCPS that considers multi-charge peaks and strong patterns associated with contiguous fragmentation points explained by peaks of the same ion-type. MCPS has been shown to give better or comparable sequencing results with other state-of-the-art algorithms for some sets of multi-charged spectra. Our algorithm makes use of several key ideas: (i) the use of the extended spectrum graph, (ii) filtering of the extended spectrum graph using mono ion-type tags to reduce noise and bring down the size of the problem while still maintaining a good upperbound on the amount of peptide recoverable (iii) using a scoring function that highlight the importance of mono ion-type tag support for a given peptide tag, (iv) a post-processing step that handles problems with competing mono ion-type tags of different ion-types.

Comparing against current state-of-the-art de novo sequencing algorithms PEAKS, PepNovo and Lutefisk, MCPS does the best for charge 3 ISB data and second best for charge 3 ISB2 data. In particular, it can recover 7% more amino acids in the peptide than the second best algorithm, PepNovo, for charge 3 ISB data. We find that the results of MCPS can be used as peptide tag for database search since it includes correctly predicted tags of length ≥ 3 more than 40% of the time for charge 3 ISB and ISB2 data.

List of Tables

List of Figures

Chapter 1

Introduction

Proteins form the very basis of life. They govern a variety of activities in all known organisms, from replication of the genetic code to transporting oxygen, and are generally responsible for regulating the cellular machinery and consequently, the phenotype of an organism. Studying what proteins are present in different organisms and their structure and interactions will help to identify how the body work. Moreover many illnesses and diseases happen due to changes in the proteins and their interactions. Thus studying proteins are an essential part of the life sciences today.

Proteomics is this large-scale study of proteins – their sequences, structures and functions. In proteomics, the identification of protein sequences is very important. However directly identifying proteins is computational complex due to their size. Instead proteins are usually broken down into smaller and more manageable fragments called *peptides* and these are sequenced. Thus peptide sequencing is essential to the identification of their parent proteins. Currently, peptide sequencing is largely done by *tandem mass spectrometry*. In a nutshell, peptides are fragmented in the mass spectrometer machine and these fragments are detected and output as a MS/MS spectra. The analysis of the MS/MS spectra in order to identify the peptide present is by itself a non-trivial problem. This is, in part, because the spectra usually contain lots of noise peaks introduced by impurities or by inaccuracies of the machines. The problem becomes more difficult because many of the peptide fragments do not have corresponding peaks in the spectrum. Deducing peptide sequences from raw MS/MS data is therefore slow and tedious

when done manually. Instead, computational approaches have been developed to help identify peptide sequences. As the volume of data output from mass spectrometers keeps increasing - current machines can generate thousands to hundreds of thousands of spectra in a single within an hour - the need for more accurate and efficient computational methods to peptide sequencing becomes even more essential. Moreover, most of the current algorithms deals with the sequencing of peptide from charge 1 or 2 mass spectrum, but do not do that well for charge 3 and above spectra.

1.1 Brief History of Peptide Sequencing Using Tandem Mass Spectrometry

Protein sequencing had its beginning with the discoveries of Pehr Edman in 1949 and Frederick Sanger in 1955 whereby chemical reagents were used to determine the amino acid sequence of a protein by cleaving each individual amino acid away from the main protein chain. Edman's method especially gained popularity and became known as the now famous *Edman Degradation*.

Mass Spectrometry was already used as a tool for analyzing individual molecules many years before either Edman or Sanger began their work on protein sequencing. From a fairly obscure beginning in the 1800's, mass spectrometry have gone through major evolution in its technology - both hardware and software - and have now become a cornerstone in the field of sequencing. Its first use in protein sequencing was in 1966 when Biemann and his collegues successfully sequenced several oligopeptides containing glycine, alanine, serine, proline, and several other amino acids using a mass spectrometer machine (Biemann et al. [5]).

As mass spectrometers became more robust and more common place in the laboratories during the 80s, sequencing using mass spectrometry began to take off. The advent of *tandem mass spectrometry* which allowed multi-stage fragmentation of the target peptide as well as the development of the two main ionization technology - *ESI (electrospray ionization)* and *MALDI (Matrix-assissted laser desorption/ionization)* in the 90s improved the dynamic range of mass spectrometry greatly and established it has the dominant tool for protein sequencing. All

this led to an explosion of protein sequencing results in the 00's, for example, in 2002 Gavin et al. [25] used mass spectrometry to characterize multiprotein complexes in *Saccharomyces cerevisiae*. Their analysis of these 589 protein assemblies revealed 232 distinct multiprotein complexes. Cellular roles were proposed for 344 proteins, out of which 232 had previously no known functional annotation. In the same year Ho et al. [31] used a method called high-throughput mass spectrometric protein complex identification (HMS-PCI) to systematically identify proteins in *Saccharomyces cerevisiae*. Starting with 10% of the predicted proteins, they were able to cover 25% of the yeast proteome. Since then many more breakthroughs have been made in protein sequencing using mass spectrometry.

1.2 Overview of Entire Process in Peptide Sequencing

We briefly explain the entire process in which peptides are sequenced using tandem mass spectrometry. Figure 1.1 explains the whole process. First a complex mixture containing the protein of interest is fractionated using 2D gel electrophoresis so as to separate out the protein of interest. The protein is then digested using an enzyme, usually trypsin, which will cleave the protein at the carboxyl end of either the lysine or argnine amino acid. This will break the protein into small pieces called peptides. The peptide of interest is then further fractionated using HPLC (high performance liquid chromatography).

This final peptide mixture is then put through the tandem mass spectrometer, where a two stage process occurs. In the first stage, the peptides are ionized (given one or more charge) using ESI (Electrospray ionization), MALDI (matrix-assisted laser dissociation/ionization) or other ionization methods. These ionized peptides called ions are then detected, registering a peak at the particular mass-to-charge ratio (m/z) value they were detected. Depending on the peptide mass and the number of charges deposited, peaks are generated at different m/z values. The height of the peaks produced indicate the abundance of ions at that particular m/z value. A mass spectrum of such peaks is then output.

In the second stage, peptides within a specific narrow mass range is selected based on the 1st mass spectrum output. This is ensure that contaminants and other chemical molecules are not

present in the final output. These peptides then undergo fragmentation through *CID* (*Collision Induced Dissociation*), *EDT* (*Electron Transfer Dissociation*) or other fragmentation methods in a collision cell, where the peptide is usually broken into 2 fragments by bombardment with chemically inert gas like Argon or Helium. One of the fragments is ionized when one or more proton are deposited on them during fragmentation, while the other becomes uncharged.

The mechanism in which a peptide fragments and its fragment becomes ionized in the mass spectrometer using CID is also known as the *Mobile Proton Hypothesis* (Wysocki et al. [68]). In short, the hypothesis states fragmentation of a peptide involves a proton at the cleavage site. Properties like the basicity of the peptide and the amino acid content will affect the way in which the fragmentation occurs, which fragment will get the charge and how much charge is deposited. All this results in different types of ions being produced (discussed in more details in Chapter 2) with different probabilities. Due to many possible competing chemical pathways leading to fragmentation based on the mobile proton hypothesis, much research has gone into discovering exactly how fragmentation occurs in the mass spectrometer by lab experiments (Dongre et al. [15], Tabb. et al. [59], Polce et al. [54], Cox et al. [11], Tang et al. [62]) and machine learning methods (Kapp et al. [33], Elias et al. [16], Sun et al. [57]). (McCormack et al. [45], Zhang [74, 75]) even studied the fragmentation using a quantum mechanical model.

After fragmentation, the fragment ions are detected at a specific m/z value depending on the mass and the amount of charge on the ions as in the 1st stage. This produces the final mass spectrum output. An actual output which has been pre-processed is given in Figure 1.2. This final output is then analyzed using various computational methods (database search, de novo peptide sequencing etc) in order to reconstruct and identify the peptide which produced it.

Bakhtiar and Tse [2] provides a comprehensive introduction and overview to the field of biological mass spectrometry.

1.3 Computational Problems in Peptide Sequencing

Computational methods for peptide sequencing has mostly be concerned with 3 major problems. The first is the sequencing of unknow peptides, the second is the sequencing of known peptides,

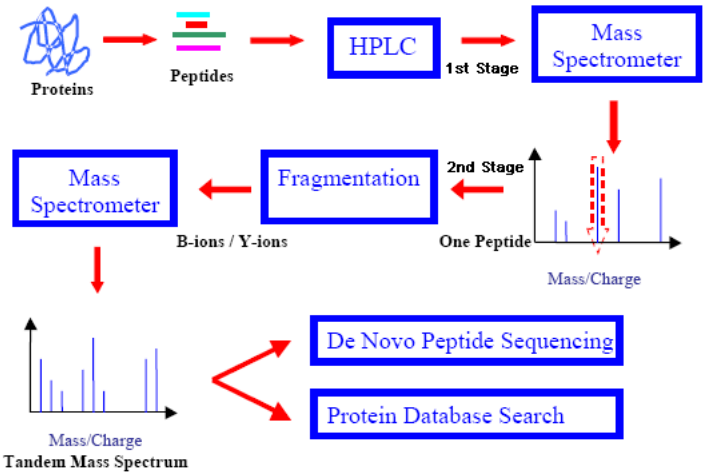


Figure 1.1: Pipeline involved in Peptide Sequencing using Tandem Mass Spectrometry.

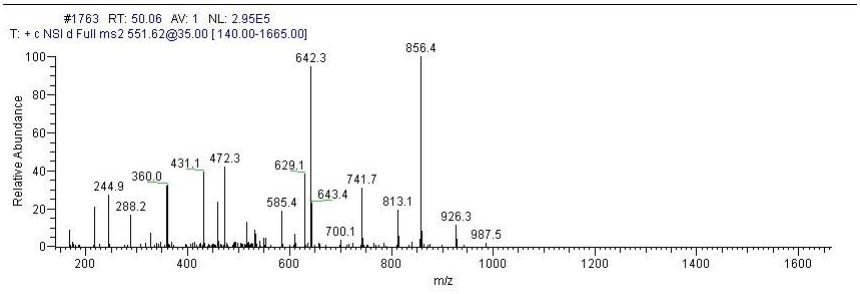


Figure 1.2: Example of a Mass Spectrum

and the third is the sequencing of peptides that have undergone PTM (post-translational modifications).

The **first problem**, *de novo peptide sequencing* or simply *peptide sequencing* tackles the problem of sequencing unknown peptides, that is those peptides which are not already discovered and cataloged. De novo sequencing is used in order to predict full or partial sequences. However, the prediction of peptide sequences from MS/MS spectra is dependent on the quality of the data, and this result in good predicted sequences only for very high quality data, while the results for mid to low quality data can sometimes be very bad. PepNovo (Frank and Pevzner [21]) and PEAKS (Ma et al. [41]) are currently two of the best de novo sequencing algorithms. Others include Lutefisk (Taylor and Johnson [66]) and Sherenga (Dancik et al. [13]). However, many of these algorithms do not explicitly handle higher charged ions (+3 and above) for higher charge spectra (one notable exception is PEAKS which does conversion of multi-charge peaks into their singly-charge equivalent before sequencing). Older versions of Lutefisk worked with singly-charged ions only, but the recent version (Lutefisk 1.0.5) have been updated to work with higher charged ions. Sherenga and PepNovo works with singly- and doubly-charged ions.

The **second problem**, *peptide identification* deals with the problem of sequencing or identifying peptides which are already cataloged. This approach is to perform a database search of such known peptide sequences with the un-interpreted experimental MS/MS data. Even though de novo peptide sequencing can also be applied in this case, database search is usually much more effective for known peptides. A number of such database search algorithms have been described, the most popular being Mascot (Eng et al. [17]) and Sequest (Perkins et al. [49]) . Others include Beavis and Fenyö [4], Pevzner et al. [53], Nathan and Ross [46], Zhang et al. [73].

Database search methods are effective but often give false positives or incorrect identifications. Recently there has been research into a hybrid approach into peptide identification called *tag-based peptide identification* which first uses de novo sequencing to get short candidate peptide fragments called peptide tags, then use these tags for searching databases. This approach have proven to give a higher hit rate then solely relying on database search (Mann and Wilm

[44]). The state-of-the-art softwares based on this approach includes InSpecT (Tanner et al. [63]) and Spider (Han et al. [29]).

The **third problem**, is the sequencing of peptides which have undergone *PTM* (Post-Translational Modification). This is a variation of the above two problems, where a peptide (known or unknown) has its amino acid chemically modified after translation, so that the actual peptide sequence is different from its canonical sequence. Some of these modified amino acids have been cataloged, but many have not, and the identification of such peptides and the modified amino acids have been attempted mainly using database (Pevzner et al. [52], Tsur et al. [67]) and tag-based approaches (Tabb et al. [58], Tanner et al. [63]).

1.4 Focus of Thesis and Key Contributions

The focus on this thesis is on solving the first problem, that is de novo peptide sequencing. Specifically, the issue addressed here is the sequencing of charge 3 and above spectra, called *multi-charge* spectra, on CID based mass spectrometer machines. We show in this thesis that integrating higher charge ion-types (charge 3 and above) for multi-charge spectra and introducing a novel scoring function for denovo sequencing can help in obtaining better sequencing results. Sequencing of multi-charge mass spectra is also highly relevant since CID fragmentation can generate up to charge 5 spectra and there are datasets available like GPM-Amethyst (Craig et al. [12]) dataset which contains spectra up to charge 5. As the throughput of mass spectrum generation increases so will the amount of multi-charge spectra produced.

As mentioned in the introduction, current algorithms mainly focus on sequencing peptides for charge 1 and 2 data, but do not *directly* handle multi-charge spectra. This is because of the additional challenges posed by including them. These challenges includes: (i) increase in problem size (number of pseudo-peaks to be considered), (ii) increase in the noise level caused by these additional pseudo-peaks, and (iii) increase in the complexity of the resulting sequencing problem. In fact, these challenges had led Pevzner Pevzner [51] to pose the following questions: Q1: Are there higher charged peaks and if so, do they increase the percentage of recoverable peptides (portions of the peptides that are "supported" by peaks)? Q2: Can we devise better

sequencing algorithms that consider these higher charge peaks?

In this thesis, we answer both these questions. We first did a characterization study that showed higher charge peaks either increases the percentage of recoverable peptides by explaining fragmentation points which are not explained by lower charge peaks, or by becoming supporting peaks for fragmentation points already explained by lower charge peaks. This work has been published in [8, 9].

We next designed a de novo peptide sequencing algorithm called *MCPS* (mono-chromatic peptide sequencer) that considers higher charge peaks and strong patterns associated with contiguous fragmentation points explained by peaks of the same ion-type. MCPS has been shown to give better or comparable sequencing results with other state-of-the-art algorithms for multi-charged spectra. MCPS has been based on ideas on strong tags published in [8, 9] as well as [48]which is a joint work with the first author Ning Kang. The work on MCPS has led to a paper [7] submitted to RECOMB Satellite Conference on Computational Proteomics 2011, and is still pending review.

In our characterization study, we show that higher charge peaks increases the percentage of recoverable peptides. To properly model higher charge peaks, we extend the notion of the extended spectrum to include pseudo-peaks of ion-types with higher charges. For a given spectrum, this step properly models the higher charge peaks, but it increases the number of pseudo-peaks to be considered and also increases the noise level. With this extended spectrum model, our characterization study of annotated spectra from the GPM-Amethyst dataset (charge 1-5) shows that there is an increase in the percentage of recoverable peptide by including higher charge peaks. Furthermore, this increase is more significant for spectra with bigger charges. For example, on charge 3 GPM spectra, we observed an increase of 12.5% (from 75% to 87.5%) by considering peaks of charge 1-3 as opposed to the traditional method of considering only peaks of charge 1 and 2. On charge 4 GPM spectra, the increase is 27% (from 61% to 88%) by considering peaks of charge 1-4 vs only considering peaks of charge 1 and 2. Although the characterization study on ISB (Keller et al. [34]) and Orbitrap (Tang [61]) data (both having charge 1-3 data) did not show much increase to the recoverable peptide when using charge 3

ion-types, we cannot conclude that they are useless since they can still act as supporting ions. This has shown to be true from our sequencing result where using charge 3 ion-types for ISB data results in an improvement in recoverable amino acids of around 1-2% as compared to not using charge 3 data.

While the characterization study shows increase in the percentage of recoverable peptide by considering higher charge peaks, the problem of actually recovering the peptide is still very challenging. To settle this question, we design a de novo peptide sequencing algorithm called MCPS that considers higher charge peaks and that gives better sequencing results. Our algorithm makes use of several key ideas: (i) the use of the extended spectrum graph, (ii) filtering of the extended spectrum graph using mono ion-type tags to reduce noise and bring down the size of the problem while still maintaining a good upperbound on the amount of peptide recoverable (iii) using a scoring function that highlight the importance of mono ion-type tag support for a given peptide tag, (iv) a post-processing step that handles problems with competing mono ion-type tags of different ion-types.

Comparing against current state-of-the-art de novo sequencing algorithms PEAKS, PepNovo and Lutefisk, MCPS does the best for charge 3 ISB data and second best for charge 3 ISB2 data. In particular, it can recover 7% more amino acids in the peptide than the second best algorithm, PepNovo, for charge 3 ISB data. We find that the results of MCPS can be used as peptide tag for database search since it includes correctly predicted tags of length ≥ 3 more than 40% of the time for charge 3 ISB and ISB2 data.

We briefly describe the ideas presented in MCPS in the following:

We introduce the extended spectrum graph (ESG) that properly represents the (very noisy) extended spectrum. The ESG generalizes the notion of the spectrum graph (introduced by Bartels [3]). In our extended spectrum graph, we represent as a distinct vertex, each pseudo-peak (corresponding to each ion-type annotation/interpretation of a given peak). Thus, each peak "generates" more vertices in the ESG (compared to the traditional spectrum graph) and the ESG also has a higher level of noise.

To deal with the increased noise level, we use the ESG to find monochromatic tags (short

contiguous sequences of pseudo-peaks of the same ion-type annotation) of the more abundant ion-types. Thus, our key idea is that the presence of a sequence of consecutive pseudo-peaks of the same ion-type is a much stronger signal than a sequence of consecutive pseudo-peaks made up of mixed ion-types. We then retain in ESG only those pseudo-peaks that belong to monochromatic tags of a certain minimum length. This preprocessing step allows us to effectively filter off a large proportion of the noise pseudo-peaks from further consideration. This does not mean that vertices of less abundant ion-types are ignored. They are used in a subsequent bridging step to act as a link between monochromatic tags that otherwise cannot be linked together.

A novel scoring function that takes into consideration the stronger signals represented by monochromatic tags by boosting their score (through a multiplicative factor based on length) is then used in the sequencing step to sequence candidate peptides.

After the sequencing, a post-processing step was introduced due to certain situations where monochromatic tags of different ion-types residing in different paths in the extended spectrum graph compete with each other, thus bringing down the quality of the sequencing result. This post-processing step normalizes the score on such tags so as to remove the competition.

1.5 Organization of Thesis

In Chapter 2, we give some background on proteins, then define the problem of peptide sequencing. We next introduce the major class of algorithms used to solve peptide sequencing, called spectrum graph methods. We review some of the major algorithms involved in this class as well as others who use a different technique. We also present algorithms which tackle certain specific sub-problems encountered in peptide sequencing.

In Chapter 3, we define a generalized model for studying multi-charge mass spectra where we introduce the new notion of an extended spectrum, and extend the definition of the theoretical spectrum and the spectrum graph.

In Chapter 4, we use the generalized model defined in Chapter 3 to do a characterization study of 3 dataset, the ISB dataset (Keller et al. [34]), the Orbitrap dataset (Tang [61]), and

the GPM-Amethyst dataset (Craig et al. [12]). The ISB and Orbitrap dataset consists of charge 1-3 spectra, while the GPM dataset consists of charge 1-5 data.

In Chapter 5, we present our new algorithm *MCPS* (mono-chromatic peptide sequencer) for performing de novo sequencing, especially of multi-charge spectra. We first present a novel scoring function that we have developed based on initial ideas of strong tags in Ning et al. [48]. Then we present the major steps in the algorithm, before delving into the details of each step.

In Chapter 6, we first present how we tweaked the parameters involved in MCPS using training sets from the ISB, GPM and ISB2 (Klimek et al. [37]) datasets.

In Chapter 7, we present experimental results comparing between MCPS and 3 other state-of-the-art algorithms - PEAKS (Ma et al. [41]), PepNovo (Frank and Pevzner [21]) and Lutefisk (Taylor and Johnson [66]).

In Chapter 8, we give a conclusion our thesis as well as future work.

Chapter 2

Peptide Sequencing and Literature Survey

In this chapter, we formally define the peptide sequencing problem and give an overview of the various algorithms that has been developed to tackle the problem.

2.1 Background on Proteins

A chain of amino acids is known as a peptide. A protein is basically made up of multiple peptides linked together, and is also known as a polypeptide chain. The amino-acids are the 20 naturally occurring acids, Valine, Leucine, Isoleucine, Methionine, Phenylalanine, Asparagine, Glutamic Acid, Glutamine, Histidine, Lysine, Arginine, Aspartic Acid, Glycine, Alanine, Serine, Threonine, Tyrosine, Tryptophan, Cysteine and Proline. The molecular masses of these amino acids are given in Table 2.1. A protein's amino acid sequence is usually written in the single alphabet amino acid string format. For example in Figure2.1, a protein consisting of the amino acid sequence methonine, aspartic acid, leucine and tyrosine from left to right is represented as MDLY.

Amino acids can be further categorized into 2 category. The first are the hydrophilic or polar residues which are residues that interact favourably with the solvent that the protein is in, and thus are found more often on the surface of the protein protruding outwards into the

Amino Acid (Single Alphabet - 1st 3 Letters - Full Name)	Mono-Isotopic Mass (daltons Da)
A - Ala - Alanine	71.037
C - Cys - Cysteine(unmodified/carboxymethylated)	103.009/161.05
D - Asp - Aspartic Acid	115.027
E - Glu - Glutamic Acid	129.043
F - Phe - Phenylalanine	147.068
G - Gly - Glycine	57.021
H - His - Histidine	137.059
I - Iso - Isoleucine	113.084
K - Lys - Lysine	128.095
L - Leu - Leucine	113.084
M - Met - Methionine	131.040
N - Asp - Asparagine	114.043
P - Pro - Proline	97.053
Q - Glu - Glutamine	128.059
R - Arg - Arginine	156.101
S - Ser - Serine	87.032
T - Thr - Threonine	101.048
V - Val - Valine	99.068
W - Try - Tryptophan	186.079
Y - Tyr - Tyrosine	163.063

Table 2.1: Mono-isotopic Masses of Naturally Occurring Amino Acids. An amino acid can be referred to by its first 3 letters or a single alphabet. The mono-isotopic mass we give here is calculated based on the standard atomic makeup of the amino acid HNCH**R**CO where R is the side-chain (refer to Figure 2.1). Note that Cysteine is usually modified during the preparation process for mass spectrometry so that its mono-isotopic mass defers from the unmodified version.

solvent. The second are the hydrophobic or non-polar residues which interact unfavourably with the solvent and thus are tightly packed together in the interior of the protein. These residues also form what is known as the core of the protein. Amino acids are further composed of 2 parts, the backbone fragment and the side-chain fragment. The chemical makeup and schematic representation of a protein is given in Figure 2.1.

2.2 The Peptide Sequencing Problem

Peptide Let A be the set of amino acids. For an amino acid $a \in A$, $m(a)$ denotes its molecular mass. A peptide $\rho = (a_1, a_2..., a_n)$ is a sequence of amino acids where a_j is the j^{th} amino acid in the sequence. The *parent mass* of the peptide ρ is given by $M = m(\rho) = \sum_{j=1}^{l} m(a_j)$. A *peptide prefix fragment* $\rho_k = (a_1, a_2, ..., a_k)$, for $k \leq n$ is a partial peptide formed from a prefix

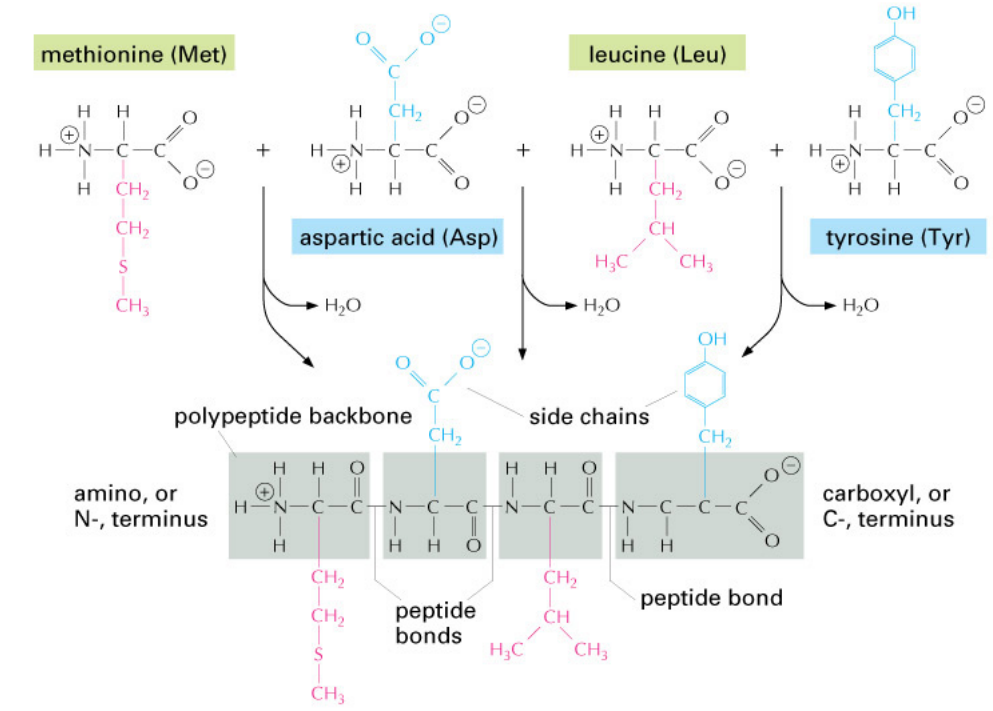


Figure 4-2 Essential Cell Biology, 2/e. (© 2004 Garland Science)

Figure 2.1: Chemical makeup and schematic of a protein. A protein is made up of 20 basic amino acids. In the example a short protein consisting of the amino acid sequence methonine (Met), aspartic acid (Asp), leucine (Leu) and tyrosine (Tyr) is shown. This protein is also referred to by the sequence MDLY (Single alphabet representation of the amino acid). The standard atomic make-up of an amino acid is HNCH**R**CO, where R is the side-chain residue or simply the side-chain, the part which is different for different amino acids and gives each amino acid its unique property. The other atoms make up the backbone portion of the amino acid. A protein is terminated at the left end by an N-terminal (amino terminus) amino acid which has an extra H atom. It is terminated on the right by the C-terminal (carboxyl terminus) amino acid which has an extra OH atoms.

of ρ. The mass of the peptide prefix fragment is $m(\rho_k) = \sum_{j=1}^{k} m(a_j)$, and is also known as the *prefix residue mass* or PRM. A peptide suffix fragment $\overline{\rho_k} = (a_{n-k+1}, ..., a_{n-1}, a_n)$, for $k \leq n$ is a partial peptide formed from a suffix of ρ that has mass $m(\rho_k) = \sum_{j=n-k}^{n} m(a_j)$. The mass of a suffix fragment is also known as the *suffix residue mass* or SRM. The set of all possible prefixes of a peptide forms the *PRM ladder* or *prefix ladder* and similarly the set of all suffixes forms the *SRM ladder* or *suffix ladder* of the peptide. The prefix and suffix ladder forms the "full ladder" of the peptide. Since each *position* $(1, 2...n)$ in the peptide string can define either a prefix or suffix fragment, we call each position a *fragmentation point*. The peptide from which an experimental spectrum is generated is known as the canonical peptide denoted as $\rho*$.

Peptide Fragmentation. An *ion* in our context is basically a charged fragment of the peptide. A peptide is usually fragmented into 2 pieces, one making up the prefix fragment and the other the suffix fragment. Either the prefix or the suffix fragment will be charged but not both. In an experiment, since there are millions of peptide copies, both the suffix and prefix ions will be generated with different probabilities.

Depending on the way the fragmentation occurs, and the terminal on which the charge is deposited, ions of different types are generated. The number of ions of each mass-to-charge ratio can be measured by a detector. The measurement is more intense if there are more ions of that mass-to-charge ratio. A plot of these intensities against the mass-to-charge ratio gives us a spectrum of many peaks, each corresponding to the ions of each mass-to-charge ratio. Figure 2.2 shows the different fragmentations that result in 6 basic ion-types.

Ion-formation. In Figure 2.2, we have a peptide ρ consisting of 4 amino acids represented by the side-chain R, R', R" and R"' and the associated atoms. Fragmentation leading to the formation of all the 6 basic ion-types are shown in the figure. This is due to the possibility of fragmenting at the C-C, C-N and N-C boundaries. The 6 basic ion-types are *a-ion*, *b-ion*, *c-ion*, *x-ion*, *y-ion* and *z-ion*. The a,b,c ion-types are the N-terminal/prefix ions, while the x,y,z ion-types are the C-terminal/suffix ions. The former is formed by the prefix fragments being charged and the latter is formed the suffix fragments being charged. Each prefix ion-type has its counterpart suffix ion-type since the fragmentation is at the same boundary, and only the

deposition of the charge is different. Thus we have a-ion with x-ion, b-ion with y-ion, and c-ion with z-ion.

As an example, the charge 1 (+1) a-ion shown is formed by breaking the peptide-bond between the C-C boundary and deposition of a charge on the prefix fragment. We see that this fragment consists of 2 amino acids. We also see the mass of this fragment is exactly the sum of the mass of the 2 amino acids plus 1 extra H atom (at the N-terminal amino acid) minus the mass of OC (lost to the suffix fragment). In short the PRM of this prefix fragment is $m(\rho_2)$, but the actual mass of the a-ion detected is $m(\rho_2) + 1 - 12 - 16 = m(\rho_2) - 27$. We say that the mass of the detected fragment is $shifted$ from its PRM by -27 Da. The x-ion on the other hand receives the OC atoms and has an extra OH at its C-terminal end. From Figure 2.2, we see that the suffix fragment also has two amino acids. The x-ion thus causes a shift of the SRM $m(\bar{\rho_2})$ by the addition of OCOH which is +45Da. In the example even though the fragmentation was at the 2nd fragmentation point, the shifts will be the same regardless of which position of the peptide is being fragmented.

There are variations on these ion-types based on neutral losses (*additional loss* of a water and/or ammonia molecule) and different number of charges deposited on the ions. A list of the +1 ion-types are shown in Table 2.2 together with the resultant shift away from the PRM or SRM (rounded off to the nearest integer). Note that in the table we give the resultant mass shift. Neutral losses like water and ammonia contribute a shift of -18 Da and -17 Da respectively in addition to the mass shift contributed by the basic ion-type.

We can see that in some cases, e.g b and c-NH_3, the same mass shift is observed, making these ion-types hard to differentiate between each other merely based on their mass shifts, since the same peak in the experimental spectrum can refer to either ion-types. It should be noted that these ions types and their neutral losses are not equally likely to be formed. For example, the presence of z ion-types in low energy CID is questionable and usually not considered in peptide sequencing. The peptide sequencing program PepNovo by Frank and Pevzner [21] for example considers only the ion-types shown in the first column of Table 2.3. The second column shows their probability of being observed in the experimental spectrum.

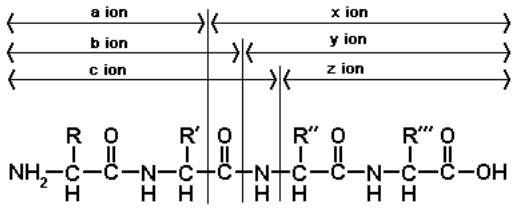


Figure 2.2: Peptide ion formation for the basic ion-types. We see that depending on where fragmentation occurs along the backbone of the peptide, and where the charge is deposited, ions of different can be generated. For example, if the fragmentation occurs at the C-C boundary, it will generate a prefix and a suffix fragment. If the prefix fragment is where the charge was deposited, it generates an a ion. On the other hand if the suffix fragment is where the charge was deposited, it generates an x ion.

Ion-type	Resultant Mass Shift
a	-27
a$-H_2O$	-45
a$-H_2O-H_2O$	-63
a$-NH_3$	-44
a$-NH_3-H_2O$	-62
b	+1
b$-H_2O$	-17
b$-H_2O-H_2O$	-35
b$-NH_3$	-16
b$-NH_3-H_2O$	-34
c	+18
c$-H_2O$	0
c$-H_2O-H_2O$	-18
c$-NH_3$	+1
c$-NH_3-H_2O$	-17
x	+45
x$-H_2O$	+27
x$-H_2O-H_2O$	+9
x$-NH_3$	+28
x$-NH_3-H_2O$	+10
y	+19
y$-H_2O$	+1
y$-H_2O-H_2O$	-17
y$-NH_3$	+2
y$-NH_3-H_2O$	-16

Table 2.2: +1 Ion-types with variation based on neutral losses and their associated resultant mass shift. $-NH_3$ and $-H_2O$ refers to the ammonia and water loss respectively.

Ion-type	Probability of Observation
b	0.83
b$-H_2O$	0.39
b$-NH_3$	0.36
b$-H_2O - H_2O$	0.13
b$-NH_3 - H_2O$	0.12
b^{+2}	0.13
a	0.34
a$-H_2O$	0.17
a$-NH_3$	0.20
y	0.87
y$-H_2O$	0.26
y$-NH_3$	0.24
y$-H_2O - H_2O$	0.11
y$-NH_3 - H_2O$	0.13
y^{+2}	0.23

Table 2.3: Ion-types used by PepNovo. The +2 superscript, e.g b^{+2} indicate charge 2 ions of the given ion-type

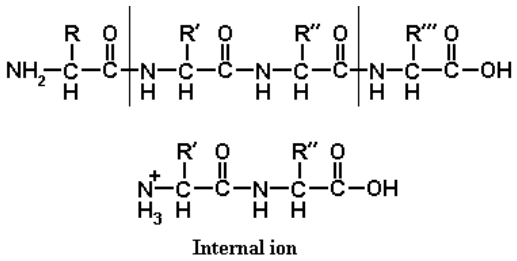


Figure 2.3: Fragmentation resulting in an internal ion

Fragmentations is however usually not so clean and other types of fragments occur. These contribute to peaks in the spectrum and complicate the detection of the N-terminal and C-terminal ion-types. One example of such fragmentations is internal fragmentation where fragmentation occurs at two fragmentation points instead of one, and the resulting middle fragment or "internal" ion is detected. Another example are immonium ions which are internal ions that have lost a CO molecule. Schematics showing the formation of internal fragments and immonium ions are shown in Figure 2.3 and Figure 2.4 respectively. Noise and contaminants can also cause a peak in the experimental spectrum.

Experimental Spectrum. We call the spectrum generated by a mass spectrometer an *experimental spectrum*. A peak in the experimental spectrum S corresponds to the detection of some charged prefix or suffix peptide fragment that results from peptide fragmentation in the mass

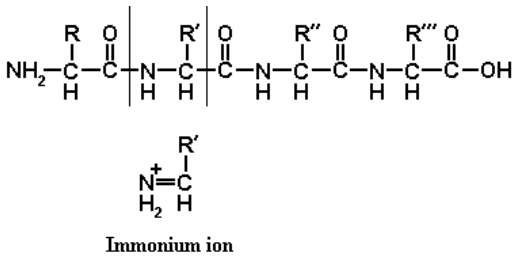


Figure 2.4: Fragmentation resulting in an immonium ion

spectrometer. Each peak p_i in the experimental spectrum S is described by its $intensity(p_i)$ and mass-to-charge ratio $mz(p_i)$. Most experimental spectrum S will give the *precursor ion mass* M' which is the detected parent mass of the ion fragments found in the spectra. In most cases the precursor ion mass $M' \approx M$ the canonical peptide mass, but can sometimes be off by a large margin.

Ion Type. An ion-type is specified by $(z, t, h) \in (\triangle_z \times \triangle_t \times \triangle_h)$, where z is the charge of the ion, t is the basic ion-type and h is the neutral loss incurred by the ion. The (z, t, h)-ion of the peptide fragment q (prefix or suffix fragment) is detected by the mass spectrometer and will produce an observed peak p_i in the experimental spectrum S that has a mass-to-charge ratio of $mz(p_i)$. We say that peak p_i is a *support peak* for the fragment q with *ion-type* (z, t, h) and we also say that the fragment q is *explained* by the peak p_i. An ion-type set is shorted-formed as $\triangle$.

The mass of a fragment q given its corresponding peak p_i and ion-type (z, t, h) can be computed using the shifting function, $Shift$ defined as follows:

$$m(q) = Shift(p_i, (z, t, h)) = mz(p_i) \cdot z - (\delta(t) + \delta(h)) - (z - 1) \tag{2.1}$$

where $\delta(t)$ and $\delta(h)$ are the mass difference associated with the ion-type t and the neutral-loss h, respectively. When an ion has a charge greater than 1, the extra charges come from protons of Hydrogen atoms being deposited on the ion. This increases the mass of the ion by +1 Da for each proton deposited. Thus we need the (z-1) term when considering higher charged ions, in order to discount the mass of the extra protons.

Given an ion of a particular ion-type δ generated for fragmentation point k of peptide ρ, we

can calculate the m/z value of the peak $p_{k\delta}$ registered by the spectrometer by using the $\overline{Shift}$ function which can be obtained from an algebraic manipulation of the $Shift$ function and is defined as

$$mz(p_{k\delta}) = \overline{Shift}(\hat{\rho}_k, (z,t,h)) = \frac{m(\hat{\rho}_k) + (z-1) + (\delta(t) + \delta(h))}{z} \tag{2.2}$$

where $\hat{\rho}_k$ is the prefix fragment ρ_k if δ is an N-terminal ion-type and the suffix fragment $\overline{\rho_k}$ if δ is an C-terminal ion-type.

Theoretical Spectrum. The *theoretical spectrum* is the spectrum of all possible true peaks for a given peptide ρ. The set of true peaks for each prefix fragment ρ_k in the prefix ladder of ρ is the set of ions generated for ρ_k given each ion-type $(z,t,h) \in \triangle$ (calculated using $\overline{Shift}$ function). The set of all true peaks is the union of all the ions generated for the entire prefix ladder.

We define $TS(\rho)$ to be the set of all *possible* observed peaks that may be present in an experimental spectrum for peptide ρ. Namely, $TS(\rho) = \{p$: p is an observed peak for the (z,t,h)-ion of the peptide fragment ρ_k, for all $(z,t,h) \in \triangle$ and $k = 1,..,n\}$.

In peptide sequencing, we are given an experimental spectrum with true peaks and noise and the problem is to try to determine the original peptide ρ that produced the spectrum. Formally the peptide sequencing problem can be defined as

Peptide Sequencing Problem. Given a spectrum S, a set of ion-types $(\triangle_z \times \triangle_t \times \triangle_h)$ and the precursor mass M', find a peptide ρ' of mass M' with the best match to spectrum S.

Dancik et al. [13] addressed the above problem using a simple matching criteria called the *shared peaks count* or *SPC*. In this matching, the theoretical spectrum $TS(\rho')$ for a candidate peptide ρ' is compared with the experimental spectrum S. The number of matching peaks between $TS(\rho')$ and S is the SPC.

An example of SPC is given as follows. Figure 2.5 shows the PRM ladder of AGFAGDDAPR. Figure 2.6 shows an experimental spectrum generated from the canonical peptide AGFAGDDAPR. Assuming an ion-type set of only +1 b- and y-ions gives rise to the theoretical spectrum as shown in Figure 2.7(b). We only annotate the peaks (dotted lines) that matches with those in the

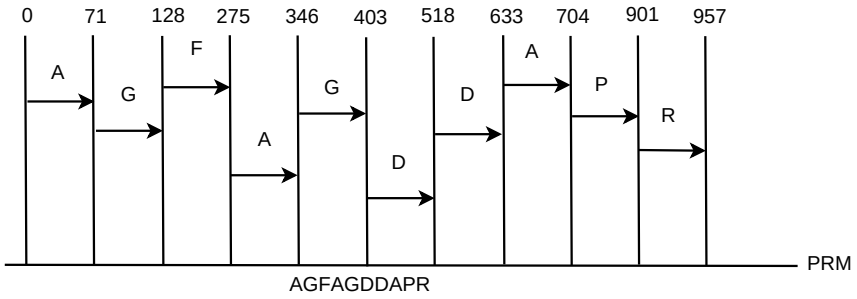


Figure 2.5: PRM ladder for peptide AGFAGDDAPR

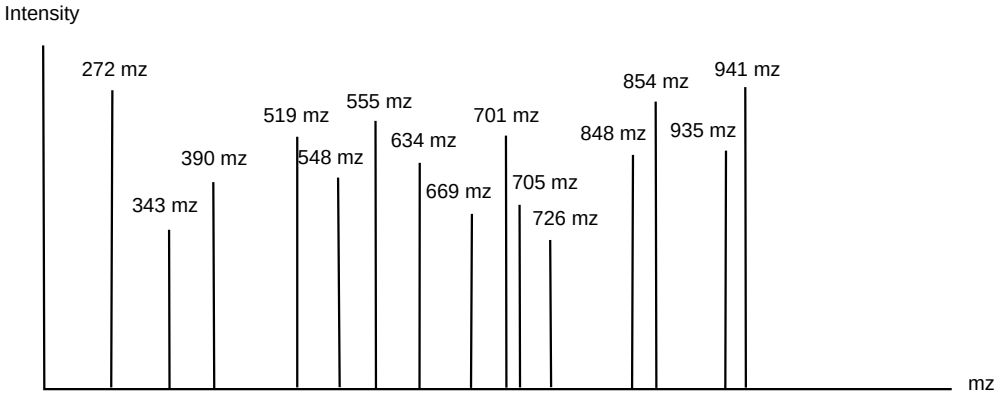


Figure 2.6: Experimental Spectrum for AGFAGDDAPR

experimental spectrum as shown in Figure 2.7(c). These peaks corresponds to fragmentation points of the peptide given the associated ion-type interpretations indicated between 2.7(a) and 2.7(b). These peaks corresponds to the subsequence [128]F[243]DA[253] (values in [] indicates the fragment masses which are not explained, IE not matched to any amino acid sequence). Any candidate peptide including this subsequence will be maximally matched to the canonical peptide.

Using the SPC however does not guarantee that a candidate peptide with a higher SPC score over another candidate will mean that it matches more of the canonical peptide. An example is given as follows. A candidate peptide fragment or a peptide tag (explained in detail in Section 2.3.2.1)containing the subsequence [128]F[243]DA[253] would be [41]SFNEDA[253] as shown in Figure 2.8. Dotted peaks in the experimental spectrum corresponds to the peaks which are interpreted to get [41]SFNEDA[253]. There are 7 such peaks making the SPC 7. Another candidate peptide fragment [35]SQGNPDA[257] given in Figure 2.9 matches 8 peaks in the experimental spectrum which gives an SPC of 8, which is higher than that of [41]SFNEDA[253].

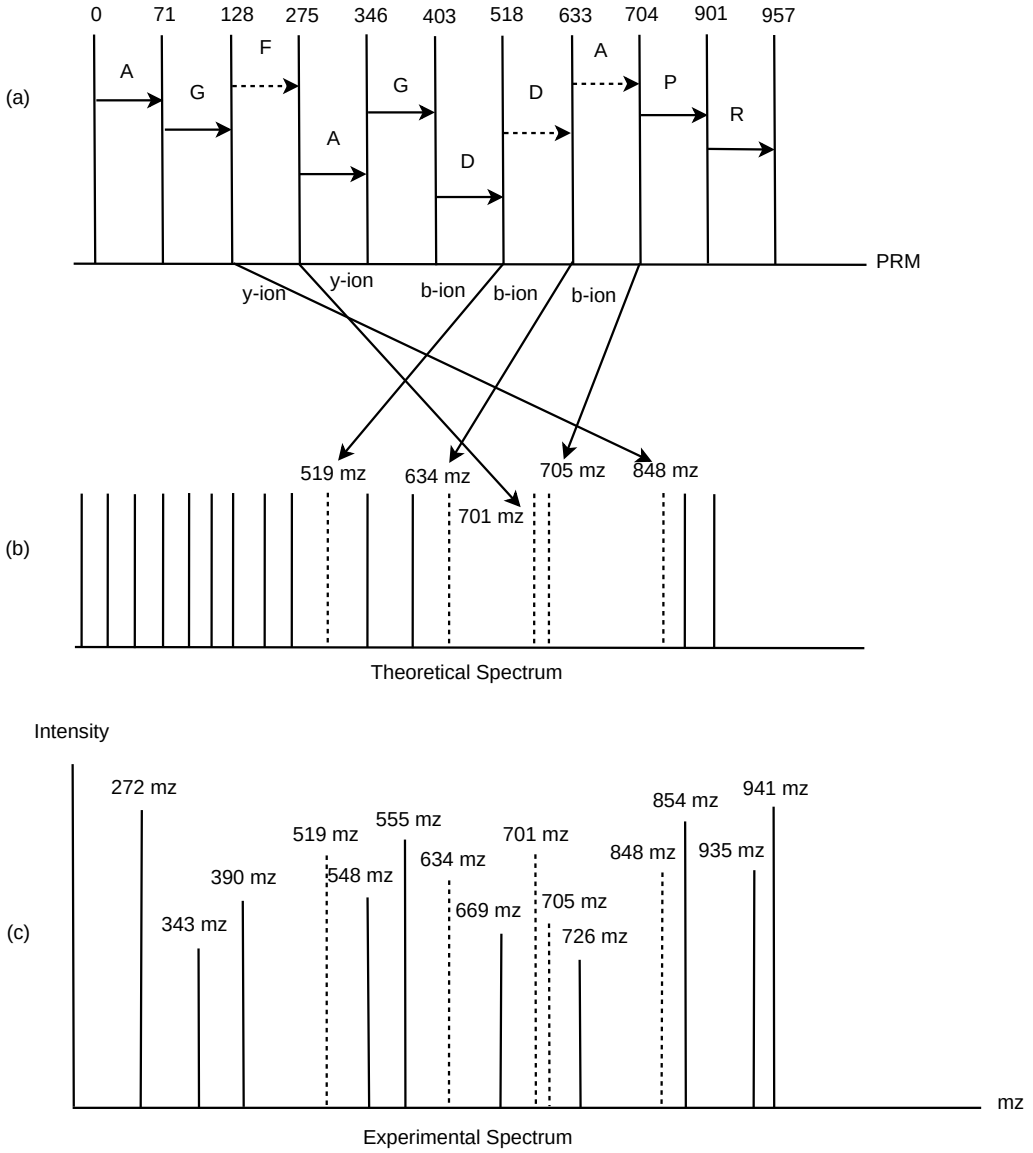


Figure 2.7: The PRM ladder of the peptide shown in (a) generates the theoretical spectrum shown in (b). All of the peaks in the theoretical spectrum (5 of them) matching with peaks in the experimental spectrum show in (c) are indicated by dotted lines. We also show the fragmentation points in the peptide and the ion-types generated which led to these peaks. An example would be the suffix fragment FAGDDAPR (mass = 957-128 = 829Da) which generated a +1 y-ion. This causes a shift of +19 (refer to Table 2.2) and resulted in the peak in both the theoretical and experimental spectrum at 848 m/z.

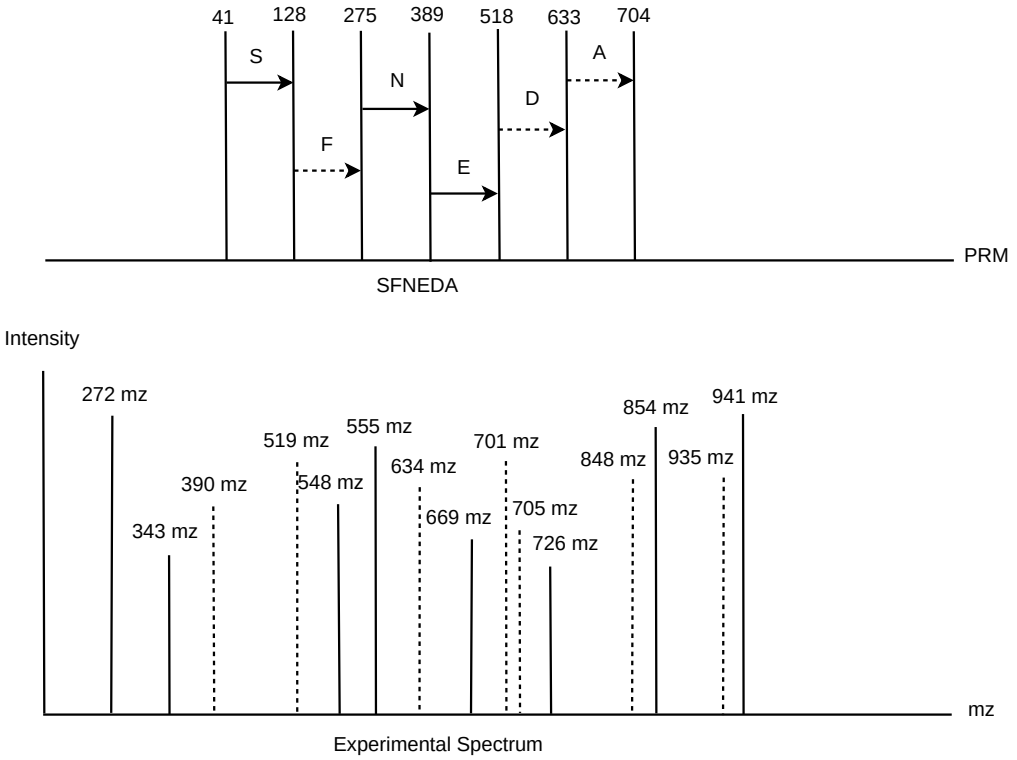


Figure 2.8: PRM ladder for peptide fragment [41]SFNEDA[253] and the matching peaks in the experimental spectrum

However it only matches the subsequence [518]DA[253] in the canonical peptide as indicated, recovering less of the peptide than [41]SFNEDA[253].

In fact currently any objective function used to measure the goodness-of-fit of a candidate peptide with a given experimental spectrum will not necessary mean that a peptide with a better score is the one closer to the canonical peptide. An on-going research problem is to find better objective functions which can give proportionally better match between candidate peptide and canonical peptide with increasing scores. An example of an improved objective function is the weighted SPC where different weights are given to different peaks based on the ion-type of the peak (given in the theoretical spectrum) and the intensity of the peak (given in the experimental spectrum S). More sophisticated functions will be discussed in section 2.4 which is a literature survey of current peptide sequencing algorithms.

In general we call the function that measures the goodness-of-fit of a candidate peptide with the given experimental spectrum the *PSM* (peptide-spectrum match) function.

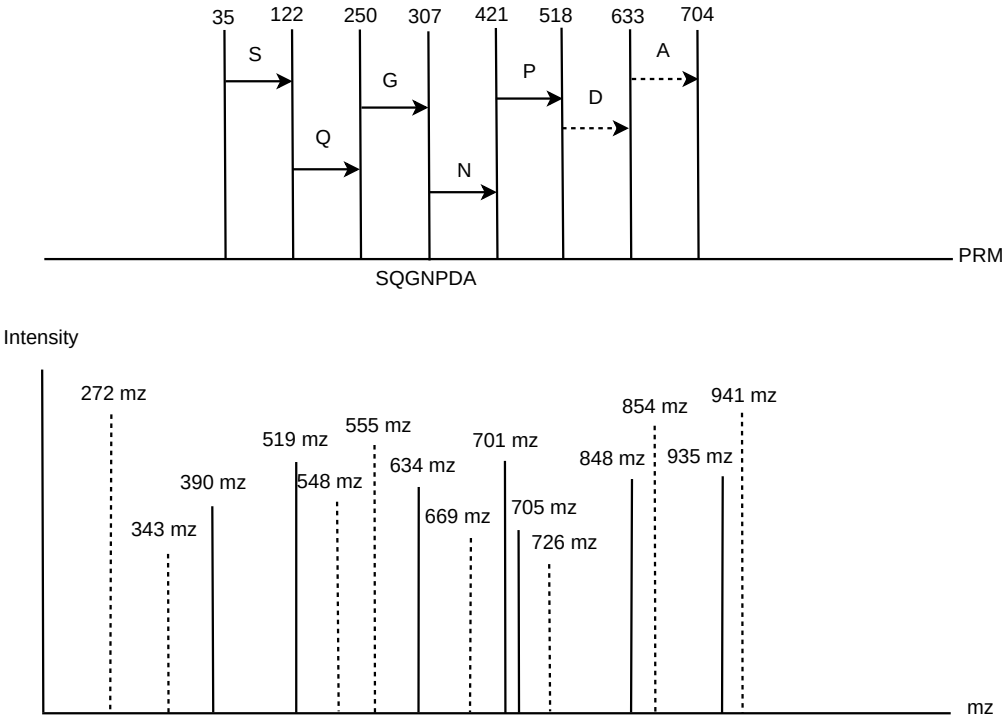


Figure 2.9: PRM ladder for peptide fragment [35]SQGNPDA[257] and the matching peaks in the experimental spectrum

2.3 Major Approaches to De Novo Sequencing

2.3.1 Exhaustive Search

Early approaches to peptide sequencing adopted an exhaustive search framework which was pioneered by Sakurai et al. [56]. This approach involves generating all peptides with peptide mass $M = M'$, the precursor ion mass, along with their theoretical spectrum. The goal is then to find the peptide with a theoretical spectrum that best matches the experimental spectrum S given some objective/scoring function. However this approach quickly becomes intractable since the length of a peptide is proportional to its mass and the number of peptide sequences grows exponentially with the length of the peptide. Hamm et al. [28], Johnson and Biemann [32], and others like Zidarov et al. [77] and Yates et al. [70] attempted to alleviate this problem using prefix pruning which restricts the solution space to sequences whose prefixes match spectrum S well. A drawback with this approach is that the spectrum information is used only after the candidate peptides are generated. Moreover, the correct peptide sequence is often discarded if its prefixes are not as well matched to spectrum S as compared to the prefixes of other candidate

peptides.

2.3.2 Spectrum Graph

The spectrum graph approach is a way to avoid generating all possible peptides by restricting the solution space of candidate peptides to those that can be generated from S itself, and an advantage is that spectrum information is used before any candidate peptide is generated. In this approach, the experimental spectrum S is mapped to a DAG (directed acyclic graph) representation called the *spectrum graph*. Every peak in S generates several vertices in the graph based on the ion-type interpretation it is given. Vertices are linked by a directed edge if they differ by the mass of some amino acid, with the edge pointing from the vertex with the smaller mass vertex with the larger mass. This approach has been used in many algorithm (Bartels [3],de Cossío et al. [14],Taylor and Johnson [65],Dancik et al. [13],Taylor and Johnson [66],Frank and Pevzner [21],Chong et al. [9],Ning et al. [48]). Some of these will be discussed in the following sections. A path in the DAG then represents a candidate peptide, and the peptide sequencing problem becomes a problem of finding the longest or best scoring path in the DAG. Efficient algorithms have been developed to find the longest or optimal path in a DAG (Cormen et al. [10]), and also variants to find the set of top k paths Pollack [55], Yen [71]. Thus in additional to restricting the solution space, searching for one solution is also computationally efficient using the spectrum graph approach.

The main difference between algorithms using this approach is in the way the nodes and edges are weighted. In almost all such algorithms, a path representing a peptide is scored by summing the weights of the nodes and edges along the path. We call this the **simple scoring of a path** and is defined as follows,

Simple Scoring of a Path. Given a path $P = (v_0 e_1 v_1 e_2 v_2 ... e_k v_k)$ of length k, and weights on the edges and nodes in P, we define the Simple Score, $SScore(P)$ of P as follows:

$$SScore(P) = \sum_{j=1}^{k} w(e_j) + \sum_{j=0}^{k} w(v_j) \tag{2.3}$$

There are also variations. For example, in Pepnovo [21], the score of a node also depends on

flanking edges (amino acids) in the path P (will be discussed in more detail in Section 2.4).

2.3.2.1 Spectrum Graph Definition

Spectrum Graph $G(S)$ Given an ion-type set Δ and an experimental spectrum S, the spectrum graph $G_\Delta(S)$ or short-formed as $G(S)$, is defined as follows.

Let each ion-type $\delta \in \Delta$ be arbitrarily numbered from 1 to $|\Delta|$. Each peak p_i in the experimental spectrum S given an ion-type interpretation δ_j is mapped to a vertex $v_{ij}$where the PRM of the vertex $mass(v_{ij}) = \{ \begin{matrix} Shift(p_i, \delta_j) & if \quad \delta_j \quad is \quad an\, n-terminal \quad ion \\ M' - Shift(p_i, \delta_j) & if \quad \delta_j \quad is \quad a\, c-terminal \quad ion \end{matrix}$. The start node v_0 representing mass = 0 and end node $v_{M'}$ representing the precursor parent mass are special nodes added to $G(S)$.

A directed edge $e(u, v)$ is generated when $mass(v) - mass(u)$ is the mass of some amino acid within a given tolerance. A spectrum S of a peptide ρ is *complete* if S contains at least one ion-type corresponding to a prefix fragment ρ_k for every $1 \leq k \leq n$. That is we can find at least 1 path from v_0 to $v_{M'}$ in $G(S)$ which corresponds to ρ.

However in most cases, fragmentation of the peptide in the mass spectrometer is not complete and thus we might not be able to find a path from v_0 to $v_{M'}$. In view of this, many spectrum graph algorithms allow for *mass* edges between vertices, where the mass difference does not correspond to any of the amino acids. This is especially used to link v_0 to the other vertices, and the other vertices to v_m, since CID fragmentation in ESI based mass spectrometers usually have much fewer fragmentation near the right and left end of the peptide (especially low-energy CID), resulting in low peak support given for sequenced amino acids near the two ends of the peptide. The resulting candidate peptides can possibly be flanked on the left and right or even anywhere in the middle by some unexplained masses. These peptides are known as *peptide tags* and *gapped peptides*.

Peptide Tags. A contiguous section of the peptide, with a left mass and a right mass representing unexplained masses. An example is [169]GDAP[356]. A whole peptide is simply a peptide tag with 0 mass at the two ends.

Gapped Peptides. A generalization of peptide tags, where the unexplained masses can be

anywhere in the sequenced peptide. Example [169]G[186]P[356].

These representation allow for cases where we only want highly confident subsequences to be reflected in the sequenced peptides or when there is no way to explain certain fragment masses.

Merged Spectrum Graph $G_{\mathbf{m}}(S)$ Many of the mass spectrometry machines are not very high precision, having resolutions of about 50-200ppm which translates to a precision of about 0.1-0.5 Da on an average sized ion (1000 Da). In order to deal with the precision issue, the masses represented by the nodes are usually rounded off to the nearest integer or mapped to a limited number of possible masses. Because of this there can be many nodes having the same mass. These nodes are then collapsed or merged into one node. The resulting graph is called the merged spectrum graph $G_m(S)$.

Figure 2.10 shows an example 2 paths in the merged spectrum graph generated for the experimental spectrum in Figure 2.6. The two paths represent the candidate peptide tags [41]SFNEDA[253] and [35]SQGNPDA[253] mentioned in Section 2.2.

2.3.3 Tag-Based Approaches

2.3.4 Others

Other approaches includes DP algorithms which do not rely on building a spectrum graph, and machine learning approaches like HMM (Hidden Markov Models) algorithms. These will be discussed in more details in the literature review (Section 2.4).

2.4 Literature Review

We give an overview of some of the state-of-the-art algorithms used in de novo sequencing. We split them into algorithm which are based on the spectrum graph (Section 2.4.1) and those which are not (Section 2.4.2). Aside from the de novo sequencing algorithms to be discussed, there are others like PRIME (Yan et al. [69]), AuDens (Grossmann et al. [27]) and many more (Bafna and Edwards [1], Malard et al. [43], Yergey et al. [72] etc) which have been developed and are being used for de novo sequencing. The survey paper by Lu and Chen [40] provides

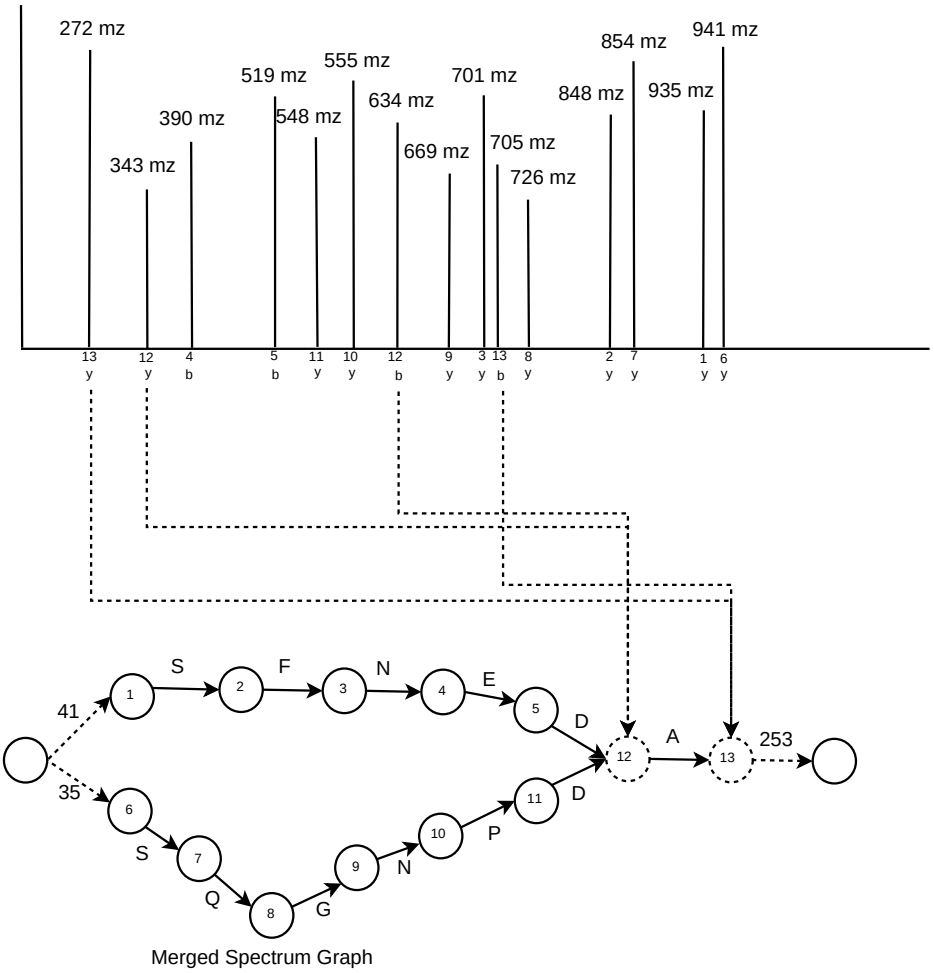


Figure 2.10: Example of two path in a merged spectrum graph $G_m(S)$ for the given experimental spectrum. The numbers beneath each peak represent the node that the peak is mapped to given the ion-type interpretation beneath it. Nodes 12 and 13 are merged nodes formed from 2 peaks, each with a different ion-type interpretation mapping to the same prefix residue mass. The two paths represent the tags [41]SFNEDA[253] and [35]SQGNPDA[253] respectively.

a good starting point to the peptide sequencing problem and gives an overview of some of the algorithms.

2.4.1 Spectrum Graph Algorithms

2.4.1.1 Sherenga

Sherenga developed by Dancik et al. [13] is one of the first commercial de novo sequencing programs using the spectrum graph method, and also one of the first to use a probabilistic scoring function based on the probability of an observed peak being noise or a real peak generated by some ion-type.

Scoring Function Sherenga evaluates a path in the spectrum graph as follows. Each ion-type δ has a certain probability $p(\delta)$ of occurring. This is calculated by finding the frequency of peaks in the training set matching annotated peaks in the theoretical spectrum of the peptide. Since the spectrum graph they use is the merged spectrum graph, each node contains multiple peaks each intepreted by some ion-type in order to map to the node. The scoring function makes use of the probability of these ion-types to score each node in a "**premium for present ions, penalty for missing ions**" scoring. In this scoring, missing ion-types (corresponding to those missing peaks) are "**penalized**", while present ion-types (corresponding to those present peaks) are given a "**premium**" in the following way

$$score\_ions(v_i) = \sum_{\forall \delta \epsilon present\ ions} p(\delta) * \sum_{\forall \delta' \epsilon absent\ ions} 1 - p(\delta') \tag{2.4}$$

this scores all the ion-types considered in the ion-type set $\triangle$ regardless if they are absent or present, since peaks corresponding to abundant ion-types should be observed and thus not being present will *penalize* them negatively as $1 - p(\delta) < p(\delta)$, while peaks corresponding to ion-types which are rare being absent will be *penalize* positively as $1 - p(\delta) > p(\delta)$, and vice versa. This scoring gives a fairer weightage to a node than simply considering present peaks.

The next part of the scoring considers the possibility that the observed peaks are noise peaks. Noise peaks have a fixed probability q of occurring (empirically computed). In the same

way as before, both absent and present noise peaks are considered. The absent noise peaks refers to those ion-type which are not observed possibly being noise. The scoring of noise is as follows

$$score\_noise(v_i) = \sum_{\forall \delta \epsilon present\, ions} p(q) * \sum_{\forall \delta' \epsilon absent\, ions} 1 - p(q) \tag{2.5}$$

Finally the scoring function gives a score for v_i as follows

$$score(v_i) = \frac{score\_ions(v_i)}{score\_noise(v_i)} \tag{2.6}$$

this scoring gives a value greater than 1 when $score\_ions(v_i) > score\_noise(v_i)$, less than 1 when $score\_ions(v_i) < score\_noise(v_i)$ and equals to 1 when both has the same value. The scoring basically compares the hypothesis that the peaks corresponding to the fragmentation point are true peaks against the hypothesis that the peaks are noise peaks which resulted in a spurious fragmentation point. In fact this scoring function is a form of **hypothesis testing** which are used in most of the recent algorithms like PepNovo [21], PEAKS [38], NovoHMM [18] and others like [30]and [16]. The score of a full path is then the summation of all the nodes scores along the path.

Ion-Type Learning In developing Sherenga, Dancik et al. also developed an semi-automated process for learning the ion-type set to be used given a training spectrum set. This process is still widely used to learn specific ion-type sets for different spectrum sets generated by different mass spectrometry machines.

The most important part this ion-type learning process is the offset frequency function $H(\delta, S)$. This function computes for ion-type δ, the frequency that all peaks generated from δ in the theoretical spectrum $TS_\delta(\rho)$ are matched with peaks in the experimental spectrum S, for all peptides ρ in a training set. Mathematically the offset frequency function for an ion-type δ is defined as

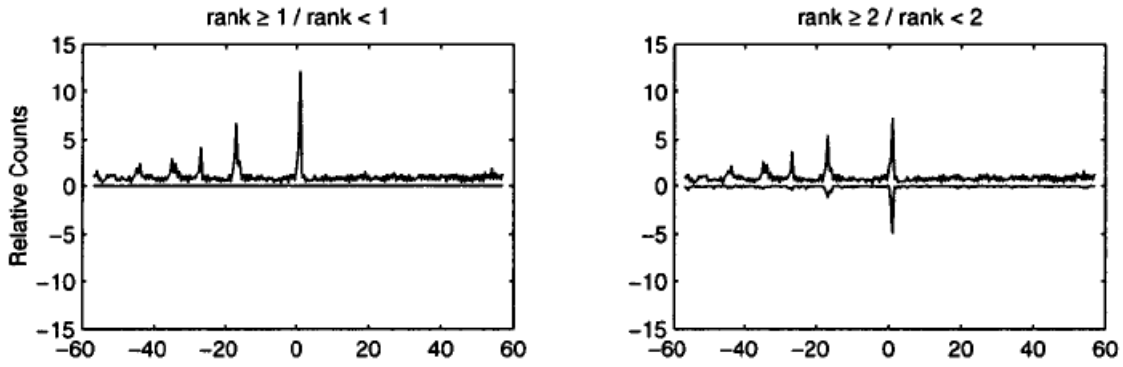


Figure 2.11: Example of offset frequency for intensity rank cutoff = 1 and 2. The graph shows the total frequency of different ion-type (represented by their offset or shift from the partial peptide mass on the x-axis). The top part shows the total frequency for rank $\geq$ the cutoff and the bottom part shows the total frequency for rank $\geq$cutoff. For rank 1 since there are no peaks better than intensity rank=1, there are no peaks on the bottom part, and the top parts represent the total frequency count for the ion-type. For rank 2 we see that the (+1,b-ion) (offset = +1) has the largest peak both in the top part and bottom part followed by (+1,y-ion) (offset = -19). There is a obvious ranking of the ion-types and this can be used to guide the selection of the ion-type set to use.

$$H(\delta, S) = \sum_{\forall \rho} TS_{\delta}(\rho) \cap S_{\rho} \tag{2.7}$$

Due to overmatching if peaks of all intensity are counted, the offset frequency function is often computed for different level of intensity ranking. That is, the peaks are binned into the top K peaks and called rank 1 then the next K peaks are binned and called rank 2 and so on.

The total frequency of an ion-type above a certain intensity rank compared with its total frequency below or equals that rank will give a good idea of how to select the ion-types. This is because high occurrence ion-types will have high frequency for high intensity rankings and low frequency for low intensity rankings and vice versa. Choosing an appropriate intensity ranking cut-off allows us to determine the ion-types to use. Figure 2.11 shows an example of the offset frequency for intensity rank cutoff = 1 and 2 as given in [13].

2.4.1.2 Lutefisk

Lutefisk is another de novo program developed by Taylor and Johnson [66]. The approach adopted by Lutefisk is to first build a spectrum graph in the same manner as [3], then quickly

generate tens of thousands of sequences without scoring them. Different error tolerances are used for different mass spectrometers and a relatively small ion-type set is used, which is also dependent on the mass spectrometer used. Mass edges corresponding to the mass of up to 2 amino acids are allowed between nodes, but only 2 of such edges are allows in the generated sequence. Mass edges up to the mass of 3 amino acids are also allowed at the terminals of the sequence, since fragmentation does not occur frequently near the terminals of the peptide. Nodes in the spectrum graph are scored by the relative probability of the ion-types of the peaks represented by the node. This scoring can be changed based on the spectrometer which the mass spectrum comes from.

The generated candidate sequences are then filtered. The main filtering criteria is to remove all sequences derived from alternating b- and y-ions (mixed paths - described in Section 5.1). This removes 90% of all the generated sequences. The remaining sequences are then scored by summing their node probabilities and ranked.

One observation in [66] is that the m/z error of ions of different types from the correct m/z value is linear with respect to the m/z value. With larger error occurring at higher m/z values. This is in line with the accuracy of the instruments. Using this observation, a novel enhancement was made to Lutefisk. At the start of the sequencing, many candidates are generated using a loose error tolerance ($\pm$0.25u for Qtof machines). A linear mass correction is determined using least-squares method applied to difference between the y- and b-ions of the same fragmentation point and their observed m/z value. This mass correction is applied to the sequences and a tighter mass tolerance is then used to filter away sequences. The algorithm then proceeds as described previously.

2.4.1.3 PepNovo

PepNovo is a recent de novo peptide sequencing algorithm developed by Frank and Pevzner [21] for CID based mass spectrometers. This algorithm uses the hypothesis testing approach. First the probability of observing a set of peak intensities $\bar{I}$ in the spectrum S given that their ion-type interpretation maps them to a fragmentation point with mass m is computed using

a probabilistic network that models the fragmentation rules in the mass spectrometer. This is defined as $P_{CID} = (\bar{I}|m, S)$.

The competing hypothesis is the random peak hypothesis which assumes that the peaks in the spectrum are caused by random process. This is defined as $P_{RAND} = (\bar{I}|m, S)$

Finally the score of a fragmentation point with mass m is the log-likelihood ratio of these two hypothesis.

$$Score(m, S) = log \frac{P_{CID} = (\bar{I}|m, S)}{P_{RAND} = (\bar{I}|m, S)} \tag{2.8}$$

Probabilistic network. The first novel approach in PepNovo is the probabilistic network which is used to compute the probability of a set of peaks with intensities I generated from a mass m by modeling the fragmentation process in the mass spectrometer machine (CID process in this case). Instead of the simple probabilistic scoring of earlier algorithms which assumes that the ion-types are independent from each other, the network models 3 types of dependencies and causal relationships.

The first type of dependency is the correlation between the peak intensities of the different ion-types. The more correlated ion-types (whether positively or negatively) are represented by edges going from one to the other in the network (direction of edge is arbitrary). An example of positively correlated ion-types would be the b-ion and y-ion. The second type of dependency is the ion intensities and the region of the peptide where the ion was generated. Ions generated from the middle of the peptide usually have a higher intensity then ions generated from regions to the left and right terminal ends. This is because fragmentation mostly occurs in the middle while the ends are rarely fragmented. The third type of dependency is the influence of the flanking amino acids (amino acids to the left and right of the fragmentation point) to the ion-types produced and their peak intensities. For example proline, glycine and serine has an N-terminal bias in their fragmentation, thus leading to higher intensity b-ions rather than y-ions (Kapp et al. 33).

In order to simplify the network model, not all possible sets of dependencies are taken into account, since the network will be too large if that is the case, and there will not be enough

training cases to train the network. Also the peak intensities are discretized to 4 levels, and the mass regions are discretized into 5 regions. By doing so, the final trained network can be used as a lookup table in order to get the probability of $P_{CID} = (I|m, S)$.

Random peak hypothesis. The second novel approach in PepNovo is the random peak hypothesis used. This hypothesis does not assume a constant probability for noise as in earlier algorithms like Sherenga (Dancik et al. [13]). Instead, it computes the probability of observing a random peak in a bin of width 2ϵ centered around the fragmentation point of mass m, by using an empirical estimate of the peak intensity distribution in the vicinity of m using the uniform distribution. This vicinity is represented by a window of width w around m. Thus given that there are n_1, n_2, n_3, n_4 peaks of intensity level 1, 2, 3 and 4 (only 4 intensity levels are used in PepNovo as explained) respectively in the window, the probability that a peak of intensity t being the *highest peak* to be randomly placed in the bin around m has the probability

$$P_{RAND}(I = t|n_1, n_2, n_3, n_4) = (1 - \alpha^{n_t}) \cdot \alpha^{\sum_{i=t+1}^{4} n_i} \tag{2.9}$$

where $\alpha = 1 - (\frac{2\epsilon}{w})$ is the probability of uniformly selecting a random location for a peak in a window of width w and having it fall outside a specified bin of width 2ϵ. $(1 - \alpha^{n_t})$ then defines the probability that at least 1 peak of intensity t by random chance falls into the bin. $\alpha^{\sum_{i=t+1}^{4} n_i}$ defines the probability that all peaks of intensity larger than t misses the bin. The multiplication of the two terms is the desired probability.

The probability that no peak falls into a bin is given by

$$P_{RAND}(I = 0|n_1, n_2, n_3, n_4) = \alpha^{\sum_{i=1}^{4} n_i} \tag{2.10}$$

Equation 2.9 and 2.10 then defines a probability density function where

$$\sum_{i=0}^{4} P_{RAND}(I = i|n_1, n_2, n_3, n_4) = 1 \tag{2.11}$$

Assuming that randomly generated peaks are independent of each other, the probability of randomly observing a combination of the k peaks intensities in the bin around m is

$$P_{RAND}(\bar{I}|m,S) = \prod_{i=1}^{k} P_{RAND}(I_i|n_{i1}, n_{i2}, n_{i3}, n_{i4}) \tag{2.12}$$

This random model ensures that in regions where the peaks have high intensities (indicating a lot of noisy peaks), finding a peak with a high intensity in the bin around m results in a relatively high P_{RAND} score, and thus $Score(m,S)$ is lower compared to a situation where we find a high intensity peak in a vicinity with very few peaks or where most peaks are low intensity. In the same way, detecting a low intensity peak in a region with very few peaks can in fact indicate that that peak was not by chance (low P_{RAND}) and thus can positively impact $Score(m,S)$.

2.4.1.4 GBST and GST-SPC

GBST and GST-SPC are a suite of de novo sequencing algorithm developed by Ning et al. [48], based on a preliminary version in [8]. GBST and GST-SPC are based on the idea of "*strong tags*" in building the spectrum graph. A strong tag T is defined here as a maximal path $T = (v_1, v_2, ...v_r)$ in the spectrum graph formed exclusively from either the b-ion or y-ion (frequently observed ion-types). The tags are called strong, as they consists entirely of frequently observed ion-types.

GBST. In the GBST (*Greedy Best Strong Tag*) algorithm, the extended spectrum graph $G_1(S_1^\alpha)$ (this is defined in Chapter 3) is built, and nodes are linked by an edge only if they differ by the mass of some amino acid. This graph may then consist of disjoint components since usually not all possible fragmentations occur and some nodes cannot be bridged. Next each component of $G_1(S_1^\alpha)$ is traversed to obtain all tags of type $(1, b, \phi)$ and $(1, y, \phi)$ of length at least 2. This set of tags are then each scored by summing up the weights of the nodes in the tag. The weight of a node is defined by the following function

$$w(v) = \frac{f_{sup}(v) + f_{loss}(v) + f_{int}(v)}{f_{tol}(v)} \tag{2.13}$$

where

- $f_{sup}(v)$ is a function of the number of v' where v' is a node where $|PRM(v)-PRM(v')| < \epsilon$ the mass tolerance, meaning that v and v' represent the same fragmentation point but are of a different basic ion-type z. We consider such v' a supporting ion of v.

- $f_{loss}(v)$ is a function of the number of v' where v' is similarly another node representing the same fragmentation point as v, but where the basic ion-type z is the same, and differs from v by the neutral losses (water, ammonia, 2*water, water+ammonia). We consider such v' as the neutral loss ions of v.

- $f_{int}(v)$ is a function of the log of the intensity of the ion represented by v.

- $f_{tolerance}(v) = \frac{1}{N}(\sum_{\forall(v',v)\in inedge,(v,v')\in outedge} |PRM(v') - PRM(v) - mass(a_k)|)$, where N is total number of inedges and outedges of v, and a_k is the amino acid represented on each of these edges.

The function basically takes into account all supporting ions and ions representing neutral losses of v which can be observed from the spectrum. It also takes into account the discrepancy between the actual mass difference of the edges coming into and going out of v, and the amino acids represented by them. Larger discrepancy will result in a greater reduction of the score. The scored tags are then ranked and the best tags are retained. The best tags of each ion-type from each component is retained and called the **BST** (best strong tag) set.

After finding the BST, the algorithm proceeds to find the best candidate peptide by linking the tags in a tail (last vertex in tag) to head (first vertex in tag) fashion, where head vertex v of tag t is linked to the tail vertex u of tag t' if they differ by the mass of d amino acid ($d = 2$ is used in the algorithm). The resulting graph is called the strong tag graph $G(BST)$. This graph which will be smaller than $G(S)$ due to the filtering away of tags is then used to find the optimal peptide using the same score function 2.13 and a DFS algorithm.

GST-SPC. This is an improvement to GBST, where firstly, higher charges of the b-ion and y-ion type is considered when computing the strong tags, instead of just the charge 1 versions. Moreover, instead of 1 best strong tag per ion-type per component, the set of *multi-charge* best strong tag are retained. That is the best strong tag corresponding to each of the different charge

version of the ion-type is retained. The score function used to compute the set of multi-charge best strong tags is the same as for GBST. The set of best strong tags are then linked in the same manner as before resulting in the graph $G(GST - SPC)$. Finally instead of computing the best peptide using the function 2.13, we use the SPC(share peaks count) , as it is a more objective criterion for determining the quality of de novo peptide sequencing.

2.4.2 Other Algorithms

2.4.2.1 PEAKS

PEAKS is another commercial de novo sequencing software developed by Ma et al. [41]. This method does not use the spectrum graph method since low-energy CID usually does not allow for full fragmentation of the peptide, introducing significant mass gaps which cannot be bridged in the spectrum graph or might not be practical to bridge (masses equivalent to 3 or more amino acids) without introducing too many edges and thus too many spurious paths into the spectrum graph.

Instead the main idea behind behind PEAKS is a DP algorithm Ma et al. [42] (a preliminary version of the algorithm was used in Ma et al. [41]) which generate candidate peptides based on the following scoring.

$$H(M') = \sum_{(x,h)\in M'} h \tag{2.14}$$

where $M'$is the peak list in the given experimental spectrum S which corresponds to the candidate peptide P'. (x, h) are peaks in M', where x is the m/z ratio of the peak and h is the intensity of the peak. Basically the score of a peptide is the sum of the intensities of all the peaks explaining each fragmentation point in P'. The above is the basic scoring function with more complex variation that have empirically proven to result in better sequences being used.

The novel idea in the DP is the use of "*chummy pairs*", which are pairs of prefix and suffix masses R and Q which overlap by some amino acid $a \in A$ where $R = R'a$ and $Q = aQ'$ and $mass(R'aQ') \leq M'$ the precursor ion mass. The score of chummy pairs $H(M'_{R'aQ'})$ is recursively

computed from those of the smaller chummy pairs present in R' and Q'. The optimal solution is then a series of chummy pairs giving the best score.

Since PEAKS does not make use of the spectrum graph, it takes into consideration all possible masses from 0 to the precursor ion mass subdivided into individual mass units of a suitable size. For all such masses, even when there are no peaks from S which can interpret it, a nominal peak of very small intensity is placed there. This ensure that all possible masses differences that amount to some amino acids can be found by the DP. Chummy pairs then restrict the mass pairs to look at, reducing the size of the solution space.

The full pipeline used by PEAKS to perform peptide sequencing for a spectrum S is as follows

1. Pre-processing - Performs noise filtering and peak centering as well as deconvolution of charge 2 and 3 peaks to charge 1 peaks.

2. Candidate computation - Uses the DP algorithm to compute the top 10000 candidate peptides

3. Re-evaluation of the top 10000 peaks - Uses a more stringent scoring function to score the candidate peptide (described in Section 2.4.2.2)

4. Re-calibration of the data - Recalibration of the mass tolerance is performed in a way that is similar to that described for Lutefisk (Section 2.4.1.2).

5. Compute confidence score for top candidate peptides

2.4.2.2 PEAKS-ETD

Liu et al. [38] developed a log likelihood scoring function for measuring the quality of match between a peptide and a given experimental mass spectrum especially for machines that uses ETD rather CID fragmentation. This scoring function has proven to be better than the original PEAKS scoring function when replacing it in the PEAKS software to sequence ETD data. The scoring function is basically another hypothesis test that compares the hypothesis that a fragmentation point is a real fragmentation point against the hypothesis that it is a random

fragmentation point (random hypothesis). The novelty in this scoring function is in the way that both hypotheses are computed. We will refer to this new scoring function as PEAKS-ETD (it was not given a name in the paper).

Peak Significance Level. The intensity values of the peaks given in the mass spectrum is not a good metric for distinguishing between noise and signal peaks. This is because not all ions form strong peaks, but instead the intensity of the peaks are dependent on the ion-type. Moreover, different regions of the mass spectrum have different noise and signal levels. Thus instead of using the intensity values in computing the probability that an observed ion corresponds to a real fragment, PEAKS-ETD uses the peak *significance level*.

The peak significance level is a measure computed for each peak using the following four features:

1. Global Rank r_g - Numerical ranking of considered peak among all peaks in the mass spectrum according to non-increasing intensity value. A higher ranking means a more significant peak.

2. Local Rank r_l - Numerical ranking of considered peak among all peaks within a $\pm$57Da window centered around the considered peak.

3. Global Intensity Ratio t_g - Ratio between the *global reference intensity* h_g and the intensity h of the considered peak. If the ratio is smaller than 1, it is set to 1.

4. Local Intensity Ratio t_l - Ratio between the minimum of h_g and the intensity h_l of the highest intensity peak in the local window, and the intensity h of the considered peak. If ratio is smaller than 1, it is set to 1. This is defined as $max(1, \frac{min(h_g,h_1)}{h})$.

The global reference intensity h_g was taken to be the average intensity of the 3rd to the 10th highest intensity peaks. The highest intensity peak was not used as the reference since some peptides are hard to fragment, resulting in mass spectrum consisting of only a few high intensity peaks and many low to mid intensity peaks. Hence the high intensity peaks should be considered as outliers and should not be used as a reference.

The four features are combined in a linear equation as follows to compute the significance level of a peak p,

$$siglvl(p) = c_{gr}log(r_{gp}) + c_{lr}log(r_{lp}) + c_{gt}log(t_{gp}) + c_{lt}log(t_{lp}) \tag{2.15}$$

where c_{gr},c_{lr},c_{gt} and c_{lt} are coefficients for the features. By fixing the local rank coefficient c_{lr} to be 1.0, the best combination of values for the other coefficients were found through an exhaustive search, by using the set of values $\{0.01,\ 0.02\ ..\ 1.0\}$ for each of them. For each combination the ROC curve for distinguishing the signal peaks and the background noise in the training data is computed. The combination that gave the best ROC curve was used.

Distribution of peak significance level for different ion-types. Different ion-types have a different peak significance level distribution. PEAKS-ETD computes the distribution of the *log likelihood ratio* of the two hypothesis for each ion-type. The first hypothesis that an ion of the given ion-type with a given significance level is a real fragmentation point, and the random peak hypothesis, that is the ion is a random peak.

Since the significance level is a continuous value, they are partitioned into a set of intervals for each δ. Given that there are n ions of δ in the training set, and m of these ions matches fragmentation points in the canonical peptides, the range of significance levels are divided into 4 intervals I_1, I_2, I_3, I_4 with each containing $\frac{m}{4}$ matched peaks. Thus the probability $Pr_{real}(siglvl(pp) \in I_j)$ of an ion pp having significance level falling into any of the four interval for $j = 1, 2, 3, 4$ is a constant $\frac{m}{4n}$.

The random peak hypothesis $Pr_{random}(siglvl(pp) \in I_j)$ which is the probability that ion pp is a random peak is computed by simple counting. Since the interval ranges are not the same for the 4 intervals, this probability will vary for each interval and thus the likelihood ratio of the two event

$$\frac{Pr_{real}(siglvl(pp) \in I_j)}{Pr_{random}(siglvl(pp) \in I_j)} \tag{2.16}$$

will possibly be different for each interval.

A 5th interval is defined for the largest 10% significance levels. All ions with significance levels falling into this interval are considered as not matching any peaks, since matching will be a very insignificant event.

Denoting the centroid of each interval as c_j, the log likelihood ratio is

$$f(c_j) = log(\frac{Pr_{real}(siglvl(pp) \in I_j)}{Pr_{random}(siglvl(pp) \in I_j)}) \tag{2.17}$$

for $j = 1, 2, 3, 4$ and

$$f(c_5) = log(\frac{Pr_{no\,match}(siglvl(pp) \in I_5)}{Pr_{random}(siglvl(pp) \in I_5)}) \tag{2.18}$$

for $j = 5$. For ion having significance level x, $f(x) = f(c_1)$ for $x < c_1$ and $f(x) = f(c_5)$ for $x < c_5$. $f(x)$ for the other x values are defined by linear interpolation.

For each δ, a set of 5 log likelihood ratio functions f_1 to f_5 are computed by considering the different regions of the peptide.

Scoring of a peptide. Since a fragmentation point P_k of a peptide P can be fragmented into a set of ions of different types $\{pp_1...pp_k\}$, where $siglvl(pp_j)$ is the significance level of the ion if it matches a peak and $siglvl(pp_j) = \infty$ if it doesn't match any peaks in the spectrum. The log likelihood score of a fragmentation point assuming the ion-types are independent of each other is then simply

$$f_{frag}(P_k) = \prod_{i=1}^{k} log(\frac{Pr(siglvl(pp_{ij}))}{Pr_{random}(siglvl(pp))}) \tag{2.19}$$

where Pr can be either Pr_{real} if $siglvl(pp_j) < c_5$ or $Pr_{no\,match}$ otherwise.

The score for the whole peptide P is then

$$Score(P) = \sum_{i=1}^{|P|} f_{frag}(P_i) \tag{2.20}$$

2.4.2.3 NovoHMM

NovoHMM is a Hidden Markov Model based de novo sequencing method developed byFischer et al. [18]. It is a generative method which aims to simulate the fragmentation of the peptide in the mass spectrometer, and thus generate the most likely mass spectrum that will result from a given peptide.

To this end, it adopts the view that the fragmentation process of a peptide in the mass spectrometer is a random process. In order to derive a model for the generation of the mass spectrum using the HMM, 2 simplifying assumptions are made. First, breaks occur only at amino acid boundaries and second, the probability of observing a fragmentation to the left of an amino acid depends only on the amino acid itself. These assumptions allow the modelling of the generative process as a Markov process on a finite state machine. The process of generating the mass spectrum of a given peptide P is then a path through the machine in 1 Dalton steps until the constraint on the mass of the precursor ion M' given in the experimental spectrum is fulfilled.

The finite state machine of the HMM for generation of the most likely mass spectrum of a peptide is given in Figure 2.12. For each amino acid $a \in A$, there is an associated list of $M(a)$ states s_1^a, .. ,$s_{M(a)}^a$, where $M(a)$ is the mono-isotopic mass of a. The machine starts at state s_0 and has 2 end states $s_-$and s_+. The bold edges in the graph corresponds to state transition probabilities $a(s, s')$ from state s to state s'. Once the machine is in state s_1^a of an amino acid a, it must pass through all the states in its associated list, thus the state transitition probability $a(s, s') = 1$ for $s = s_x^a$ and $s = s_y^a$ for $1 \leq x \leq M(a) - 1$ and $x \leq y \leq M(a)$. It mimics the fact that once an amino acid is selected, it must be fully generated, before moving on to another amino acid. Once the machine reaches $s_{M(a)}^a$, it can reach $s_1^{a'}$ of another amino acid a' with state transition probability $p_{a'}$.

The state transition probability from start state s_0 to any s_1^a is just the probability p_a. In order to ensure that the peptide mass constraint is satisfied, the state transition probability changes from $a(s, s')$ to $a'(s, s')$ at step M' (mass of the precursor ion). $a'(s, s')$ gives a probability of 1 of transiting to s_- if s is not some $s_{M(a)}^a$, otherwise it gives a probability of 1 of

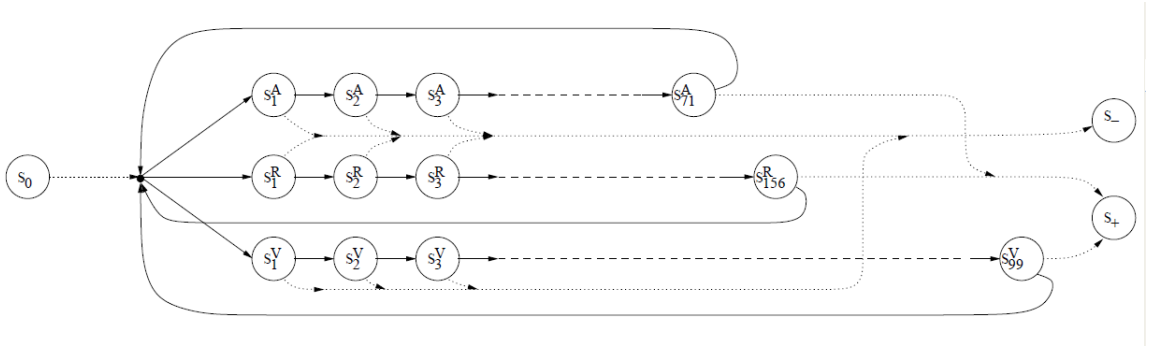


Figure 2.12: Finite State Machine of the HMM for mass spectrum generation. For each amino acid a, there are $M(a)$ states.

transiting to s_+. This ensures that all mass spectra output that ends in s_+ are generated from valid candidate peptides, since their mass is the same as the precursor ion mass given in the experimental spectra.

At each state, an ion count value (a signal generated at a specific m/z value related to the current mass described by the state) is emitted with a certain probability of emission. Since the fragmentation of a peptide can result in either prefix (N-terminal) ion-types or suffix (C-terminal) ion-types, one forward Markov chain and one backward Markov chain is used to simultaneously process peaks generated from either ion-types of either groups. The forward Markov and backward Markov chains are then extended to hidden Markov models to describe the ion counts in the spectra. Since ion counts can be generated at all possible states, random peaks are also generated (ion counts emitted by non $s^a_{M(a)}$ states for any a).

Since the actual canonical peptide mass M can vary from the given precursor ion mass M', the best canonical peptide mass estimate $\hat{M}$ can be computed using a maximum likelihood approach. Using $\hat{M}$, the maximum posterior of the best sequence matching the experimental spectrum is then found using the viterbi algorithm.

2.4.3 Anti-Symmetric Longest Path

One problem associated with de novo sequencing is the fact that it is possible for candidate sequences to use the same peak for different fragmentation points in the sequence. This is due to different ion-type interpretations of the same peak giving rise to different masses that can

ultimately be linked in a candidate sequence. This should not be the case since each of the interpretations still refer to the same fragmentation point in the peptide. Thus all paths which contain multiple instances of the same peak are invalid.

This is the **anti-symmetric longest path** problem associated with de novo sequencing. Here only paths where no peak is assigned to multiple fragmentation points are valid paths, and we are to find the best among these paths. The general problem has been shown to be NP-Complete Cormen et al. [10], and there is no known efficient algorithm to solve it. However, Chen et al. [6] observed some special properties associated with the anti-symmetric longest path problem when applied to peptide sequencing, that makes the problem solvable in polynomial time.

First they observe that a peak p can only appear at most twice in a path in the spectrum graph, and they corresponds to an N-terminal ion and a C-terminal ion. This is because the smallest amino acid has a mass of 57 Da, but the maximum difference between ions from the same terminal is only 45 Da (discounting a$-H_2O - H_2O$ and a$-H_2O - NH_3$ which are very rare ions and usually not included into the ion-type set) and thus a peak given the two different ion-type interpretations which comes from the same terminal can never be found in the same path as they cannot be linked by an edge.

Therefore since a peak can appear at most twice in a path, the vertices are called **forbidden pairs** v_{pn} (vertex given N-terminal ion interpretation) and v_{pc} (vertex given C-terminal ion interpretation) respectively.

The next property of forbidden pairs in a path is that given peaks p and p' and the ion-type interpretations n and c, the forbidden pairs v_{pn},v_{pc} and $v_{p'n}$,$v_{p'c}$ are non-interleaving. This is because $mass(v_{pn}) + mass(v_{pc}) = M'$ the precursor mass, since one is the PRM and the other the SRM of the same fragmentation point. Similarly, $mass(v_{p'n}) + mass(v_{p'c}) = M'$. Thus if $mass(v_{pn}) < mass(v'_{pn})$, then $mass(v'_{pc}) < mass(v_{pc})$ and vice versa.

Assuming an ion-type set of charge 1 b-ion and y-ion, vertices can be generated for each of the ion-type. The b-ion vertices are called N vertices and the y-ion vertices, C vertices. This is called the **NC-spectrum graph**. Vertices are linked as usual, and they are then placed

on a real line at positions corresponding to their masses. The vertices are then renamed in order from left to right (b_0,b_1,..,b_k, y_k, y_{k-1}, ..y_0), where every pair b_i and y_i corresponds to the different ion-type interpretation of the same peak, that is $mass(b_i) + mass(y_i) = M'$.

A DP was developed to find the path with the maximum score from x_0 to y_0 which contains the edge $e(x_i, y_j)$ where $i \neq j$. Lu and Chen [39] further developed the algorithm to include more ion-types and also find all sub-optimal solutions instead of just the optimal solution, since the optimal solution is not always practical.

Many current de novo sequencing algorithms use this algorithm for solving the anti-symmetric longest path problem. Others do not disallow forbidden pairs, since there are situations where two different fragmentation points can possibly coincide at the same peak due to different ion-type generation. PEAKS for example allow for forbidden pairs, but only score one of the interpretations of the peak. In fact a study (Ning and Leong [47]) shows that strictly disallowing peptides with peaks given multiple assignments may affect the sequencing result adversely.

2.4.4 Post-processing candidate peptides

Most sequencing methods score candidate peptides using local information (current fragmentation points or flanking amino acids of current fragmentation points). Paths representing candidate peptides are then a simple sum of such local scoring. This is to be expected since this scoring allows for efficient algorithms such as DP to be used to generate top x candidate peptides.

After sequencing however, the generated candidate peptides can be rescored based on more detailed scoring functions that exploit global information present in the peptides. **PepNovo+** (Frank [20]) is an improvement to the PepNovo software which seeks to improve the ranking of the generated candidate peptides by using a machine learning ranking algorithm called RankBoosting. A discriminative model instead of the usual probabilistic models for scoring peptides against the experimental spectrum is generated using RankBoosting. To train this discriminative model a variety of features that capture the quality of a PSM (Peptide Spectrum Match) between a candidate peptide and the experimental spectrum is used. These include the original

score of the peptide's path in the spectrum graph, the ions generated for each fragmentation point in the peptide, intensity ranks for each ion peak based on how a novel peak ranking algorithm (Frank [19]), features examining the amino acid makeup of the peptide etc. The trained model is then applied to candidate peptides generated from PepNovo to rescore and re-rank them. Improvements of 22% in pepide identification was noted when applying the top peptide tag after post-processing for database search.

Spectral Profiles (Kim et al. [35]) is another post-processing algorithm developed for re-evaluating candidate peptides. The main idea behind spectral profiles is that full peptide sequencing is usually not attainable since 1.) highly similar peptides have highly similar spectra which makes disambiguation difficult without additional information, and 2.) variable local quality along peptide makes some regions not amenable to sequencing. Thus instead of using the full candidate peptide, spectral profiles seek to re-evaluate the peptide and generate a gapped version, where portions of the candidate peptide which is of low quality is replaced by a mass tag. This gapped version is then re-scored and re-ranked. The advantage of the gapped peptide over the full peptide is that it is almost as accurate as short sequence tags and it generates more unique hits when used with database search as opposed to short sequence tags.

The method generates these gapped peptide using the spectral profile for a mass spectra. A spectral profile in this context is a compact representation of all high scoring de novo reconstructions (a *spectral dictionary* [36]) for the spectra even when there are billions of such reconstructions.

More specifically, in this method, a peptide can be represented as a k-mer boolean vector $P = x_1...x_k$ where $k = \frac{M}{min.\ unit\ of\ mass}$. Thus $x_i = 1$ if $\frac{i}{min.\ unit\ of\ mass}$ represents a prefix mass of the peptide and 0 otherwise. A spectral dictionary $D = \{P_1..P_m\}$ is the set of "high" scoring candidate peptides. A spectral profile is then $Profile(D) = \frac{1}{m}\sum_{j=1}^{m} P_j$.

Generating a spectral dictionary can be prohibitive especially for long peptides since there could be billions of such recontructions. The method uses a forward-backward DP solution to generate the spectral profile without having to generate the spectral dictionary. Spectral profiles are usually generated by setting a threshold such that total probabilities of peptides

in the spectral dictionary do not exceed a predefined probability called the *spectral probability*. For de novo sequencing, the top x generated candidates can be also be used as the spectral dictionary.

Now given a candidate peptide $P = x_1..x_k$, $Profile = f_1...f_k$ and a parameter $minprob$, the gapped peptide for P is $GappedPeptide(P, Profile, minprob) = g_1...g_k$ where $g_i = x_i$ if $f_i \geq minprob$, 0 otherwise. The gapped peptide can then be rescored and re-ranked. In experiments done where spectral profile was generated from the top x ranking peptide generated from PepNovo, the top ranking gapped peptides after applying the spectral profile led to a 90% hit rate when used in database search compared to 26% when simply using the candidates tags from PepNovo. The average number of false matches were 1.6 for gapped peptides and 80.3 for tags from PepNovo.

Chapter 3

Generalized Model for Multi-Charge MS/MS Spectra

In this chapter, we discuss peptide sequencing on multi-charge tandem mass (MS/MS) spectra, that is, peptide sequencing on spectrum with charges from +1 to higher than +3. Here we introduce a generalized model of peptide sequencing that accommodates higher charges. We also redefine the concept of supporting ions and extend this to the concept of supporting edges. Our extended spectrum and extended spectrum graph model allows us to first of all include the higher charged spectra (>+3) in our characterization of multi-charge MS/MS spectra in Chapter 4. Our extended spectrum graph model also allows us to also discuss the development of a new peptide sequencing algorithm in Chapter 5.

3.1 Extended Theoretical Spectrum

We define the *extended theoretical spectrum* $TS_\alpha^\alpha(\rho)$ for peptide ρ for precursor charge (or maximum charge) α to be the set of all possible observed peaks that may be present in an experimental spectrum for ρ with maximum charge α. More precisely, $TS_\alpha^\alpha(\rho)$ =$\{p$: p is an observed peak for the (z,t,h)-ion of the peptide prefix fragment ρ_k for all $(z,t,h) \in \triangle$ and $k = 1, ..., n\}$.

Some peptide sequencing algorithms consider only ion-types of charge up to 2 even when the

spectrum is of a higher charge (namely, $\alpha > 2$). To take this into account, we introduce a new parameter β and we extend the definition of extended theoretical spectrum to $TS^{\alpha}_{\beta}(\rho)$ which is defined to be the set of observed peaks with charge $z \in \{1, 2, ..., \beta\}$. More precisely, $TS^{\alpha}_{\beta}(\rho) = \{p : p \in TS^{\alpha}_{\alpha}(\rho)$ and $z \in \{1, 2, ..., \beta\}\}$. Namely, $TS^{\alpha}_{\beta}(\rho)$ only accounts for peaks of ion-types with charge 1, 2, . . ., β and does not account for peaks that correspond to ion-types with charge $\beta + 1$, . . ., α. The case $\beta = 1$ reflects the assumption that all peaks are assumed to be of charge 1, and the peaks in an experimental spectrum are compared against $TS^{\alpha}_{1}(\rho)$. We use $\beta = 2$ for algorithms that consider ion-types of charge up to 2 and thus, the corresponding extended theoretical spectrum used is $TS^{\alpha}_{2}(\rho)$. Clearly, the more charges that an algorithm takes into account, the higher will be the *recovery rate* (how much of the canonical peptide can be recovered from the candidate peptide) that can be attained — since $TS^{\alpha}_{1}(\rho) \subseteq TS^{\alpha}_{2}(\rho) \cdot \cdot \subseteq TS^{\alpha}_{\alpha}(\rho)$. We denote $TS_0(\rho)$ to be the set of all "uncharged" prefix fragment masses of the peptide ρ (the prefix ladder). That is $TS_0(\rho) = \{m(\rho_1), m(\rho_2), ...m(\rho_n)\}$.

3.2 Extended Spectrum

Conversely, in an experimental spectrum $S = \{p_1, p_2, ..., p_n\}$ of maximum charge α, the real peaks may be from different ion-type of different fragments (prefix or suffix fragment, depending on the ion-type). We do not know, *a priori*, the ion-type $(z, t, h) \in \triangle$ of each peak p_i. Therefore, we "extend" each peak $p_i \in S$ into $|\triangle|$ pseudo-peaks (or guesses) — one pseudo-peak for each of the different ion-type $(z, t, h) \in \triangle$. More precisely, the extended spectrum $S^{\alpha}_{\alpha} = \{(p_i, (z, t, h)) : p_i \in S$ and $(z, t, h) \in \triangle\}$, where $(p_i, (z, t, h))$ denotes the pseudo-peak for the peak p_i and ion-type (z, t, h) and has an "assumed" (uncharged) fragment mass computed by the *Shift* function in Equation 2.1. For each p_i, at most one of these pseudo-peaks is a real peak, while the others are noise "introduced" by the extension process. We denote $pp_i(\delta)$ as the pseudo peak (p_i, δ) for some $\delta \in \triangle$.

To account for algorithms that only uses charges 1 and 2, we generalize our definition of extended spectrum to $S^{\alpha}_{\beta} = \{(p_i, (z, t, h))$: $(pi, (z, t, h)) \in S^{\alpha}_{\alpha}$ and $z \in \{1, ..., \beta\}\}$. Many current algorithms uses S^{α}_{2}, even for higher charge spectra — ions of higher charge (≥ 3) are ignored.

We always express a fragment mass in the extended spectrum using its PRM representation, which is the mass of the prefix fragment (PRM). For suffix fragments, we use the mass of its corresponding prefix fragment. Mathematically, for a fragment q with mass $m(q)$, we define $PRM(q) = m(q)$ if q is a prefix fragment, and we define $PRM(q) = M - m(q)$ if q is a suffix fragment. By calculating the PRM for all fragments, we can treat all fragment masses uniformly.

Example 1. We have a peptide $\rho =$ **GAPWN**, with parent mass $M = m(\rho) = 525.2$ and an experimental spectrum $S = 113.6, 412.2, 487.2$ with maximum charge $\alpha = 2$. For simplicity, we only consider ion-types $\Delta_t = \{$b-ions, y-ions$\}$ and $\Delta_h = \{\phi\}$ in this example. The first peak $p_1 =$ "113.6" is a (2, b-ion,ϕ)-ion of the prefix fragment **GAP**; the peak $p_2 =$ "412.2" is a (1, b-ion,ϕ)-ion of the prefix fragment **GAPW**; and $p_3 =$ "487.2" is a (1, y-ion, ϕ)-ion for the fragment **G**.

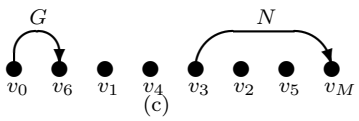
Figure 3.1 illustrate the use of extended spectrum for Example 1. If only charge 1 is considered, then we have the extended spectrum $S_1^2 = \{112.6, 430.6, 411.2, 132, 486.2, 57\}$ (we give the m/z ratios fo the peaks), as shown in Fig. 1(a). In S_1^2 , the peak p_1 extends to two pseudo-peaks $v_1 = (p_1, (1,$ b-ion,$\phi))$ with PRM value of 112.6, and $v_2 = (p_1, (1,$ y-ion, $\phi))$ with PRM value of 430.6. Both pseudo-peaks are not true peaks. The peak p_2 extends to two pseudo-peaks $v_3 = (p_2, (1,$ b-ion,$\phi))$ with PRM value of 411.2, and $v_4 = (p_2,(1,$ y-ion,$\phi))$ with PRM value of 132. The first pseudo-peak is a true peak while the second is a noise peak. However, the true peak p_1 cannot be captured if we use only S_1^2 since it is a charge 2 peak.

z	$mz(p_1) =$ 113.6		$mz(p_2) =$ 412.2		$mz(p_3) =$ 487.2	
1	B	Y	B	Y	B	Y
	v_1 112.6	v_2 430.6	v_3 411.2	v_4 132	v_5 486.2	v_6 57

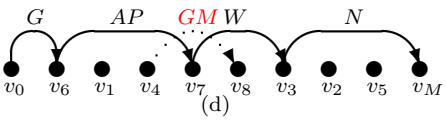
(a) The spectrum S_1^2 (only B and Y ions considered).

z	$mz(p_1) =$ 113.6		$mz(p_2) =$ 412.2		$mz(p_3) =$ 487.2	
2	B	Y	B	Y	B	Y
	v_7 225.2	v_8 318	- -	- -	- -	- -

(b) Extending the peaks for charge 2 ions.



(c)



(d)

Figure 3.1: Example of extended spectrum graph for mass spectrum generated from peptide GAPWN. The parent mass is $M = m(\rho) = 525.2$

However, if charge 2 is also considered, then we have the extended spectrum $S_2^2 = \{112.6,$ 430.6, 411.2, 132, 486.2, 57, 225.2, 318$\}$ as shown in Fig. 3.1(b) and it captures all the true

peaks in S. In S_2^2, the peak p_1 extends to two new pseudo-peaks of charge 2, namely $v_7 = (p_1,$ (2, b-ion,ϕ)) with PRM value of 225.2, and $v_8 = (p_1,$ (2, y-ion, ϕ)) with PRM value of 318. However, some extensions are clearly infeasible, such as the extension (p_2, (2, b-ion, ϕ)) with a putative PRM value of 822.4, which is larger than the parent mass, $M = 525.2$.

3.2.1 Supporting Ions.

In a CID or similar process, different ions from the ion-type set Δ can arise for the same fragmentation point. In this case, we say that they are *supporting* ions of one another. These supporting ions have the same prefix residue mass (PRM), but have different mass/charge ratios and so they appear as different peaks in the experimental spectrum S.

Formally, for any pseudo-peak $pp_1(\delta)$ called the *main peak* or *main ion*, another pseudo-peak $pp_2(\delta')$, $\delta \neq \delta'$ is said to a *supporting peak* or *supporting ion* if they have the same prefix residue mass (subject to an error tolerance ϵ), i.e., $|pp_2(\delta) - pp_1(\delta')| \leq \epsilon$. For any pseudo-peak $pp_1(\delta)$, we let $SI(pp_1(\delta))$ denote the set of supporting pseudo-peaks for $pp_1(\delta)$.

The number of the supporting pseudo-peaks is an indication of the likelihood that the main ions are real peaks corresponding to real fragmentation points of the peptide, and not just a collection of independent random noise peaks. For example, in Figure 3.2, pseudo-peaks $(p_1, y - ion)$ and (p_4,b-ion) are supporting ions of each other since their PRM=635, while pseudo-peaks (p_2,y-ion) and (p_3,b-ion) are supporting ions of each other since their PRM=706. The notion of supporting peaks have been used in scoring of nodes in a spectrum graph in many de novo algorithms such as PEAKS (Liu et al. [38]), PepNovo (Frank and Pevzner [21]), and Sherenga (Dancik et al. [13]).

3.2.2 Duality between extended spectrum and extended theoretical spectrum

We now describe a duality relationship between the extended spectrum S_β^α and the extended theoretical spectrum $TS_\beta^\alpha(\rho)$. Given an experimental spectrum S of a known peptide ρ, the set $RP_\beta^\alpha(S, \rho)$ of real peaks in the spectrum S is given by:

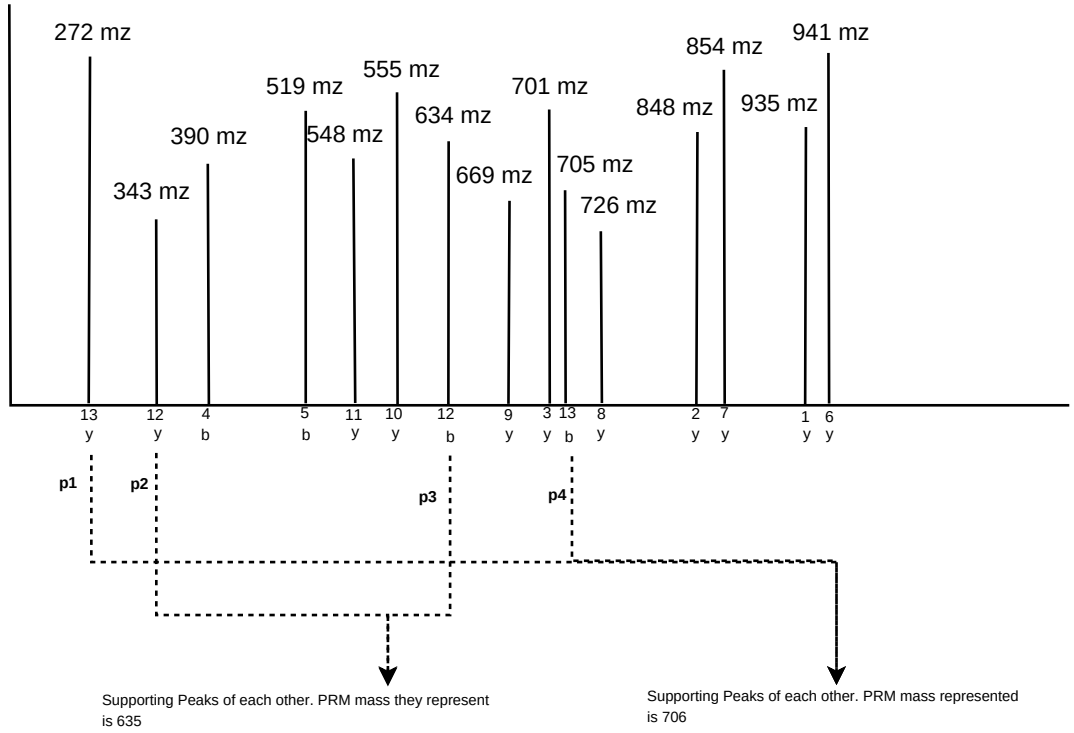


Figure 3.2: Example of Supporting Ions. The pseudo-peaks (p_1,y-ion) and (p_4,b-ion) represents $PRM = 635$ and are supporting ions for each other. On the other hand, the pseudo-peaks (p_2,y-ion) and (p_3,b-ion) represents $PRM = 706$ and are supporting ions for each other.

$$RP_{\beta}^{\alpha}(S,\rho) = TS_{\beta}^{\alpha}(\rho) \cap S \tag{3.1}$$

This relationship is used by many database search algorithms to compare an experimental spectrum S against a putative peptide ρ generated from the protein database. Clearly, considering higher charge increases the set of real peaks obtained, namely, $RP_1^{\alpha}(S,\rho) \subseteq RP_2^{\alpha}(S,\rho) \cdot \cdot \cdot \subseteq RP_{\alpha}^{\alpha}(S,\rho)$. For Example 1, when $\beta = 1$, we have $RP_1^{\alpha}(S$,GAPWN) = {p_2(1, b-ion,$\emptyset$), p_3(1, y-ion,$\emptyset$)}. For $\beta = 2$, we have RP_2^{α} (S,GAPWN) = {p_1(2, b-ion,$\emptyset$), p_2(1, b-ion,$\emptyset$), p_3(1, y-ion,$\emptyset$)}.

The set $EF_{\beta}^{\alpha}(S,\rho)$ of explained fragments in the peptide ρ, namely fragments that are supported ("explained") by peak(s) or pseudo-peak(s) in S_{β}^{α}, is given by:

$$EF_{\beta}^{\alpha}(S,\rho) = TS_0(\rho) \cap PRM(S_{\beta}^{\alpha}) \tag{3.2}$$

where, we recall that $TS_0(\rho)$ is the set of prefix fragment masses for the peptide ρ. This

relationship is implicitly used by many *de novo* sequencing algorithms. Clearly, considering higher charge increases the set of explained fragments, namely, $EF_1^\alpha(S,\rho) \subseteq EF_2^\alpha(S,\rho) \cdot \cdot \cdot \subseteq EF_\alpha^\alpha(S,\rho)$. For Example 1, when $\beta = 1$, we have EF_1^2(*S*,GAPWN) = {GAPW, G}, supported respectively by the pseudo-peaks (p_2,(1,b-ion,Ø)) and (p_3(1,y-ion,Ø)). For $\beta = 2$, we have EF_2^2(*S*,GAPWN) = {GAP, GAPW, G}, supported by (p_1,(2, b-ion,Ø)), (p_2,(1, b-ion,Ø)) and (p_3(1, y-ion,Ø)). Some algorithms such as Lutefisk and PepNovo consider only charges 1 and 2 (namely, $\beta = 2$) and their set of explained fragments is bounded by $EF_2^\alpha(S,\rho)$. Consequently, we expect them to perform less well on higher charge spectra. In the set $RP_\alpha^\alpha(S,\rho)$, there may be several real peaks that are support peaks for the same fragment. Similarly, in the set $EF_\alpha^\alpha(S,\rho)$, there may be multiple pseudo-peaks in S that helps to "explain" the same fragment. This is more formally stated in the following duality theorem.

Theorem 1 (Duality Theorem). For any experimental spectrum S of a known peptide ρ with a maximum charge of α, we have

$$EF_\alpha^\alpha(S,\rho) = PRM(Shift(RP_\alpha^\alpha(S,\rho))) \tag{3.3}$$

3.3 Extended Spectrum Graph

We also introduce an extended spectrum graph, denoted by $G_d(S_\beta^\alpha)$, for the extended spectrum S_β^α, where d is the "connectivity". For simplicity, we first define $G_1(S_\beta^\alpha)$, the extended spectrum graph for S_β^α with connectivity 1. Each vertex $v = (p_i, (z,t,h))$ in this graph represents a pseudo-peak $(p_i, (z,t,h))$ in the extended spectrum S_β^α, namely, the (z,t,h)-ions for the peak p_i. Each vertex represents a possible peptide fragment mass given by $PRM(Shift(p_j\ , (z,t,h)))$. Two special vertices are added — the start vertex v_0 corresponding to the empty fragment with mass 0 and the end vertex v_M corresponding to the parent mass M. There is a directed edge (u,v) from vertex u to vertex v in the graph $G_1(S_\beta^\alpha)$ iff $PRM(v)$ is larger than $PRM(u)$ by the mass of a single amino acid. We note that our graph extends the more "standard" spectrum graph $G(S)$ by taking into account pseudo-peaks with higher charges.

In the extended spectrum graph of connectivity d, $G_d(S_\beta^\alpha)$, we further extend the definition

of an edge to mean "a directed path of up to d amino acids". Thus, we connect vertex u and vertex v by a directed edge (u, v) if the $PRM(v)$ is larger than $PRM(u)$ by the total mass of d' amino acids, where $d' \leq d$. In this case, we say that the edge (u, v) is connected by a path of length up to d amino acids. Note that the number of possible paths to be searched is $20d$ and increases exponentially with d.

For **Example 1**, Figure 3.1(c) shows the extended spectrum graph $G_2(S_1^2)$ for S_1^2 with connectivity 2. We can see that only the edges (v_0, v_6) for amino acid **G** and (v_3, v_M) for amino acid **N** can be obtained. The subsequence **APW** is longer than two amino acids long and so $G_2(S_1^2)$ is unable to elucidate this information. For $\beta = 2$, we use the extended spectrum S_2^2 and the corresponding extended spectrum graph $G_2(S_2^2)$ shown in Figure 3.1(d). New edges can be obtained, edge (v_6, v_7) for path **AP** of length two amino acids and (v_7, v_3) for amino acid **W**. This gives a full path from v_0 to v_M and the full peptide can now be elucidated (as shown in Figure 3.1(d)). This example illustrates the advantage of considering higher charges.

We also note that in $G_2(S_2^2)$, fictitious edges may also be introduced due to the introduction of noise pseudo-peaks. In Figure 3.1(d), the fictitious edge (v_4, v_8) is shown using dashed line. Many such fictitious edges can result in fictitious paths from v_b to v_e, thus giving a higher rate of false positives. Indeed, one challenge with higher charge spectra is that of dealing carefully with false positives.

Example 2. To further illustrate the ability to extract more of the peptide by considering higher charge ion-types, we consider a charge 4 spectrum from the GPM-Amethyst dataset (Craig et al. [12]). The peptide that generated the spectrum is **AGFAGDDAPRAVFPSIVGRPR** and the data file is shown on the left of Figure 3.3. The real peaks are colored, with ion-type, and corresponding prefix fragment indicated on the right. The full peptide ladder is given on the far right and the fragments which are present in the spectrum are also shown in italics (and underlined) and annotated with the peaks which were generated from them. The Figure on the left of Figure 3.4 shows that if we only consider charge 2 (namely, use $\beta = 2$), then only 10 of the 21 different prefix fragments and six of the 21 amino acids can be obtained from the extended spectrum graph $G_1(S_2^4)$. However, if all the charges are considered (using β

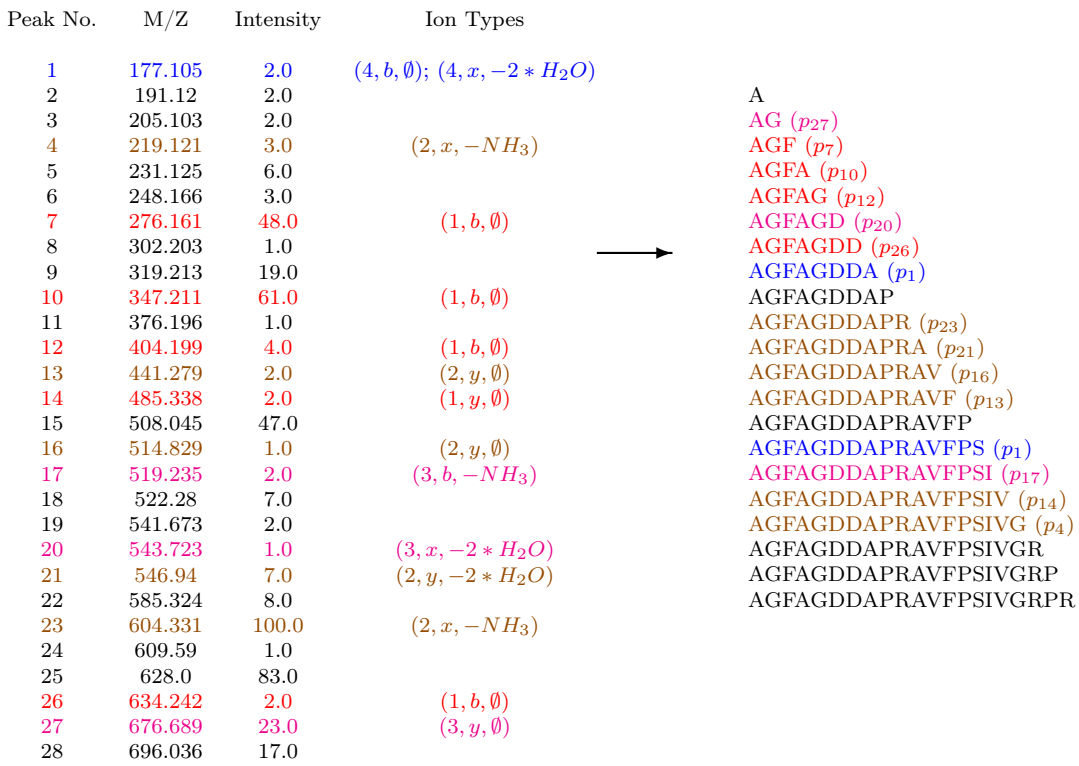


Figure 3.3: A charge 4 spectrum from the GPM-Amethyst dataset. The real peaks are colored, with ion-type, and corresponding prefix fragment indicated on the right.

$= 4$), then 15 of the 21 prefix fragments and 12 of the 21 amino acids can be obtained using $G_1(S_4^4)$.

3.3.1 Supporting Edges

We extend the idea of supporting ions (described in Section 3.2.1) to that of supporting edges in the extended spectrum graph $G_d(S_\beta^\alpha)$. In order to do so, we first associate a *color* to each of the different ion-type in $(z, t, h) \in \triangle$. Then each vertex v in G has a color that represents the ion-type (z, t, h) of the corresponding pseudo-peak.

An edge (u, v) is called a *mono-chromatic edge* if both vertices u and v have the same color (i.e., of the same ion-type). If u and v have different colors (different ion-types), then the edge (u, v) is called a *mixed edge*.

For any mono-chromatic edge (u, v), another mono-chromatic edge (u', v') is a supporting edge of (u, v) if

1. $|PRM(u) - PRM(u')| < \epsilon'$

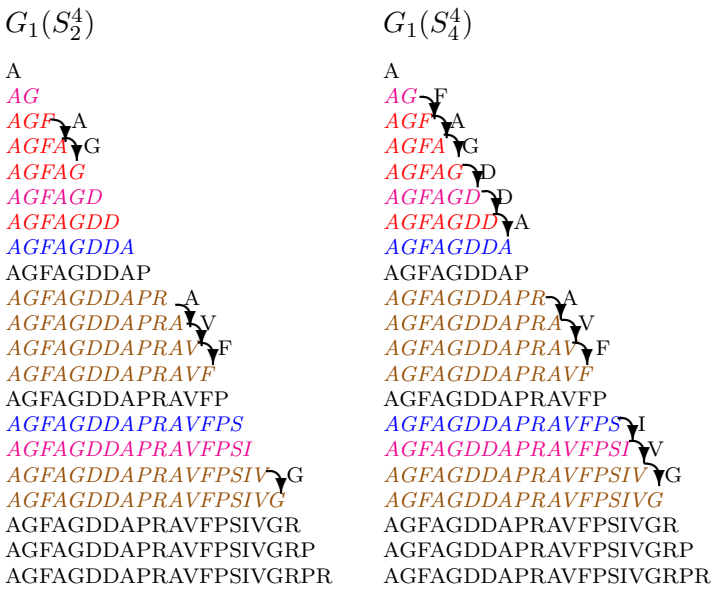


Figure 3.4: Progression in amount of peptide that can be elucidated, if higher charges were to be considered. The colored fragments are those obtainable. Here only single amino acid differences are considered. On the left, when considering up to charge 2 for the charge 4 spectra, we can obtain only 6 out of 21 of the amino acids. On the right, by considering up to charge 4, we can obtain 12 of the amino acids.

2. $|PRM(v) - PRM(v')| < \epsilon'$

3. $|(PRM(v) - PRM(u)) - (PRM(v') - PRM(u'))| < \epsilon'$

Condition 1 and 2 ensures that supporting edges are not too far off from each other, and condition 3 ensures that they represent the same mass difference or amino acid. Thus for any mono-chromatic edge $e = (u, v)$, let $SE(e)$ be its set of supporting edges.

The idea behind supporting edges is that instead of supporting ions for one fragmentation point, we find the supporting ions for a consecutive pair of fragmentation points (represented by an edge in the graph). In order to retain the concept of main ion and supporting ions in this extension, we can apply it to pairs of fragmentation points with the same main-ion, that is mono-chromatic edges, and the supporting edges will also have to be mono-chromatic. Thus a mixed edge e' will never have any supporting edges, and $SE(e') = \phi$.

Just as the number of supporting ions are an indication of the likelihood that the main ions are real peaks, the number of supporting edges are an indication that the main edge represents a pair of real fragmentation points.

3.3.2 Advantage of Extended Spectrum Graph over Merged Spectrum Graph

Extended Spectrum Graph has no merged nodes. An aspect of the extended spectrum graph $G_d(S^{\alpha}_{\beta})$ is that we do not merge the nodes that are close in mass as opposed to merged spectrum graph (described in Section 2.3.2.1) algorithms like Sherenga (Dancik et al. [13]) and PepNovo (Frank and Pevzner [21]). Even though node merging will help to reduce the size of the graph and thus the solution space, it has 2 disadvantages.

1. **Merged nodes causes "gaps" in the merged spectrum graph**. The center of mass of merged nodes is usually the average mass of the nodes collapsed to form them. However, regardless of how the mass is calculated, the full spread of masses as represented by the peaks in the experimental spectrum S is not encoded in the graph when the nodes are merged. This can cause problem by introducing "gaps" in the spectrum graph. Two unmerged nodes can be bridged by some amino acid, but after merging, the difference between the center of masses of the merged nodes exceeds the error tolerance and this causes a gap to appear. This is illustrated in Figure 3.5, where the amino acid A (mass = 71) can no longer be bridged in the merged graph if an error tolerance of $\epsilon = 0.5Da$ is used. This is because the merged node has an average mass of 385.5 resulting in a mass difference of 71.7 (error tolerance exceeded). This artifact of merging causes a drop in the amount of peptide recoverable

2. **Main ions and supporting ions are no longer distinguishable in the merged spectrum graph**. In the unmerged spectrum, the concept of pseudo peaks $pp_i(\delta)$ is translated to the nodes in the graph, and the concept of the supporting peaks set $SI(pp_i(\delta))$ is translated to the set of nodes which each $sp \in SI(pp_i(\delta))$ is translated to. However in a merged spectrum, the main pseudo peak or ion and the supporting peaks are no longer distinguishable. In algorithms such as Sherenga and PepNovo this is not an issue, since their scoring does not depend on separating out the main ions and supporting ions, but only need to know what is the set of ions explaining a given fragmentation point. However this separation will become useful information in a new scoring scheme to be introduced

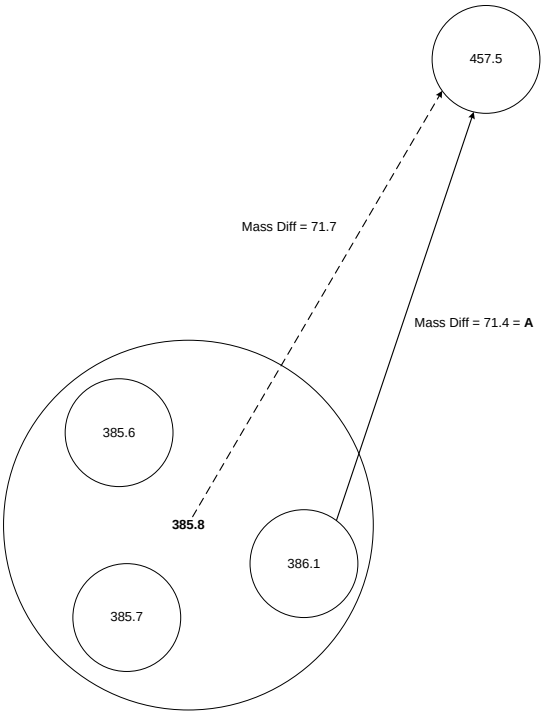


Figure 3.5: Example of Merged node causing gaps. Node with mass = 386.1 can be linked to node with mass = 457.5 with the amino acid A (Alanine) (mass=71) as the difference 71.4 is within error tolerance $\epsilon = 0.5$. However in the merged node, where the average mass is taken as the center of mass (385.8), the error exceeds 0.5 and can no longer be linked (represented by the dashed edge).

in Chapter 5.

Throughout the thesis, G will be used to refer to the extended spectrum graph, unless explicitly stated, and for all practical purposes, we only use $d = 1$ for our experiments.

Chapter 4

Characterization Study of Multi-Charge MS/MS Spectra

In the last chapter, we have introduced generalized models for higher charged peaks. In this Chapter, we address the first question raised in this thesis: namely, whether there are higher charged peaks and if so, do they help to increase the percentage of recoverable peptides. We do this by analyzing anotated multi-charge spectra (with charges up to 5) from the GPM database (Craig et al. [12], ftp://ftp.thegpm.org/quartz), as well as spectra (with charges up to 3) from the ISB (Keller et al. [34]) and Orbitrap (Tang [61]) database.

4.1 Impetus for Characterization Study of Multi-Charge MS/MS Spectra

As mentioned in Chapter 1, current *de novo* sequencing methods work well on good quality spectra of charges 1 and 2. However, they do not do well on spectra with charges 3 to 5 since they do not explicitly handle multi-charge ions (one notable exception is PEAKS by Ma et al. [41] which does conversion of multi-charge peaks into their singly-charge equivalent before sequencing). Older versions of Lutefisk by Taylor and Johnson [66] worked with singly-charged ions only, but the recent version (Lutefisk 1.0.5) have been updated to work with higher charged ions. Sherenga by Dancik et al. [13] and PepNovo works with singly- and doubly-charged ions.

Therefore, some of the higher charge peaks are mis-annotated, leading to lower recovery rates.

We therefore seek to do a systematic study of multi-charge spectra to first evaluate false positive levels on multi-charge spectra due to consideration of multi-charge peaks, and other artifacts of the mass spectrometry process. We then evaluate whether considering multi-charge ions can potentially help in improving sequencing results. The sequencing models proposed in Chapter 3 facilitates the evaluation of the quality of MS/MS spectra with respect to $PSpec(\alpha, \beta)$ (ratio of real peaks in the spectra) and $Comp(\alpha, \beta)$ (ratio of the prefix peptide fragments recovered from the spectra). The $Comp(\alpha, \beta)$ measure we define is also an upper bound on the sensitivity result obtained by any algorithm that consider charge up to β. These measures will be defined in Section 4.3.

4.2 Effect of Measurement Error, Random Peaks and Multi-charge Peaks on False Positive levels

The models described in the previous chapter is based on the ideal case in which all the masses (both theoretical and experimental) are precise. However, in reality, mass spectra contain errors. First, there is *measurement error* in mass of the peaks — this error depends on the machine and process used to generate the spectra. Second, there is an error due to the presence of *noise peaks*. Third, as more ion-types are considered in $\triangle$ (especially with multi-charge spectra), the extended theoretical spectrum becomes more dense and there may be multiple interpretations for a given experimental peak. Next, we consider these errors in turn.

False positives due to measurement error. To account for measurement error, we let ϵ be the error tolerance associated with peak measurements in the experimental spectrum. This parameter depends on the machine and process used to generate the mass spectrum. Given this error tolerance ϵ , we say that an experimental peak p_e in S *matches* a theoretical peak p_t in $TS^{\alpha}_{\beta}(\rho)$ when their mass difference is at most ϵ . In that case, we say that the p_t is a *possible interpretation* of the experimental peak p_e. We also say that p_e is a *real peak*. We extend the definition of real peaks in an experimental spectrum S to account for the error tolerance

as follows: $RP^{\alpha}_{\beta}(S, \rho, \epsilon) = TS^{\alpha}_{\beta}(\rho) \cap^{\epsilon} S$. Here, we generalize the set intersection operation to ϵ-intersection (denoted by $\cap^{\epsilon}$) which is defined as follows: $A \cap^{\epsilon} B = \{b \in B\text{: } \exists a \in A, |a-b| \leq \epsilon\}$. The standard set intersection operation corresponds to the case where $\epsilon = 0$.

False positives due to random noise peaks. We first estimate the rate of false positives due to random noise peaks. Given an experimental spectrum S with n peaks, assume there are $\gamma n (0 \leq \gamma \leq 1)$ random noise peaks. We want to estimate how many of these noise peaks will match some theoretical peaks, that is how many of them are *false positives*. Unfortunately, we do not know which are the true peaks in S, and which are the noise peaks. We consider a workaround where we generate a *random* spectrum S^R with γn peaks. Each peak in S^R is a randomly generated noise peak with mass that is uniformly distributed between $(0,M)$, where $M = m(p)$ is the parent mass of some assumed peptide ρ. Then, we match S^R with the extended theoretical spectrum for ρ (with tolerance ϵ) to get the set of false positive noise peaks. Namely, the set $PR^{\alpha}_{\beta}(S^R, \rho, \epsilon)$ is precisely the set of peaks in S^R that are false positives — those that match (with tolerance ϵ) with some theoretical peak.

We run this simulation on 2250 random spectra of charge 1, 2, 3, 4, and 5 (450 each), using an assumed peptides ρ taken from the ISB dataset (see Section 4.3.2), with an error tolerance from 0.1 to 1.0 mz unit (at 0.1 unit intervals), and with γ = 0.2, 0.4, 0.6, 0.8. We have used the full ion-type set $\triangle = (\triangle_z \times \triangle_t \times \triangle_h)$, where $\triangle_z = \{1, 2, \ldots, \alpha\}$, $\triangle_t$ = {a-ion, b-ion, c-ion, x-ion, y-ion} and $\triangle_h = \{\phi, -H_2O, -NH_3, -H_2O-H_2O, -H_2O - NH_3\}$. The ratios of false positive due to random peaks obtained from our simulation are shown in Figure 4.1. Only those for charges 3 and 5 are shown — the ones for 2 and 4 are similar and are omitted. We have also highlighted the relevant ranges (for γ and ϵ) for the three datasets we used[1]. From these simulation, we expect that the ratio of false positive due to random peaks to be (a) less than 0.15 for the GPM and ISB datasets, and (b) less than 0.1 for Orbitrap datasets.

False positives due to multiple interpretations. Multiple interpretations of a peak can lead to the anti-symmetric longest path problem as given in Section (2.4.3). However it

[1] The relevant ranges for γ and are obtained from the peak specificity results presented in Section 4.3. For the GPM-Amethyst dataset, $\epsilon = 0.5$, $0.1 \leq \gamma \leq 0.4$; for the ISB dataset, = 1.0, $0.2 \leq \gamma \leq 0.3$; and for the Orbitrap dataset, = 0.1, $0.5 \leq \gamma \leq 0.8$.

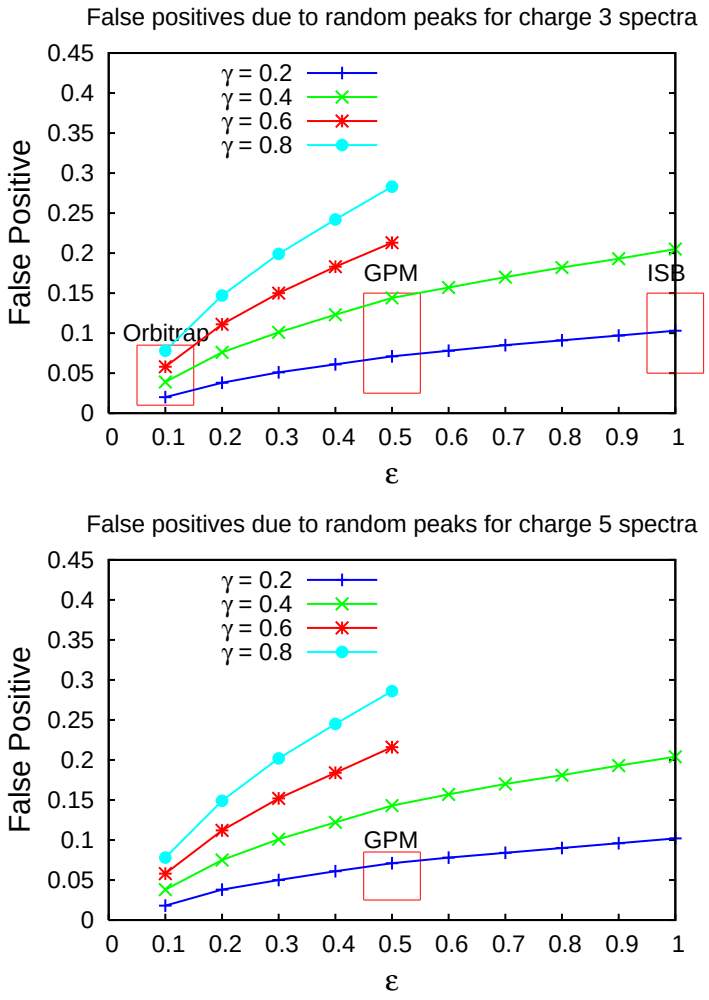


Figure 4.1: Ratio of false positive due to random noise peak matching spectra of charges 3 and 5. (The results on spectra of charges 2 and 4 are similar.)

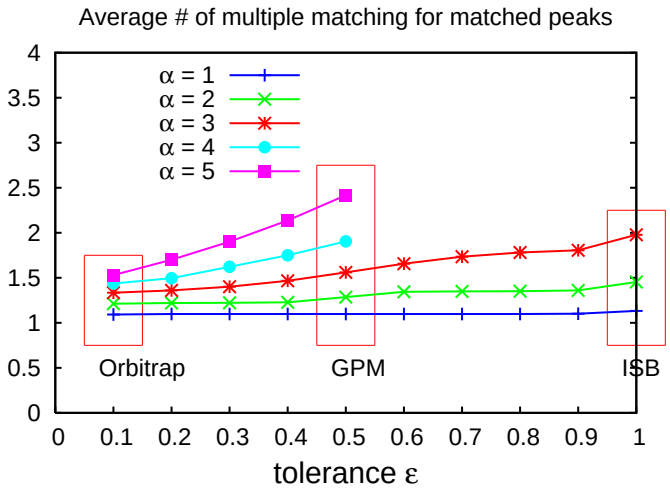


Figure 4.2: Average number of interpretation per matched peak in the experimental spectrum.

is to be noted that there are situations where two different fragmentation points can possibly coincide at the same peak due to different ion-type generation, and not all cases of multiple interpretations are necessarily wrong.

To study the extent of the problem due to multiple interpretation, we define the *set of possible interpretation* PI where $\text{PI}(p_e) = \{p_t \in TS^{\alpha}_{\beta}(\rho) : |m(p_e) - m(p_t)| \leq \epsilon\}$ for an experimental peak $p_e \in S$. Hence $\text{PI}(p_e)$ is the set of theoretical peaks that matches with p_e within tolerance ϵ. If $|\text{PI}(p_e)| = 0$, then the peak p_e is a noise peak. (Note that the proportion of noise peaks in a spectrum S depends on the machine and the process.) If $|\text{PI}(p_e)| = 1$, then the peak p_e is a matched peak with a unique interpretation. If $|\text{PI}(p_e)| > 1$, then the peak p_e is a matched peak with multiple interpretations. Let S be the subset of S that contains all the matched peaks. We want to measure the average number of possible interpretations per matched peak, denoted by $\text{API}(S')$, defined as

$$API(S') = \frac{\sum_{p \epsilon S'} |PI(p)|}{|S'|} \tag{4.1}$$

We note that $\text{API}(S)$ depends on the tolerance ϵ, and the set of ion-types $\triangle$ considered in the definition of the extended theoretical spectrum. We computed $\text{API}(S')$ using spectra of charge 1, 2, 3, 4, 5 from the GPM-Amethyst dataset (see Section 4.3.1), with an error tolerance from 0.1 to 1.0 mz unit (at 0.1 unit intervals). Figure 4.2 shows $\text{API}(S')$ over the relevant ranges for γ and ϵ. For the ISB dataset, with $\epsilon = 1.0$, the $\text{API}(S')$ increased from 1.5 to 2.0 when α goes from 2 to 3. For the GPM dataset with $\epsilon = 0.5$, there are 2.5 possible interpretations per matched peak for charge 5 spectra. Thus, for the GPM dataset, it might be important for sequencing algorithms to be able to disambiguate the interpretations of these matched peaks.

4.3 Increase in Recoverable Peptides in Multi-Charge Spectra

We now use our new model to analyze multi-charge spectra with *known* peptides. By matching the peaks in each spectrum S with our extended theoretical spectrum of the peptide, we can evaluate the abundance of the various ion-types (including higher charge ions-types), as well as

the different prefix fragments that are represented in the spectrum S. One important aim of this study is to evaluate the *significance of higher charge ions* in these multi-charge spectra.

Evaluation measures for mass spectra. We define two measures, the *peak specificity* (denoted by $PSpec(\alpha, \beta)$), and *completeness* (denoted by $Comp(\alpha, \beta)$), for evaluating multi-charge spectra from known peptide. Each measure is defined with respect to the precursor charge (or maximum charge) α, and the maximum charge considered β.

- $PSpec(\alpha, \beta)(S, \rho) = \frac{|TS^{\alpha}_{\beta}(\rho) \cap^{\epsilon} S|}{|S|} = \frac{|RP^{\alpha}_{\beta}(S,\rho,\epsilon)|}{|S|}$
- $Comp(\alpha, \beta)(S, \rho) = \frac{|TS_0(\rho) \cap^{\epsilon} PRM(S^{\alpha}_{\alpha})|}{|\rho|} = \frac{|EF^{\alpha}_{\beta}(S,\rho,\epsilon)|}{|\rho|}$

Peak specificity measures the proportion of true peaks in the experimental spectrum S. It can also be considered as a measure of the signal-to-noise ratio of S. The higher the peak specificity, the better the quality of the spectrum S. However, having high peak specificity does not necessarily mean that more of the peptide can be recovered. This is because for a given PRM, there may be *multiple* support peaks in $RP^{\alpha}_{\beta}(S, \rho)$, which lead to "double counting". The *completeness* measure avoids this by taking into account only the explained fragment masses — multiple support peaks for the *same fragment* are not double-counted.

The annotated datasets used. For our spectrum characterization study, we have selected multi-charge spectra from three sources: (a) the Amethyst data set from GPM (Craig et al. [12])(b) a dataset from the ISB (Keller et al. [34]) and (c) two datasets based on Orbitrap (Tang [61]). These spectra are annotated with their corresponding peptides that we use to generate the extended theoretical spectra for comparison.

Setup of our analysis. For each dataset (GPM and ISB), we separate the spectra into groups of different charges α = 1, 2, 3, 4, 5. For each group, we compute the group average for $PSpec(\alpha, \beta)$ and $Comp(\alpha, \beta)$ for β = 1, 2, . . ., α. In these computations, we use the annotated peptide ρ for each mass spectrum S and the full ion-type set $\triangle$ to generate the extended theoretical spectrum for ρ and use it for comparison with the spectrum S. The error tolerance ϵ used is different for each dataset.

4.3.1 Analysis of the GPM-Amethyst dataset

The GPM-Amethyst dataset are MS/MS spectra obtained from QSTAR mass spectrometers, from both MALDI and ESI sources. The entire Amethyst dataset consists of a total of 12,558 spectra of different charges from 1 to 5. We exclude spectra for which the difference between the parent ion mass and the mass of the annotated peptide exceeds a threshold of 3 Da. After this filtering, the GPM-Amethyst dataset consists of a total of 6890 spectra — consisting of 2281, 2881, 1231, 411, 86 spectra with charges $\alpha = 1, 2, 3, 4$, and 5, respectively.

Normally, QSTAR datasets are highly accurate and *usually* it is possible to determine the charge state of the peaks by examining the isotope peaks. However, the Amethyst dataset that is publicly-available from the GPM web-site are *preprocessed* datasets — each spectrum has between 20–50 peaks (usually high quality peaks). The average number of peaks per spectrum is about 40. The processed spectra have low resolution thus making it impossible to do charge state determination using isotopic peaks. (We do not have access to the corresponding unprocessed spectra.) To analyze the GPM-Amethyst dataset, we use error tolerance $\epsilon = 0.5$.

Peak specificity results for GPM dataset. The average peak specificity results for this dataset are shown in Figure 4.3. There are five curves corresponding to different precursor charge $\alpha = 1, 2, 3, 4, 5$. For a fixed α, the peak specificity increases significantly as more higher charge ions are considered (as β increases). For example, when $\alpha = 5$, the peak specificity increases from 0.53 when $\beta = 2$, to 0.90 when $\beta = 5$. For higher-charge spectra, the peak specificity are high — between 0.72 to 0.90.

The significance of higher charge ions is measured by the difference between $PSpec(\alpha, \alpha)$ and $PSpec(\alpha, 2)$. An algorithm that uses $\beta = 2$ is limited to a peak specificity of $PSpec(\alpha, 2)$ while an algorithm that considers all the charge ions ($\beta = \alpha$) can potentially achieve a higher peak specificity of $PSpec(\alpha, \alpha)$. Figure 4.3 clearly shows that this difference is significant, and the gap increases with α.

We note that this impact is preserved even in the presence of false positive due to noise peaks. To see this, we observe that for a given spectrum with charge α, the expected rate of false positive is the same for all values of β. Thus, if the rate of false positive is ψ, then the

real peak specificity is $(PSpec(\alpha,\beta)-\psi)$. Thus, false positive merely lower the absolute values of the $PSpec(\alpha,\beta)$, but not their relative order. In particular, the impact due to higher charge ions $(PSpec(\alpha,\beta) - PSpec(\alpha,2))$ is independent of the rate of false positive due to noise peaks, ψ.

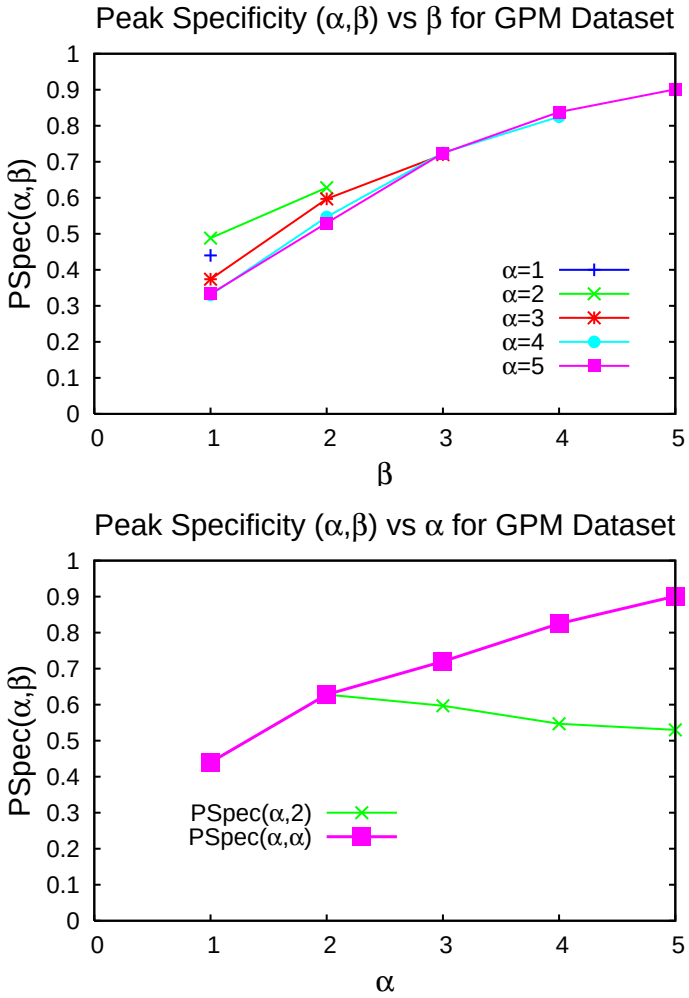


Figure 4.3: Peak specificity results for the GPM-Amethyst dataset. The five curves in the Figure on the left shows $PSpec(\alpha,\beta)$ for $\alpha = 1$ to 5. The graph on the right shows the impact of considering higher charged ions. The lower curve, $PSpec(\alpha,2)$ considers only charges 1 and 2 ($\beta = 2$). The upper curve, $PSpec(\alpha,\alpha)$ considers all charges ($\beta = \alpha$).

Completeness results for GPM dataset. The completeness results for the GPM-Amethyst dataset are shown in Figure 4.4. For a fixed α, the completeness also increases significantly with β, showing that more fragments can be recovered by considering higher charges. To highlight the impact of the higher charge ions on the completeness, we also plot $Comp(\alpha,\alpha)$ and $Comp(\alpha,2)$ against α as shown in the right of Figure 4.4. For each $\alpha = 3, 4, 5$, this dispar-

ity is shown by the difference between the top and bottom curves. We note that the disparity increases with α as seen from the widening gap. For example using $\beta = 4$ for charge 4 ($\alpha = 4$) data compared to using $\beta = 2$ is an improvement of about 27% in completeness.

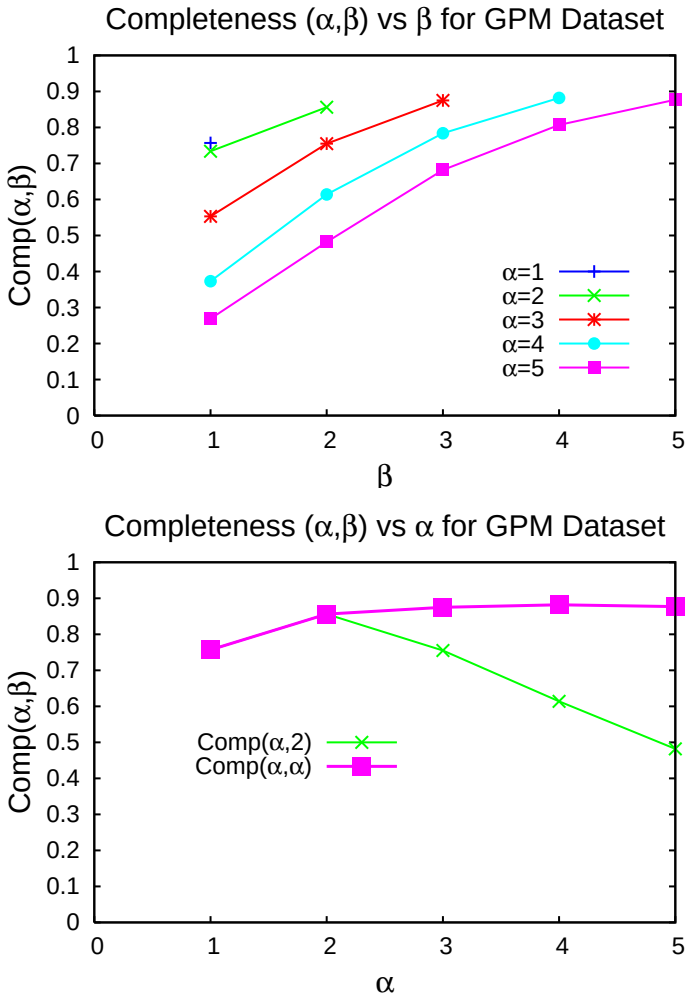


Figure 4.4: Completeness results for the GPM-Amethyst dataset. The five curves in the left Figure show $Comp(\alpha, \beta)$ for $\alpha = 1$ to 5. The graph on the right shows the impact of considering higher charge ions on completeness — by plotting $Comp(\alpha, 2)$ and $Comp(\alpha, \alpha)$.

4.3.2 Analysis of the ISB dataset

The ISB dataset are Iontrap data generated using an ESI source from a mixture of 18 proteins and consists of 5334 spectra with charge $\alpha = 1$, 2, and 3. For each multi-charge spectrum, the machine outputs two spectra (one for charge 2 and one for charge 3) because there is not enough resolution to determine the precursor charge. Both spectra are then searched using SEQUEST

to find the best matching peptide among them — based on a better SEQUEST XCorr score (Keller et al. [34]).

We further exclude spectra with low XCorr scores (≤ 2.0) or if the the difference between the parent ion mass and the mass of the annotated peptide exceeds a threshold of 3 Da. After this filtering, the ISB dataset consists of a total of 2267 spectra — consisting of 50, 1329, 888 spectra with charge $\alpha = 1$, 2, and 3, respectively. The ISB dataset have between 200–700 peaks per spectrum and an average of 400 peaks per spectrum. There are, generally, more noise peaks and ISB spectra generally have lower peak specificity. To analyze the ISB dataset, we use error tolerance $\epsilon = 1.0$.

Peak specificity results for ISB dataset. The average peak specificity results for the ISB dataset are shown in Figure 4.5. The three curves correspond to different precursor charge $\alpha = 1$, 2, 3. The peak specificity for multi-charge spectra is a little lower than those for the GPM dataset — between 0.63 and 0.77. For $\alpha = 3$, the peak specificity increases only slightly when charge 3 ions are included. This seem to indicate that relatively few charge 3 fragments are produced.

Completeness results for ISB dataset. The completeness results for the ISB datasets are shown in Figure 4.6. The results show that the ISB spectra have very high completeness values — close to 1 even when $\beta = 2$. This means that for the ISB dataset, almost all of the fragments are supported by some peaks in the spectrum even when $\beta = 2$. Thus, there is almost no increase in the completeness when charge 3 ions are included. However, the slightly larger peak specificity for $\beta = 3$ means these fragments are supported by more peaks — which gives rise to higher scores for these spectra.

4.3.3 Analysis of the Orbitrap dataset

While the GPM datasets are true multi-charge datasets, the spectra have been preprocessed. Ideally, our analysis should have been done on the original unprocessed spectra, but these are not available to us. The ISB datasets are unprocessed, but they have low resolution ($\epsilon = 1.0$) and so are not ideal for our study.

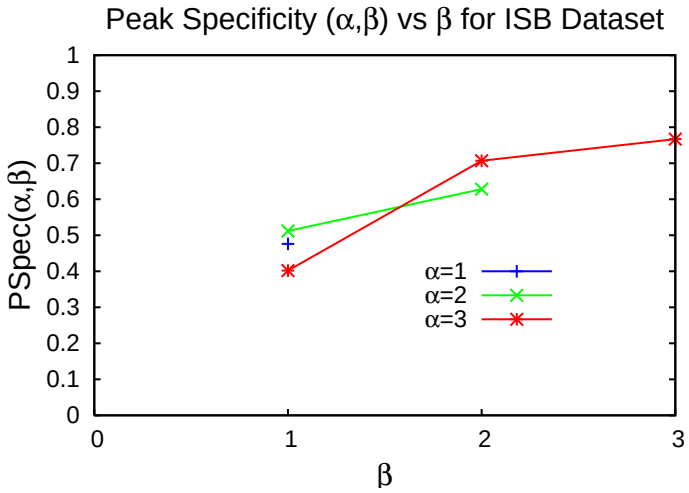


Figure 4.5: Peak specificity of the ISB dataset. The three curves show $PSpec(\alpha, \beta)$ for $\alpha = 1, 2, 3$. Considering charge 3 ions improves the peak specificity slightly.

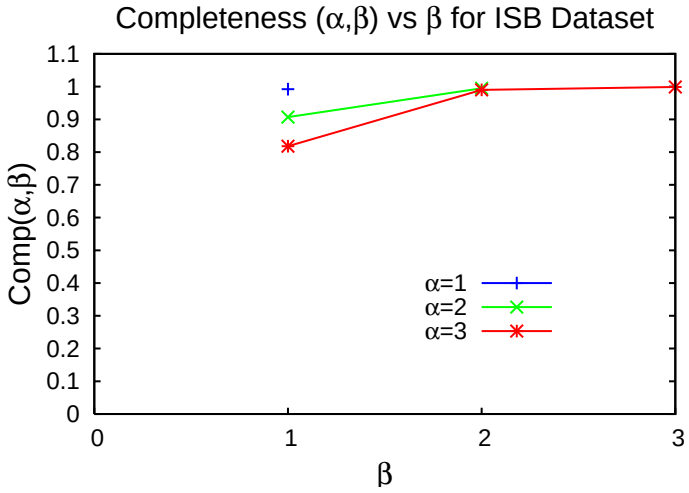


Figure 4.6: Completeness of the ISB dataset. The three curves shows $Comp(\alpha, \beta)$ for $\alpha = 1, 2, 3$. Completeness scores are very high for the ISB dataset.

To complement this study, we also obtained two sets of annotated mass spectra (of charge $\alpha = 1, 2, 3$) that are produced using a high resolution Orbitrap mass spectrometry process — Orbitrap-FT (with 122 spectra) and Orbitrap-LTQ (with 252 spectra). We filter these datasets using the same conditions for the ISB dataset. After filtering, our Orbitrap-FT dataset consists of a total of 103 spectra, consisting of 4, 78, 21 spectra with charge $\alpha = 1$, 2, and 3, respectively; and our Orbitrap-LTQ dataset contains 240 spectra, with 32, 165, 43 spectra with charge $\alpha = 1$, 2, and 3, respectively. The Orbitrap-FT dataset has several hundred peaks per spectrum, while the Orbitrap-LTQ dataset has several thousand peaks per spectrum. Since these are high resolution datasets, we use an error tolerance $\epsilon = 0.1$ to analyze the Orbitrap datasets.

Peak specificity results for Orbitrap datasets. The average peak specificity results for the two Orbitrap datasets are shown in Figures 4.7 and 4.8. The peak specificity for the multi-charge Orbitrap-FT datasets are between 0.43 and 0.49, while those for Orbitrap-LTQ is even lower, between 0.11 and 0.18. Similar to the ISB datasets, there is a slight increase in peak specificity when charge 3 ions are included.

Completeness results for the Orbitrap datasets. The completeness results for the two Orbitrap datasets are shown in Figures 4.9 and 4.10. The Orbitrap-FT datasets has completeness of between 0.70 and 0.78, and those for Orbitrap-LTQ is even higher, between 0.98 and 0.99. Thus, there are sufficient real peaks in these spectra to explain most of the peptide fragments, even in the presence of a lot of noise peaks. Considering $\beta = 3$ does not improve the completeness by much from $\beta = 2$ (about 3% for Orbitrap-FT data and about 1% for Orbitrap-LTQ data).

4.4 Discussion and Conclusion on the analysis of multi-charge spectra

Our analysis of multi-charge mass spectra suggest that for true multi-charge GPM dataset, there are higher charged peaks/ions and they have significant impact on both the peak specificity and completeness, therefore potentially improving the amount of peptide recoverable. However, for

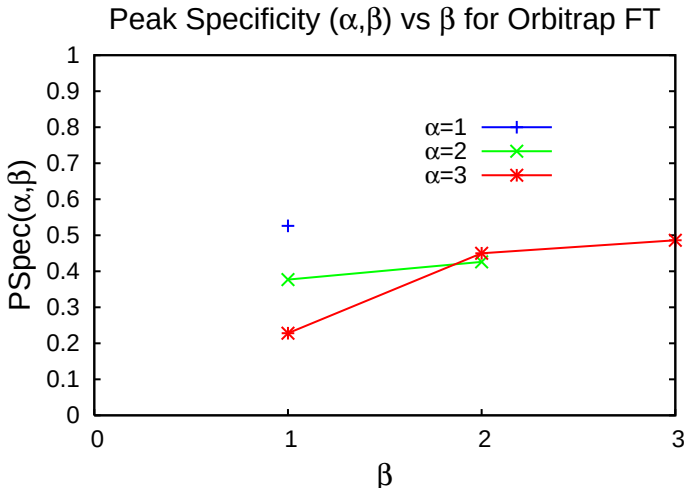


Figure 4.7: Peak specificity of the Orbitrap-FT dataset. The three curves shows $PSpec(\alpha, \beta)$ for α = 1, 2, 3. Ions of charge 3 gives slight improvement.

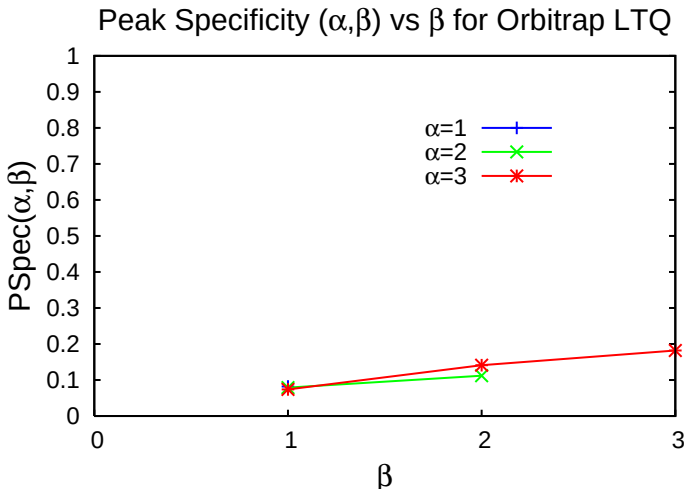


Figure 4.8: Peak specificity of the Orbitrap-LTQ dataset. The three curves shows $PSpec(\alpha, \beta)$ for $\alpha = 1, 2, 3$. Ions of charge 3 gives slight improvement.

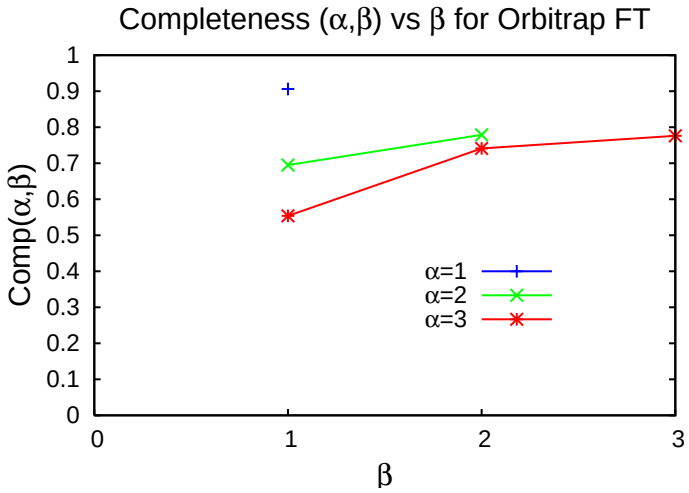


Figure 4.9: Completeness for Orbitrap-FT dataset. The three curves shows $Comp(\alpha, \beta)$ for $\alpha = 1$, 2, 3. Considering ions of charge 3 improves slightly the completeness.

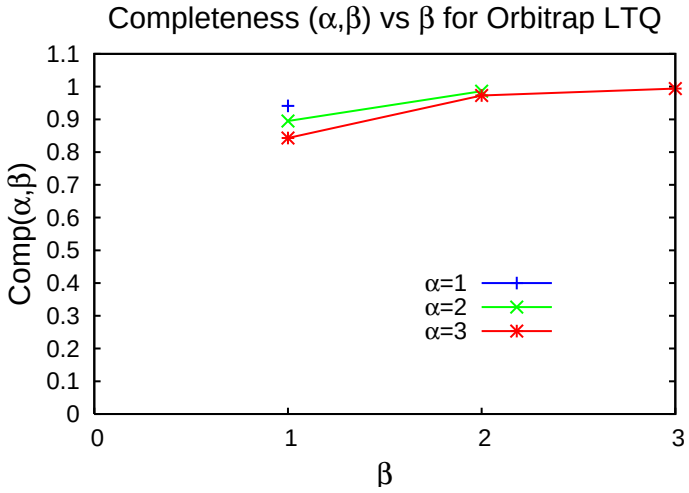


Figure 4.10: Completeness for Orbitrap-LTQ dataset. The three curves shows $Comp(\alpha, \beta)$ for $\alpha = 1$, 2, 3. Considering ions of charge 3 improves slightly the completeness.

the charge 3 spectra from the ISB and the Orbitrap datasets, charge 3 ions give only very slight increase in completeness. The slightly better improvement in peak specificity suggests that charge 3 ions may be useful as supporting ions.

Specifically, for GPM data, the improvement in $PSpec(\alpha, \beta)$ when considering $\beta \geq 3$ is in spite of the presence of false positive due to noise peaks. This leads to the conclusion that there are higher charged peaks in GPM data. We also show that the *Comp* measures for GPM dataset are low for $\beta = 1$ and $\beta = 2$. This imply that *any* algorithm that considers only charge 1 or 2 ions will suffer from low sensitivity. The results also show that the *Comp* measure increases significantly with largerβ. This implies that considering higher charged ions improves the amount of peptide recoverable, and therefore it is important for peptide sequencing algorithms to consider higher charge ions for QSTAR multi-charge data.

For ISB and the Orbitrap datasets, charge 3 ions give only very slight increase in peak specificity and completeness of charge 3 spectra. This seems to indicate that for these charge 3 spectra, relatively few charge 3 fragments ions are produced from the charge 3 precursors. However this does no indicate that charge 3 ions are useless since the support they give to existing charge 1 and 2 ions can help to determine the correct sequence during sequencing. This issue will be addressed in Chapter 5.

The number of multiple matching for peaks is low for all datasets except for GPM charge 5 spectra.

Chapter 5

MCPS (Mono-Chromatic Peptide Sequencer) for Multi-Charge Mass Spectra

We have shown, in the last chapter, that there are higher charged peaks in multi-charge spectra and they contribute to sequencing results by increasing the potential amount of peptide recoverable. In this Chapter, we address the second question on whether it is possible to devise better de novo sequencing algorithms that considers these higher charged peaks.

We present a new algorithm MCPS (Mono-Chromatic Peptide Sequencing) for peptide sequencing which gives better sequencing result especially for higher charged spectra ($> 5\%$ better peptide recovery compared to the next best algorithm PepNovo for charge 3 ISB data), using the generalized model developed in Chapter 3.

The main idea of MCPS is to isolate and identify mono ion-type tags which are peptide tags supported by consecutive peaks of the same ion-type. MCPS uses mono ion-type tags in two ways: (i) to define a novel mono-chromatic score function $MCScore(P)$ that amplifies the weight of mono ion-type tags in putative peptide P, and (ii) to effectively prune away noise peaks. MCPS finds the peptide P that maximizes the mono-chromatic score $MCScore(P)$. We show that this problem is equivalent to that of find a suffix-k path-dependent longest path in

a DAG (directed acyclic graph) which can be solved using a dynamic programming algorithm. (In practice, we use k=1,2,3.)

We will first define the Mono-Chromatic Scoring function which is the heart of the MCPS algorithm, and then we will show the steps that MCPS uses to sequence a peptide.

5.1 New Scoring Scheme - Mono-Chromatic Scoring Function

In this section, we present our new mono-chromatic scoring function for putative peptides that amplifies sub-paths of the same ion-types. When we refer to the *color* of a vertex, we are referring to its ion-type. These two terms will be used interchangeably throughout the rest of the thesis.

Mono-Chromatic Scoring of a Path. We introduce a novel scoring, $MCScore(P)$ of a path P which differentiates between *mono-chromatic sub-paths* (of the *same* ion-type) and *mixed edges* (supported by different ion-types) in P. To formally define $MCScore(P)$, we need several definitions.

A path $P = (w_0, e_1, w_1, e_2, w_2, ..., e_k, w_k)$ is called a *mono-chromatic path* if each edge $e_j = (w_{j-1}, w_j)$ in P is mono-chromatic (defined in Section 3.3.1) for $j = 1, 2, ..., k$. Otherwise, the path is called a *mixed path*. Note that in a mono-chromatic path P, all the vertices in P are of the same ion-type.

In de novo sequencing, we search the extended spectrum graph G for paths. Each path P in G represents a putative peptide given by the sequence of amino-acid labels on the edges in the path P.

The significance of using the extended spectrum graph lies in the fact that each path P can be *uniquely* decomposed into maximal mono-chromatic sub-paths or tags. More precisely, the path $P = (w_0, e_1, w_1, e_2, w_2, ..., e_k, w_k)$ in G can be uniquely decomposed into maximal monochromatic sub-paths that are linked by mixed edges and is given by

$$P = ([P_0]e_{m1}[P_1]em_2...e_{mr}[P_r])$$

each $[P_j]$ is a maximal mono-chromatic sub-path for $j = 0, 1, ..., r$ and consecutive monochromatic tags $[P_{j-1}]$ and $[P_j]$ are linked via mixed edge e_{mj}. (Note that if two consecutive edges, e_{j-1} and e_j are mixed edges, then the intervening sub-path $[P_{j-1}]$ is the empty sub-path).

A mono-chromatic path of length k represents a strong signal consisting of $(k + 1)$ consecutive pseudo-peaks of the same ion-type. Therefore, we believe that mono-chromatic sub-path represents stronger signals compared to mixed sub-path of the same length.

As an example we have two alternate fictitious paths in the graph shown in Figure 5.1, both representing the peptide tag **GFGGED**. Even though both represents the same peptide tag, the top path $P_1 = (v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_e)$ consists of stronger signals than the bottom path $P_2 = (v_0, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_e)$. This is because in P_1, the sub-path $P_1' = (v_1, v_2, v_3, v_4, v_5)$ representing the tag **GFGG** is a maximal mono-chromatic sub-path made up of y-ions. This sub-path when compared to a similar mixed sub-path $P_2' = (v_8, v_9, v_{10}, v_{11}, v_{12})$ in P_2 should reflect a stronger signal for **GFGG**. If for example we boost the score of the sub-path P_1' by multiplying $SScore(P_1')$ (the **simple scoring of a path** defined in Section 2.3.2) by the length of the tag **GFGG** as shown in the figure, the score of P_1' (22) will be greater than that for the P_2' (5.5), even though their edge weights are the same. In the end, P_1 will have a higher score (26.5) than P_2 (8.5).

Hence, the novel idea is to amplify the score of mono-chromatic sub-paths in P based on their lengths. Our mono-chromatic scoring function, $MCScore(P)$ is defined as follows:

$$MCScore(P) = \sum_{j=0}^{r} \lambda(l_j) * SScore([P_j]) + \sum_{j=1}^{r} w(e_j) \tag{5.1}$$

l_j is the length of the $[P_j]$. The λ function takes into consideration the length of P_j to amplify the score of P_j. We remark that in our score function $MCScore(P)$, the *actual* weight contribution of an edge e is not fixed, but changes depending on the length of the maximal mono-chromatic sub-path that contains e via the multiplier $\lambda(l_j)$.

This however does not mean that a strong signal pertaining to an ion-type that is rarely see will have a higher score, since the original weights of the edges for the mono-chromatic tag is low, the strong signal will not increase by much even when boosted. Worse, for ion-types

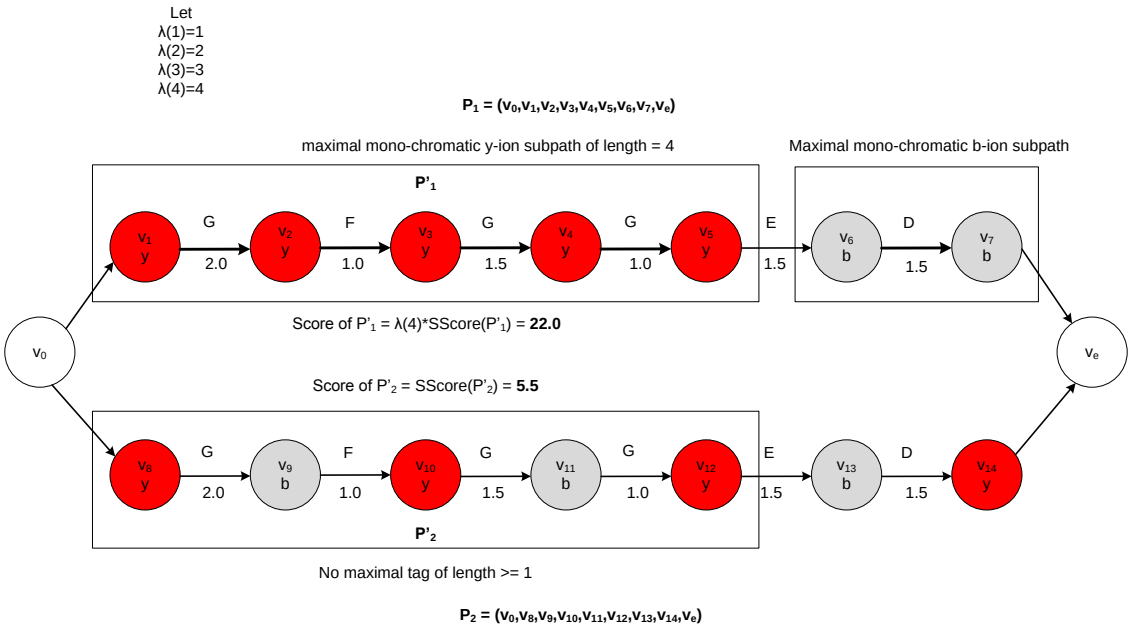


Figure 5.1: Example of mono-chromatic path vs a mixed path. Top path P_1 consists of 2 mono-chromatic maximal sub-paths, one belong to y-ion, the other belonging to b-ion. The bottom path P_2 is a mixed path consisting of all mixed edges. Even though they represent the same tag **GFGGED**, the top path has a higher score due to the fact that the two maximal mono-chromatic sub-paths give a strong signal for the fragment they represent.

with negative weights (rare ion-types), the boosting implies a bigger drop in the score of the tag. Thus in our scoring function we only boost ion-types which have a high probability of observation in the mass spectrum.

In general, a function such as $MCScore(P)$ violates the optimality principle which is the basis of many shortest path on a DAG algorithms. As an example, given 2 paths $P' = (v_0, v_1, v_2, v_3, v_7)$ and $P'' = (v_0, v_4, v_5, v_6, v_7)$ connecting to $P = (v_7, v_8, v_9, v_e)$ in Figure 5.2, we see that $MCScore(P'') = 18.5$ and this is a better score than $MCScore(P') = 7.5$. For the optimality principle to work, at node v_7, using P'' should be the optimal sub-path, and the score of $P'' + P$ should be better than that for $P' + P$. However due to the extension of the mono-chromatic y-ion sub-path to (v_3, v_7, v_8, v_9) as shown by the bolded nodes, $MCScore(P' + P) = 31.5$ which is better than $MCScore(P'' + P) = 31$. This violates the optimality principle since we cannot determine whether to use P' or P'' until we have gone through both the paths $P' + P$ and $P'' + P$.

Using our mono-chromatic scoring of a path, the peptide sequencing problem will then become one of finding the *path-dependent longest path* in a DAG that was previously study in Tan and Leong [60]. In general, path-dependent optimal path is an NP-Hard problem Tan and Leong [60].

In practice, we can limit this dependence to a suffix of length at most k. Thus, mono-chromatic sub-paths of length greater than k will be amplified by the same factor as that for those of length k. Mathematically, this means that $\lambda(h) = \lambda(k)$ for all $h \geq k$. We call this the *suffix-k path-dependent longest path* in a DAG. This can be solve in polynomial time Tan and Leong [60].

In the next section we will give the MCPS algorithm that uses the Mono-Chromatic Scoring function ($MCScore$) to sequence a peptide.

5.2 MCPS (Mono Chromatic Peptide Sequencer)

Given the above mono-chromatic scoring function, MCPS (Mono-Chromatic Peptide Sequencer) takes the following steps to sequence a peptide.

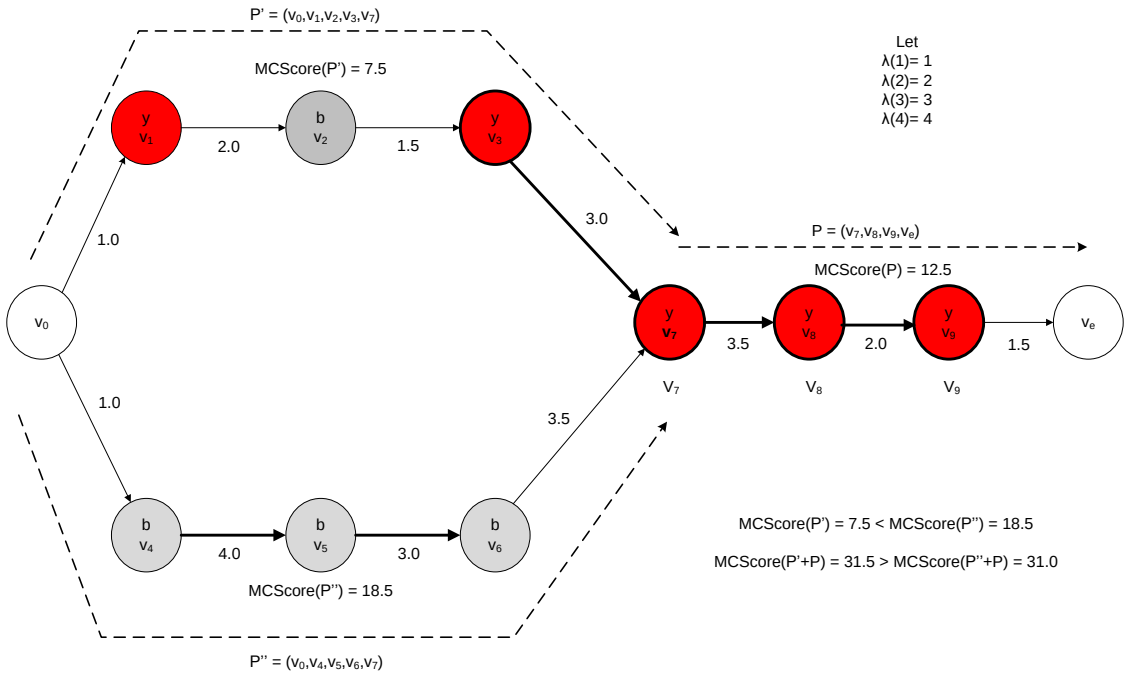


Figure 5.2: MCScore violates optimality principle. $MCScore(P') = 7.5 < MCScore(P'') = 18.5$. In order for the optimality principle to work, at node v_7, P'' should be the optimal sub-path to use, and score of P"+P should be better than P'+P. However, $MCScore(P' + P) = 31.5 > MCScore(P'' + P) = 31$ due to the extension of the mono-chromatic y-ion sub-path in P'+P to (v_3, v_7, v_8, v_9) (bolded nodes). This violates the optimality principle that many shortest path algorithms depend on.

MCPS

1. Peak filtering
2. Build extended spectrum S^{α}_{β} from spectrum S
3. Build extended spectrum graph $G(S^{\alpha}_{\beta})$ given extended spectrum S^{α}_{β}
4. Prune off noisy vertices in $G(S^{\alpha}_{\beta})$ to get pruned spectrum graph $G_p(S^{\alpha}_{\beta})$
5. Bridge vertices in $G_p(S^{\alpha}_{\beta})$ to get final spectrum graph $G_b(S^{\alpha}_{\beta})$
6. Scoring edges in $G_b(S^{\alpha}_{\beta})$
7. Sequence peptide
8. Post-process candidate peptides

Step 1-5 seeks to generate a subgraph of the extended spectrum graph $G_b(S^{\alpha}_{\beta})$ which is small in size, thus making it manageable for de novo sequencing and also filters away a lot of the noisy paths (false candidate peptides). Step 6-7 uses the new score function to generate good candidate peptides (eliminates more noisy paths) and Step 8 seek to improve the ranking of the candidate peptides generated.

5.2.1 Peak Filtering

For our experiments, we have used ISB Keller et al. [34], ISB2 Klimek et al. [37]and GPMCraig et al. [12] datasets (refer to Chapter 6 for more details). ISB and ISB2 spectrums consists on the average ~250 peaks. These consists of a few high intensity peaks among many low intensity peaks. Since ISB data is usually not very heavily pre-processed, a lot of the peaks are noise. A simple peak filtering method used in our pre-processing step is to take only the top 100 peaks order by their intensity. This reduces the amount of noisy peaks selected, and reduces the size of the final extended spectrum graph. GPM spectrums on the other hand consists of on the average only about 40 peaks. These are heavily pre-processed spectra, and most of the time, we will use all the peaks present in the spectra using our filtering criteria.

Parent Mass Correction. Experiments were also done on correcting the parent ion mass given in the spectrum. These results however are not yet included in the MCPS algorithm. Please refer to Appendix A for more details.

5.2.2 Build extended spectrum S^{α}_{β} from spectrum S

Building the extended spectrum S^{α}_{β} from the experimental spectrum S has been described in detail in Chapter 3. We have used a greedy ranking algorithm to obtain different ion-type sets for use for the different spectra grouped by their maximum charge α.

An example of the ion-type set $\triangle$ used for charge 1 ISB2 data is the set $\triangle$ ={(+1,y), (+1,b), (+1,y,-water), (+1,b,-water), (+1,y,-(water+ammonia)), (+1,b,-ammonia), (+1,x,-ammonia)}. For more details on the ion-type sets used and the greedy ranking algorithm please refer to Section 6.2.1.

5.2.3 Build extended spectrum graph $G(S^{\alpha}_{\beta})$ given extended spectrumS^{α}_{β}

In this step we build the extended spectrum graph $G(S^{\alpha}_{\beta})$ given the extended spectrum S^{α}_{β} as described in Chapter 2.

5.2.4 Prune noisy vertices in $G(S^{\alpha}_{\beta})$ to get pruned spectrum graph $G_p(S^{\alpha}_{\beta})$

Even after peak filtering, $G(S^{\alpha}_{\beta})$ is still a noisy spectrum graph. An important step will be to remove noise by removing nodes in the extended spectrum graph which are unlikely to be real fragmentation points in the canonical peptide. In order to do this, we remove all vertices which do not participate in a mono-chromatic path of at least some length l. The rationale is that in most sequencing results, the good candidate peptides will be paths in $G(S^{\alpha}_{\beta})$ containing mono-chromatic maximal tags of at least a certain length l.

Now not all mono-chromatic maximal tags in a good candidate peptide will be long, only those of abundant ion-types. For an analysis of the probability of observation of a mono-chromatic tag of an ion-type δ of length l please refer to Appendix B. We split the ion-type set into the set of top x ion-types $\triangle_x$ which will be considered for pruning using the above strategy.

Vertices and the associated edges pertaining to the remaining $\triangle - \triangle_x$ ion-types will first be temporarily "*switched off*" in this step before we begin pruning, and will be used in the next step of the algorithm (the bridging step). We denote the extended spectrum graph obtained in this step the pruned spectrum graph $G_p(S^{\alpha}_{\beta})$.

$G_p(S^{\alpha}_{\beta})$ is defined as the sub-graph of $G(S^{\alpha}_{\beta})$ which contains all vertices in the set $V_p = \{v \mid v \in V(G(S^{\alpha}_{\beta}))$, $ion\_type(v) \in \triangle_x$, and $\exists[P]$ in $G(S^{\alpha}_{\beta})$ where $length([P]) \geq l$, and where v is in $[P]\}$, and all edges in the set $E_p$induced by V_p. These rest of the vertices which are not "*switched off*" or not in V_p are removed along with the edge set they induce.

Note that edges linking vertices of different color in V_p remains and this is required since it reflects the actual fragmentation process, where it is unlikely to get a peptide fragmented by solely one ion-type, and a candidate path usually contains mono-chromatic tags of different ion-types. In fact different mass regions of the peptide have different probability distribution of the ion-types, since some ion-types are more abundantly found in one region and not in others.

It is to be noted that we do not perform the pruning or bridging on GPM data, since they produce very small spectrum graphs due to the small number of peaks their spectrum contains, and also for charge 3 and higher data, pruning severely affects the amount of peptide (refer to Section 6.2.2) that can be sequenced. Instead for GPM data, we use $G(S^{\alpha}_{\beta})$ at the sequencing step.

Performing Pruning. We can determine which are the set of such vertices by first finding all vertices which participate in a path of at least length l. This is done by considering each node v in $G(S^{\alpha}_{\beta})$ as root and doing a BF traversal until we hit a depth of l, or we have finished traversing the subtree anchored at v. All the nodes in paths of length l are marked. After going through all the nodes, the unmarked nodes are the ones which do not participate in any path of at least length l. This set of vertices is correctly computed, since if any node u in this set resides on a path of at least length l, they will have been found and marked when we perform BF traversal on any node in that path which is $\leq l$ edges away from it. In all, we perform $n * 20^l$(20 is the maximum branching factor since there are only 20 amino acids) node visits. Since l is fixed, the computational complexity of this step is $O(n)$. All the unmarked nodes are

then pruned from the spectrum graph.

5.2.5 Bridge vertices in $G_p(S^\alpha_\beta)$ to get final spectrum graph $G_b(S^\alpha_\beta)$

In this step, we re-introduce the remaining ion-types $\triangle - \triangle_x$ by "*switching on*" a vertex in $G_p(S^\alpha_\beta)$ whose ion-type is in $\triangle - \triangle_x$ only if they can bridge two vertices of ion-type in $\triangle_x$. This is in vein with the idea that low probability ion-types found in consecutive fragmentation points are highly unlikely (refer to Appendix B for an analysis of probability of such an occurrence) and will usually be a bridge for high probability ion-types. We denote the final mono-chromatic extended spectrum graph after the bridging step as $G_b(S^\alpha_\beta)$. In Section 6.2.2.1, we show there is a huge drop in the number of nodes and edges going from $G(S^\alpha_\beta)$ to $G_b(S^\alpha_\beta)$, while still maintaining a good upper bound on the amount of the canonical peptide recoverable.

5.2.6 Scoring edges in $G_b(S^\alpha_\beta)$

In order to provide the raw weights for the nodes, we have used the probabilistic model in Liu et al. [38] (explained in detail in Chapter 2) to train the log likelihood ratios of observing an ion-type at a given fragmentation point. Since each vertex in our final extended spectrum graph $G_b(S^\alpha_\beta)$ corresponds to a possible fragmentation point, this is equivalent to scoring each vertex by the log likelihood of observation of its main ion.

The weights of **supporting ions** (described in Section 3.2.1) are factored into the final weight of a vertex v as follows

$$w'(v) = \frac{w(v) + \sum_{j=0}^{|SI(v)|} w(m)}{\mu} \tag{5.2}$$

where $m \in SI(n)$ and μ is some constant. Since the suffix-k mono-chromatic scoring function $MCScore(P)$ works on the edge scores, the weights of the nodes are pushed onto the edges by using the function $w(e) = \frac{w'(in\_node)+w'(out\_node)}{2}$. Finally, weights of **supporting edges** (described in Section 3.3.1) are factored into the final weight of an edge e as follows

$$w'(e) = \{ \begin{array}{ll} \frac{w(e)+\sum_{j=0}^{|SE(e)|} w(f)}{\mu'} & if\, suffix\, k > 1 \\ w(e) & otherwise \end{array} \tag{5.3}$$

where $f \in SE(n)$ and μ' is some constant. For our algorithm we have used $\mu = 2$ and $\mu' = 2$. Note that for mixed edges $w'(e) = w(e)$. If the suffix $k = 0$, no supporting edges are taken into consideration.

5.2.7 Sequence peptide

After building the mono-chromatic extended spectrum graph $G_b(S^{\alpha}_{\beta})$, we proceed to perform peptide sequencing. As stated in Section 5.1, this is equivalent to finding the *suffix-k path-dependent longest path in a DAG* using the Mono-Chromatic Scoring $MCScore(P, h)$. Details of the DP formulation for the algorithm is given in Section 5.3.

5.2.8 Post-processing of candidate peptides

During sequencing, we usually generate the top 100 candidate peptide, using a modified version of Yen's algorithm (Yen [71]) that consists of disabling paths that have already been generated in the spectrum graph and finding the next best path and so on until the top 100 is generated. Since actually generating the top 100 currently takes too long to run for *suffix-k* values ≥ 3, we generate the actual top 10 and take all remaining sequences in the pool (which may not be the actual rank 11-100) which are then ranked by their score and appended to the top 10 to make the top 100.

These candidate peptides then undergo post-processing to improve the sequencing result. We tackled two problems during the post-processing stage. The first is the anti-symmetric path problem described in Section 2.4.3. The second is the problem of *Competing Sub-paths*.

5.2.8.1 Anti-Symmetric Path Problem

In our experiments, we tried tackling the anti-symmetric path problem by first generating the top 100 candidates peptides regardless of how many are requested. These 100 candidates are

then checked if any of them give a peak multiple ion-type interpretations, that is a peak is used to explain 2 different fragmentation point in the peptide. All candidates which do so are pushed to the back of candidates which do not do so. The top n candidate peptides requested are then output. This gives a soft solution to the anti-symmetric longest path problem (refer to Section 2.4.3 for more details), since we still allow for such candidates in our solution, but we do not rank them as high as the others, instead of totally rejecting them.

However, we found that this did not impact the result of the top ranking candidate peptides or the best ranking peptide within the top 100 candidates for the training data. Thus we concluded that the anti-symmetric problem rarely occurs for good candidate sequences generated by MCPS, and did away with this step in the final MCPS algorithm.

5.2.8.2 Competing Sub-paths

Due to the path dependent scoring and the fact that each node only represents the interpretation of a peak given 1 ion-type, mono-chromatic sub-paths of different ion-types basically compete against each other. This includes sub-paths that represent similar fragments of the canonical peptide but are formed from different ion-types. The following situation exemplifies this.

Figure 5.3 shows part of the actual spectrum graph generated for an actual experimental spectrum (LQ20060105_s_18MIX_12.1230.1230.1.dta) of the ISB2 dataset. The canonical peptide associated with the experimental spectrum is AGFAGDDAPR. The spectrum graph shows the paths representing the top ranked candidate peptide (score = 23.57) and the 10th ranked candidate peptide (score = 18.12). The numbers on the edges represent the score of the edges. Nodes representing +1 y-ions are coded red and nodes representing +1 b-ions are coded yellow.

The top ranked candidate peptide is given by the path $(v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9)$, where edges are implicitly represented by consecutive vertices. This path represents the candidate peptide tag [35.0719]SQGNPDA[253.03]. The 10th ranked candidate peptide is given by the path $(v_0, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}, v_9)$. This represents the candidate peptide tag [41.1905]SFNEDA[253.295].

The top ranked peptide tag [35.0719]SQGNPDA[253.03] matches the canonical peptide at the bolded amino acids AGFAGD**DA**PR (matches 2 amino acids), while the 10th ranked peptide tag [41.1905]SFNEDA[253.295] matches that canonical peptide at AG**F**AGD**DA**PR (matches 3 amino acids).

From the figure we can see that the sub-paths (v_6, v_7, v_8, v_9) and $(v_{14}, v_{15}, v_{16}, v_9)$ represent the same fragment DA[253.03] and are in fact competing against each other since they belong to 2 different paths.

An alternative route to get [41.1905]SFNEDA[253.295] might be taken in order to make use of the higher edge score of 1.9286 compared with 1.2679 (v_7, v_8) vs (v_{15}, v_{16}). This can be done by using the edge from v_{14} to v_7 (dashed line) and continuing from there. However, because of a change from b- to y-ion, the score of edge (v_{14}, v_7) is 0.05 which is lower than that of edge (v_{14}, v_{15}) with a score of 1.9436. We won't be able to get the benefit of the score of the edge (v_7, v_8), since the drop from 1.9436 to 0.05 is greater than the increase from 1.2679 to 1.9286.

In fact what we want is swap (v_{14}, v_{15}, v_{16}) with (v_6, v_7, v_8) which will increase the score of [41.1905]SFNEDA[253.295] to 25.45 and this is better than the current top ranked candidate [35.0719]SQGNPDA[253.03]. However this is not possible during the sequencing step. This problem is due to the fact that it is hard to switch from one ion-type to the other ion-type when you have a long substring (since the multiplier increases) and the cost to switch increases too.

In our post-processing step, we tackle this problem by first restricting the set of edges and nodes in the spectrum graph to only those induced by the set of candidate peptides. We then form groups out of all the edges representing the same peptide fragment (same amino acid at the same fragmentation point). We re-assign the weights of the edges in each group to that of the edge with the largest weight in the group. The rationale would be that since they represent the same thing, they should have the same weight. The sequencing algorithm is then run again for the updated spectrum graph.

The point of doing this at the post-processing step is so that spurious paths will not have their scores boosted, since we only restrict ourselves to the good ranked candidate peptides from the first round of sequencing.

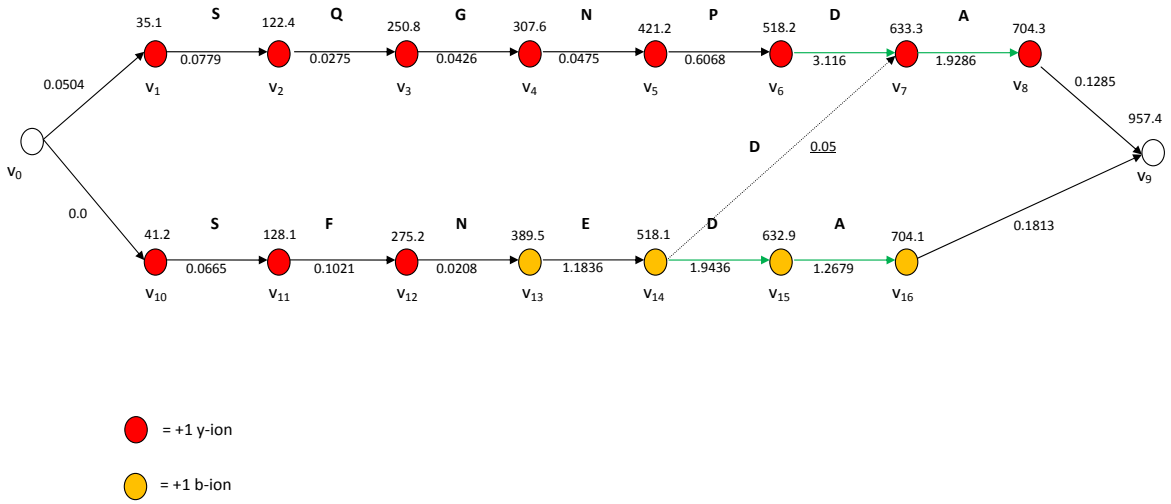


Figure 5.3: Example of Competing Sub-paths. Subpaths (v_6, v_7, v_8, v_9) and $(v_{14}, v_{15}, v_{16}, v_9)$ represent the same fragment DA[253.03], but are of different ion-types. They basically compete with each other since they belong to different paths in the graph. An alternative route to get [41.1905]SFNEDA[253.295] might be taken in order to make use of the higher edge score of 1.9286 compared with 1.2679 (v_7, v_8) vs (v_{15}, v_{16}). This can be done by using the edge from v_{14} to v_7 (dashed line) and continuing from there. However, because of a change from b- to y-ion, the score of edge (v_{14}, v_7) is 0.05 which is lower than the score of edge (v_{14}, v_{15}) and we won't be able to get the benefit of the score of the edge (v_7, v_8).

5.3 DP algorithm for Suffix-K Path-Dependent Longest Path

For every node v_j, let $SP_j(h)$ be the set of all sub-paths leading into and terminating at v_j of length k for some constant k or with length $< k$ if the sub-path starts at node v_0 and hits v_j before reaching a length of k. For each sub-path $p \in SP_j(k)$ where $p = (v_{i-k+1}, .., v_{i-1}, v_i, v_j)$ or $(v_0, ..., v_{i-1}, v_i, v_j)$, let SP^m be the set of maximal mono-chromatic tag in p, and E_p^{mi} be the set of mixed edges in p. This is easily computed by performing a linear scan of the path, adding edges to tags of the same color or adding mixed edges to $E_p^{mi}$when a mono-chromatic edge or mixed edge is encountered. Note that each edge e in p is either a mixed edge or it belongs to exactly one maximal mono-chromatic tag. Let $mctag(e, p)$ be the maximal mono-chromatic tag

that e in p belongs to if e is a mono-chromatic edge. $mctag(e, sp) = []$ otherwise.

At any vertex v_j, the computation of the $MCScore$ of a path $p = (v_0, ..., v_{i-2}, v_{i-1}, v_i, v_j)$ terminating at v_j can be categorized into 4 cases.

1. **Case 1 - edge (v_i, v_j) is mixed.** The score of p is then $MCScore(p - v_j) + w'(v_i, v_j)$. This is because (v_i, v_j) does not contribute to any maximal mono-chromatic tag found in the path and thus does not affect the $MCScore$ of the sub-path $p - v_j$.

2. **Case 2 - edge (v_i, v_j) is mono-chromatic and edge (v_{i-1}, v_i) is mixed.** The score of p is $MCScore(p - v_j) + \lambda(1) * w'(v_i, v_j)$. In this case, (v_{i-1}, v_i) is the start of a new maximal mono-chromatic tag of length 1 and similarly does not affect the $MCScore$ of the sub-path $p - v_j$.

3. **Case 3 - edge $(v_i v_j)$ and $(v_{i-1} v_i)$ are both mono-chromatic.** Note that both can only be mono-chromatic if they are of the same color (same ion-type). This case can be split into 2 sub-cases

   (a) $mctag((v_{i-1}, v_i), p - v_j))$ is of length $\geq k$. In this case, the maximum multiplier has already been considered for the maximal mono-chromatic tag $(v_{i-1} v_i)$ is a part of, thus there is no change to $MCScore(p - v_j)$, and The score of p is $MCScore(p - v_j) + \lambda(k) * w'(v_i, v_j)$.

   (b) $mctag((v_{i-1}, v_i), p - v_j))$ is of length $< k$. In this case, $MCScore(p - v_j)$ has to be updated, since the addition of (v_i, v_j) increases the length of maximal mono-chromatic tag that (v_{i-1}, v_i) is a part of by 1. Only the score of the tag $mctag((v_{i-1}, v_i), p - v_j))$ need be updated, as other maximal mono-chromatic tags are not affected. The updated score of $t = mctag((v_{i-1}, v_i), p - v_j))$ is

$$update\_score(t) = \sum_{\forall e \in t} \lambda(length(t) + 1) * w'(e) - \lambda(length(t)) * w'(e) \tag{5.4}$$

   We minus off the original score of t in equation 5.4 so that the updated $MCScore$

for $p - v_j$ is then

$$MCScore'(p - v_j) = MCScore(p - v_j) + update\_score(t)$$

The score of p is then $MCScore'(p - v_j) + \lambda(length(t) + 1) * w'(v_i, v_j)$

DP Formulation

The above leads to a DP formulation for computing the $SMCScore$ of the optimal sub-path terminating at a node v_j with a suffix $p = (v_{i-k+1}, .., v_{i-1}, v_i, v_j)$ (p does not begin at the start node v_0) as follows

$$opt(v_j, p) = \max_{\substack{\forall p' \in SP_i(k) \, and \\ p - v_j \subset p'}} \begin{cases} opt(v_i,p') + w'(v_i,v_j) & if\ case\ 1 \\ opt(v_i,p') + \lambda(1) * w'(v_i,v_j) & if\ case\ 2 \\ opt(v_i,p') + \lambda(h) * w'(v_i,v_j) & if\ case\ 3a \\ opt(v_i,p') + \lambda(length(mctag((v_{i-1},v_i),p')) + 1) * w'(v_i,v_j) + update\_score(mctag((v_{i-1},v_i),p')) & if\ case\ 3b \end{cases} \tag{5.5}$$

where $p - v_j \subset p'$ means that $p - v_j$ is the $k - 1$ suffix of p'.

If p begins at v_0,

$$opt(v_j, p) = MCScore(p) \tag{5.6}$$

Initialization: for all sub-paths of length $\leq k$ from v_0, compute their $MCScore$.

DP execution: For each node in topologically sorted order, apply the DP formulation until last node is processed.

The optimal path can be obtained by going from the last node backwards, choosing the edge $(v_h v_l)$ from the sub-path $sp = (v_{h-k+1} ... v_h, v_l)$ that optimizes $opt(v_l, p)$ and then moving to v_h and repeating the process until we reach v_0.

We see that for the case of $k = 0$, $MCScore(P) = SScore(P)$. This reduces the problem to finding the longest path in a DAG.

5.3.1 Computational Complexity of DP algorithm

For any vertex v_j, $|SP_j(k)| = (c+20)^k$ since there are at most c+20 number of different edges (20 different amino acids and repeated edges due to error tolerance represented by the constant c) coming into any vertex and we only consider sub-path of length up to k.

For a given sub-path $p = (v_{i-k+1},..,v_{i-1},v_i,v_j)$ in $SP_j(k)$ we go through all sub-path $p' \in SP_i(k)$ where $(v_{i-k+1},...,v_i)$ is the $k-1$ suffix of p', and find the one that maximizes $opt(v_j,p)$. There are at most $(c+20)$ of such sub-paths, since p' differs from p only at its start edge. For each p', checking which of the four cases it falls into takes O(1) time. This is because in the worst case we modify the score of a tag of length at most k, while the other cases is merely a table lookup. Thus we have to process $(c+20)^{k+1}$ sub-paths in order to find $opt(v_j,p)$ for all p. In all we process at most $n*(c+20)^{k+1}$sub-paths, where n is the number of nodes in the spectrum graph. We see that this is equivalent to processing all sub-path of length $k+1$ found in the spectrum graph. Now each unique node and vertex can be found in at most a constant c' number of sub-paths of length $k+1$. The time complexity can then be expressed as $O(c'(n+m))$. The constant c' however can be very big depending on the value of k, and in practice can take a long time to compute when k is large.

Chapter 6

MCPS Parameter Tuning

In this chapter, we describe how we set the parameters in the various steps involved in generating candidate peptides by MCPS. We will first describe the datasets used in our experiments. Next we will describe how the ion-types sets used by MCPS were obtained. After that, we will evaluate the effect of varying different parameters in step 4-8 of the MCPS algorithm. We will then give the final parameter settings for each dataset.

6.1 Datasets

For our experiments, we have used spectra that are annotated with their corresponding peptides – the **GPM-Amethyst** dataset (Craig et al. [12], the **ISB** dataset (Keller et al. [34]) and the **ISB2** dataset (Klimek et al. [37]). These datasets will be split into training and testing subsets (except for ISB which is fully used for testing). The training sets are used to determine the ion-type sets used, the parameter setting, as well as for training the probabilistic model given Liu et al. [38] which will provide the raw weights for MCPS's scoring function. The test sets are used to evaluate sequencing results of MCPS and other algorithms in Chapter 7.

GPM-Amethyst Dataset. The GPM-Amethyst dataset are MS/MS spectra obtained from QSTAR mass spectrometers, from both MALDI and ESI sources. The entire Amethyst dataset consists of a total of 12,558 spectra of difference charges from 1 to 5. Normally, QSTAR datasets are highly accurate and usually it is possible to determine the charge state of the peaks

by examining the isotope peaks. However, the Amethyst dataset that is publicly-available from the GPM web-site are *pre-processed* datasets – each spectrum has between 20-50 peaks (usually high quality peaks). The average number of peaks per spectrum is about 40. Peptides in the GPM dataset are on average 14-15 amino acids long.

We exclude spectra for which the difference between the parent ion mass and the mass of the annotated peptide exceeds a threshold of 3 Da. This is so as to exclude spectra of peptides which contains PTMs (which can modify the peptides mass by a large amount), since MCPS does not handle PTMs. We also exclude spectra having an X-correlation score (Xcorr) < 2.0 (a rule of thumb for "good" quality spectra).

After this filtering, our GPM-Amethyst dataset consists of a total of 2327 spectra – 756, 874, 453, 207, 37 spectra with charges 1,2,3,4 and 5 respectively. Out of this set, we further partition them randomly into a test set and training set. The testing set consists of 1076 spectrum in all - 302, 349, 181, 207 and 37 spectra with charges 1,2,3,4 and 5, respectively. For the training set, we did not use any charge 4 or 5 spectra as they are too small in number to be used for training puposes. Thus for charge 4 and 5 spectra we used the parameters trained for the charge 3 data. The training set consists of 1251 spectrum in all - 454, 525 and 272 spectra of charges 1,2 and 3 respectively. We used a random 40/60 split between the testing and training sets.

To perform sequencing on the GPM-Amethyst dataset, we use error tolerance $\varepsilon = 0.5Da$ for joining vertices in the spectrum graph.

ISB Dataset. The ISB dataset consists of low energy CID ion-trap data generated using an ESI source from a mixture of 18 proteins and consists of 5334 spectra with charge 1,2 and 3. For each multi-charge spectrum, the machine outputs two spectra (one for charge 2 and one for charge 3) because there is not enough resolution to determine the precursor charge. Both spectra are then searched using SEQUEST to find the best matching peptide among them – based on a better SEQUEST XCorr score. Peptides in the ISB dataset are of length 15-16 on average.

We similarly exclude spectra with low XCorr scores (≤ 2.0) or if the the difference between the parent ion mass and the mass of the annotated peptide exceeds a threshold of 3 Da. After

this filtering, our ISB dataset consists of a total of 995 spectra – consisting of 16, 489, 490 spectra with charge 1, 2, and 3, respectively. The ISB dataset have between 200-700 peaks each and an average of about 250 peaks per spectrum. There are, generally, more noise peaks and ISB spectra generally have lower peak specificity.

To sequence the ISB dataset, we use error tolerance $\varepsilon = 0.5Da$, which is the same as that used in many other studies.

ISB2 Dataset. Recently there is a new set of data which we will call the **ISB2** dataset generated from using the same mixture of proteins as ISB data, but using many different mass spectrometers Klimek et al. [37]. We have selected the data that corresponded to the mass spectrometer used to generate ISB data. ISB2 was used as the training set for the weights and ion-types for both itself and ISB data, and consists of a total of 3373 spectra – 535, 2069, 769 spectra of charge 1,2 and 3 respectively, after applying the same filtering criteria as ISB. Similar to the GPM dataset, we use a 40/60 split between the training and testing set. This resulted in a training set of 2024 spectra – 313, 1238, 473 spectra of charge 1,2 and 3 respectively, and a testing set of 1349 spectra – 222, 831, 296 spectra of charge 1,2 and 3 respectively. The entire ISB dataset was used as the testing set.

6.2 Parameter Tuning

6.2.1 Determining Ion-Type Sets

After the peak filtering step in the MCPS algorithm, we will build the extended spectrum S^{α}_{β} from the given experimental spectrum S using an ion-type set $\triangle$, and then the extended spectrum graph $G(S^{\alpha}_{\beta})$ from S^{α}_{β}. In order to obtain the best $\triangle$ to use, we perform a ranking of all the possible ion-types based on a greedy cumulative completeness function which first finds the ion-type giving the highest average completeness value (refer to Section 4.3) across a training dataset. The next ion-type which gives the highest *additional* average completeness values based on the unrecovered portions is listed as the next ion-type. We call this additional average completeness $comp^{+}$. This process goes on until all possible ion-types are considered.

After normalizing the cumulative completeness value against the total completeness attainable using all the ion-types, we pick the set of ion-types that allows us to achieve ≥ 0.85 normalized cumulative completeness value (the 85th percentile of recoverable peptide), or when we hit a hard limit of 15 ion-types.

We use such a scheme, instead of the offset frequency function by Dancik [13], because that function does not take into consideration that some of the ion-types are highly correlated to other ion-types, that is whenever we find a ion of one type, we will also find a similar ion of the other type. In this way, considering both ions does not allow us to recover more of the peptide (although the support for recoverable fragmentation points will be high). Our method tries to ensure maximum recovery of a peptide. We determine a set of ion-types used for each dataset (ISB2 and GPM) and each sub-dataset based on spectrum charge. The ion-types sets used for ISB2 and GPM are as listed in Tables 6.1 and 6.2. The column $cum.comp$ refers to the cumulative completeness value up to that point. We include both the un-normalized ($unor$) and the normalized values (nor). The underlined ion-types are those that are considered in the ion-type set, with the corresponding cumulative completeness values underlined and in bold. The last row shows the last ranking ion-type.

For GPM, we only consider up to charge 3 since the number of charge 4 and 5 data is small, and not representative enough to be used for learning charge 4 and 5 ion-types. Thus for charge 4 and 5 data, we will used the ion-types learned for charge 3 data. Note that the final cumulative completeness values in the tables seem low because we used a very strict error tolerance of $\varepsilon = 0.5Da$ (as opposed to actual sequencing where $\varepsilon = 2.5Da$) when matching recovered amino acids to the canonical peptide. This is to prevent too much over-estimation for amount of the peptide recoverable, and also does not affect the relative ranking of the ion-types.

However we do observe that the un-normalized cumulative completeness values for GPM in general in much lower that for ISB when considering ion-types of the same rank. Moreover, the additional average completeness $comp^+$ for the top ranking ion-types for GPM data is significantly lower than that for ISB. This shows that GPM data do not have a strong preferences for ion-types.

Charge 1 data			Charge 2 data			Charge 3 data		
ion-type	comp$^+$	cum. comp unor / nor	ion-type	comp$^+$	cum. comp unor / nor	ion-type	comp$^+$	cum. comp unor / nor
(+1,y)	0.301	0.301/0.42	(+1,y)	0.338	0.338/0.43	(+1,b)	0.149	0.149/0.18
(+1,b)	0.193	0.494/0.68	(+1,b)	0.145	0.483/0.62	(+2,b)	0.102	0.251/0.31
(+1,y-w)	0.048	0.542/0.75	(+2,y)	0.057	0.540/0.69	(+2,y)	0.085	0.336/0.41
(+1,b-w)	0.041	0.583/0.80	(+1,b-w)	0.031	0.571/0.73	(+1,y)	0.068	0.404/0.49
(+1,y-(w+a))	0.021	0.604/0.83	(+2,y-w)	0.023	0.593/0.76	(+1,b-w)	0.028	0.432/0.53
(+1,b-a)	0.018	**0.623/0.86**	(+1,y-w)	0.018	0.611/0.79	(+2,y-w)	0.026	0.458/0.56
(+1,x-a)	0.016	0.639/0.88	(+2,x-a)	0.014	0.625/0.80	(+2,b-a)	0.023	0.482/0.59
(+1,a-w2)	**0.0004**	**0.726/1.0**	(+2,y-(w+a))	0.012	0.637/0.82	(+2,y-(w+a))	0.020	0.502/0.61
			(+1,b-a)	0.011	0.648/0.83	(+2,x-a)	0.019	0.521/.64
			(+2,x-w2)	0.010	**0.658/0.85**	(+3,y)	0.016	0.537/0.66
			(+1,x-w)	0.010	0.668/0.86	(+1,y-w2)	0.016	0.553/0.68
			(+2,b-w)	**0.0005**	**0.778/1.0**	(+2,a)	0.015	0.568/0.69
						(+3,y-w)	0.013	**0.581/0.71**
						(+2,c)	0.013	0.594/0.73
						(+1,c-w2)	**0.0004**	**0.819/1.0**

Table 6.1: Ion-type ranking for ISB2 data according to spectrum charge type. Column 1 represents the ranking ion-type, column 2 is their *additional* completeness ($comp^+$) value and column 3 is the cumulative completeness value (*cum. comp*), both the un-normalized (*unor*) and normalized ones (*nor*). Ion-types selected until a cumulative completeness value of ≥ 0.85 or a hard limit of 15 ion-types were reached.

Charge1			Charge2			Charge3		
ion-type	comp$^+$	cum. comp unor / nor	ion-type	comp$^+$	cum. comp unor / nor	ion-type	comp$^+$	cum. comp unor / nor
(+1,y)	0.159	0.159/0.35	(+1,y)	0.137	0.137/0.28	(+1,b)	0.061	0.061/0.12
(+1,b)	0.083	0.242/0.53	(+1,b)	0.075	0.212/0.43	(+1,y)	0.025	0.086/0.17
(+1,c-w)	0.031	0.273/0.600	(+2,y)	0.023	0.234/0.48	(+2,b)	0.016	0.103/0.20
(+1,y-a)	0.023	0.296/0.65	(+2,x-w)	0.013	0.247/0.50	(+2,y)	0.013	0.116/0.23
(+1,b-a)	0.018	0.314/0.69	(+2,x-a)	0.012	0.260/0.53	(+2,a-(w+a))	0.012	0.128/0.25
(+1,b-w)	0.015	0.330/0.73	(+1,y-a)	0.012	0.272/0.56	(+3,x-w2)	0.012	0.140/0.28
(+1,x-a)	0.014	0.344/0.76	(+2,y-w)	0.012	0.284/0.58	(+1,y-w)	0.011	0.151/0.30
(+1,y-w)	0.014	0.358/0.79	(+1,b-a)	0.011	0.295/0.60	(+1,c-w)	0.011	0.161/0.32
(+1,y-w2)	0.010	0.368/0.81	(+2,x-(w+a))	0.010	0.305/0.62	(+2,y-a)	0.010	0.172/0.34
(+1,x)	0.010	0.378/0.83	(+2,x)	0.010	0.315/0.64	(+2,x)	0.010	0.182/0.36
(+1,a)	0.009	**0.387/0.85**	(+1,y-w)	0.010	0.325/0.66	(+2,x-a)	0.010	0.192/0.38
(+1,c-w2)	0.009	0.395/0.87	(+1,b-w)	0.010	0.336/0.69	(+3,x-w)	0.010	0.202/0.40
(+1,a-w2)	**0.003**	**0.455/1.0**	(+2,y-(w+a))	0.009	0.345/0.70	(+2,b-a)	0.010	0.212/0.42
			(+2,c)	0.009	0.354/0.72	(+2,y-w)	0.010	0.222/0.44
			(+2,y-a)	0.009	**0.363/0.74**	(+3,y-w2)	0.009	**0.232/0.46**
			(+1,a-w)	0.007	0.370/0.76	(+1,y-a)	0.009	0.240/0.48
			(+1,a-(w+a))	**0.001**	**0.490/1.0**	**(+2,x-w2)**	**0.002**	**0.505/1.0**

Table 6.2: Ion-type ranking for GPM data according to spectrum charge type. The table is similar to that for ISB2 data. However we observe that in general the un-normalized cumulative completeness and comp$^+$values are lower for same rank ion-types in GPM compared to ISB2. This indicates that there are not as strong an ion-type preference in GPM compared to ISB2.

6.2.2 Determining Parameters For Pruning and Bridging Step in MCPS

After building $G(S^{\alpha}_{\beta})$, we need to determine the ion-type sets $\triangle_x$ and tag length l used for the pruning and bridging step in MCPS. To this end, we compute for different $\triangle_x$ and tag length l, the resulting UB on the sensitivity of the best possible candidate peptide that can be obtained from the resultant $G_b(S^{\alpha}_{\beta})$. We call this resultant sensitivity the *UB-Sensitivity* measure. It is defined as follows

UB-Sensitivity. Let ES_b be the extended spectrum induced by $G_b(S^{\alpha}_{\beta})$. That is, it contains the peaks p of the extended spectrum S^{α}_{α} where $mass(p) = mass(v)$ for some vertex v in $G_b(S^{\alpha}_{\beta})$. Let F be the entire PRM ladder for the canonical peptide ρ. Let $F_m = |F \cap ES_b|$ where there is a match between a mass in F and ES_b if they do not differ by more than εDa. We consisting include only the mass from F into the intersection set so that there is no double counting. We then replace the the masses in F_m by the position of the fragmentation point they represent and sort them in ascending order. Next we perform a linear scan to count off the consecutive fragmentation points. The resultant count m represent the number of matching amino acids in ρ.

$$UB\_Sensitivity = \frac{m}{|\rho|} \tag{6.1}$$

In our graph, the x-axis shows the normalized cumulative completeness value of using different $\triangle_x$ based on increasing rank as given in Tables 6.2 and 6.1. We run the experiments for 5 sets of $\triangle_x$ for each charge category, with increasing cumulative completeness ∂. Each plot in the graph represents different tag length l used. $l = 0$ is the case where we do not restrict the minimum length of the mono-chromatic tags of each ion-type in $\triangle_x$. In our experiments we let the tolerance of matching $\varepsilon = 0.5$Da. This will result in UB-Sensitivity values closer to the actual sensitivity of the best possible candidate peptide, since consecutive matching fragmentation points are more likely to be linked with an edge in the graph. This is not the case when ε used is big. We will refer to the **normalized cumulative completeness as simply completeness** in our discussion for the rest of this chapter.

Tuning for ISB2 Data. Figure 6.1 shows the result for For ISB2 data. For +1 and +2 data, we observe that after the big improvement in UB-Sensitivity going from $\triangle_x$ with completeness = 0.4 to that with completeness = 0.6, though the improvement is not as big in the case of +2 data. A bigger $\triangle_x$ does not improve the UB-Sensitivity by much (ave. increase of 0.12 in UB-Sensitivity going from completeness= 0.4 to 0.85 for +1 data). +3 data is different from +2 and +1 data in this respect, in that the plots for the different l values are relatively flat for completeness values of 0.3 to 0.5. There is then a sudden huge increase in the UB-Sensitivity going from completeness = 0.5 to 0.6 and from 0.6 to 0.7. On average there is a 0.15 increase in UB-Sensitivity going from completeness= 0.5 to 0.7. If we look at the Table 6.1, this could be due to the number of extra ion-types required to go from completeness= 0.5 to 0.6 (3 extra ion-types) and from 0.6 to 0.7 (5 extra ion-types). Even so, we can conclude that the extra ion-types at the higher completeness values do help in recovering more amino acids.

From our observation we conclude for +1 and +2 data that the ion-types in $\triangle_x$ with completeness of 0.6 have the largest impact in sequencing. From Table 6.1, these are the top 1-4 ranked ion-types. The rest of the ion-types might not have much impact in recovering more amino acids. However they are still possibly useful as supporting ion-types.

Next we see for +1 data that there is very little drop in the UB-Sensitivity value going from $l = 0$ to $l = 5$ (on average a drop of ~0.03 in UB-Sensitivity). As opposed to +1 data, for +2 and +3 data, we observe a drop which becomes wider at higher completeness values (for +2 data at completeness = 0.85, there is a drop of 0.1 in UB-Sensitivity comparing plot for $l = 0$ to that for $l = 5$).

From this observation we conclude that for +1 data, we will find mono-chromatic tags of such ion-types with length ≥ 5 most of the time. For +2 and +3 data, we conclude that the extra ion-types at the higher completeness values (> 0.6) do not form as many mono-chromatic tags of length ≥ 5 as the better ranked ion-types. This indicates that the extra ion-types can be used in the bridging step rather than the pruning step, since they do not form long mono-chromatic tags.

Based on our analysis, for the MCPS algorithm, we have set $\triangle_x$ to be the ion-types that

gives a completeness of 0.6. We have also set $l = 5$ for +1 data and $l = 3$ for +2 and +3 data.

Tuning for GPM Data. For the GPM data, Figure 6.3 shows the UB-Sensitivity plots for +1 to +5 data. In general comparing the UB-Sensitivity of ISB2 and GPM data, we see that the values for GPM are low even for $l = 0$ and at the highest completeness value (max. of UB-Sensitivity$\approx$ 0.5).

For +1 and +2 data, the UB-Sensitivity does not increase much after completeness = 0.7. This is especially so for +2 data where UB-Sensitivity are almost flat for larger l values. This indicates we practically get no mono-chromatic tags of length$>$ 1, since considering only such tags does not increase the UB-Sensitivity at all even as we use more ion-types (going up the completeness values)There is a drop in UB-Sensitivity of +1 and +2 data going from $l = 1$ to 5 even at low completeness values and this is especially apparent for +2 data (a drop of 50%-70% in UB-Sensitivity between $l = 0$ and $l = 5$).

For +3 data, the UB-Sensitivity plateaus at completeness of 0.2 for $l \geq 2$, while it continues rising at higher completeness values for $l = 0$ and 1. Both + 4 and +5 data have the same pattern, but with much lower UB-Sensitivity values in general. The UB-Sensitivity caps at ~0.3 for charge 3 and above data.

In general we conclude that for all GPM data, ion-types regardless of rank do not have a strong pattern in generating mono-chromatic tags of $l > 1$ or 2. Moreover, instead of the problem of noise, the major obstacle for sequencing multi-charge (≥ 3) GPM data is the fact that a lot of information is missing in the experimental spectrum for peaks corresponding to ion-types in $\triangle$, and we are unable even in the best case to get very good results. This is also confirmed by the low completeness of 0.24 for the top 15 ion-type for charge 3 GPM in Table 6.2 as stated in Section 6.2.1. In order to recover most of the peptide, most of the ion-types and not just the top 15 will have to be considered.

Based on our analysis, for the MCPS algorithm, we do not restrict $\triangle_x$ but instead use the whole of $\triangle$ and skip both the pruning and bridging step when dealing with GPM data. Since the number of peaks in the spectrum for GPM data is small to begin with (< 50 peaks), the spectrum graph generated is of a reasonable size even though pruning is not done.

Analysis of ISB Data. Even though we did not use ISB data for our parameter tuning, we show the UB-Sensitivity plots as a comparison with ISB2 data. We see from the Figure 6.2 that they have the same pattern as that for ISB2 data. Of note is that for charge 1 data, the plots for $l = 1$ to 4 are exactly the same, and there is a drop in UB-Sensitivity only when going to $l = 5$.

Conclusions for Pruning and Bridging Step. For ISB2 the the better ranked ion-types (rank 1 to 4) are the ones which help in recovering the largest amount of the canonical peptide and are usually involved in mono-chromatic tags of a significant length (≥ 3).

For GPM data, we conclude that there are not many mono-chromatic tags of length> 1. Moreover, instead of the problem of noise, the major obstacle for sequencing multi-charge (≥ 3) GPM data is the fact that a lot of information is missing in the experimental spectrum for peaks corresponding to ion-types in $\triangle$, and we are unable even in the best case to get very good results.

For ISB/ISB2 data, we set $\triangle_x$ to be the ion-types that gives a completeness of 0.6. We have also set $l = 5$ for $+1$ data and $l = 3$ for $+2$ and $+3$ data.

For GPM data, we do not restrict $\triangle_x$ but instead use the whole of $\triangle$ and skip both the pruning and bridging step.

6.2.2.1 Comparing Size of Extended Spectrum Graph $G(S^{\alpha}_{\beta})$ and Final Spectrum Graph $G_b(S^{\alpha}_{\beta})$

After the bridging step, we build the final spectrum graph $G_b(S^{\alpha}_{\beta})$. We can compare the reduction in size of $G_b(S^{\alpha}_{\beta})$ after step 4 to the extended spectrum graph $G(S^{\alpha}_{\beta})$. We only tabulate the results for ISB and ISB2 data since GPM skips the pruning and bridging step and builds $G(S^{\alpha}_{\beta})$ instead of $G_b(S^{\alpha}_{\beta})$. In the tables we tabulate

1. *# nodes* - average number of nodes
2. *# edges* - average number of edges
3. *connectivity* - Average number of out-going edges per node

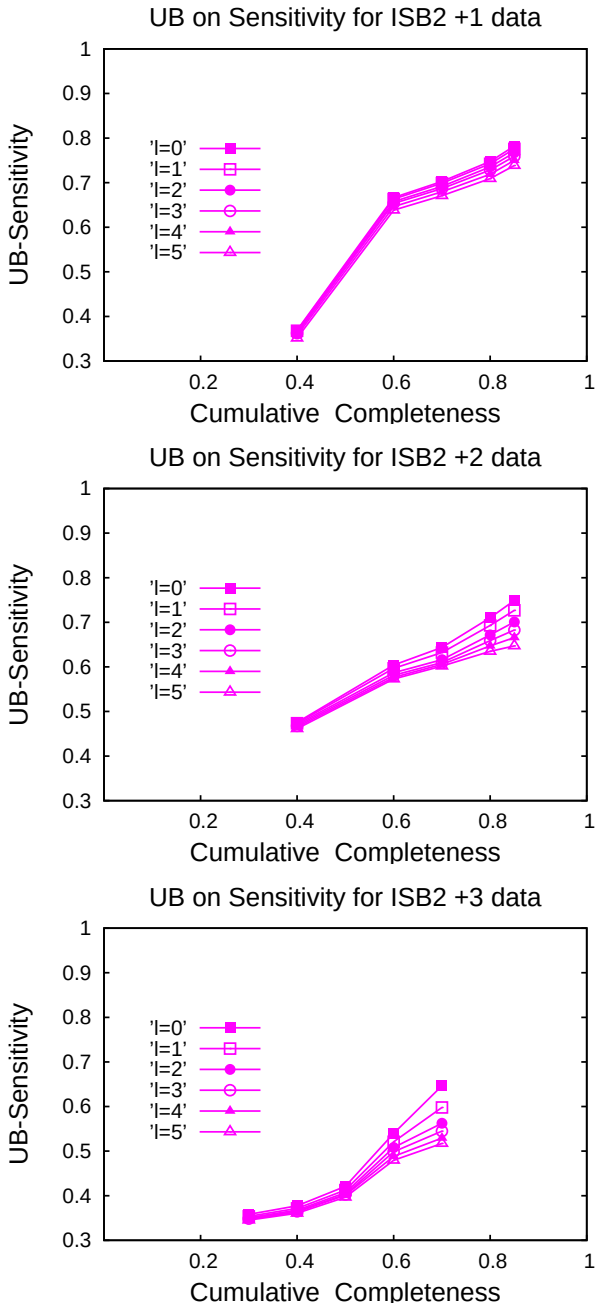


Figure 6.1: UB-Sensitivity for ISB2 Data. Graphs for +1,+2 and +3 data is shown. For each graph, plots are made for $l = 0, 1, 2, 3, 4, 5, 6$. The x-axis shows the normalized cumulative completeness value of using different $\triangle_x$ based on increasing rank as given in Tables 6.2 and 6.1. The y-axis shows the corresponding UB-Sensitivity value for the given l value and completeness value. For +1 & +2 data, greatest increase in UB-Sensitivity is going from completeness of 0.4 to 0.6. For +3 data, plots for all l values are relatively flat for completeness = 0.3 to 0.5, with a sudden huge increase in UB-Sensitivity for completeness = 0.5 to 0.7. For +1 data there is very little drop in UB-Sensitivity going from $l = 0$ to 5 at each of the completeness values plotted, while for +2 and +3, the drop in UB-Sensitivity going from $l = 0$ to 5 becomes more apparent at higher completeness values.

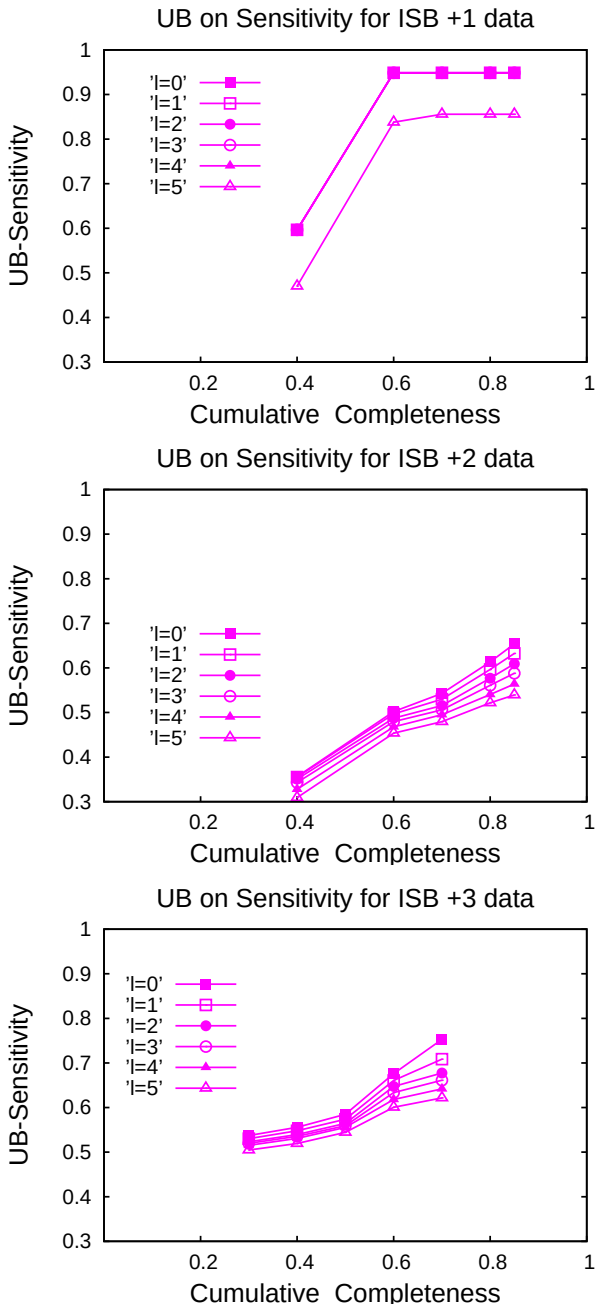


Figure 6.2: UB-Sensitivity for ISB Data. For +1 data, plots for $l = 0$ to 4 are exactly the same with a drop in UB-Sensitivity only for $l = 5$. +2 and +3 data have the same pattern as ISB2 +2 and +3 data.

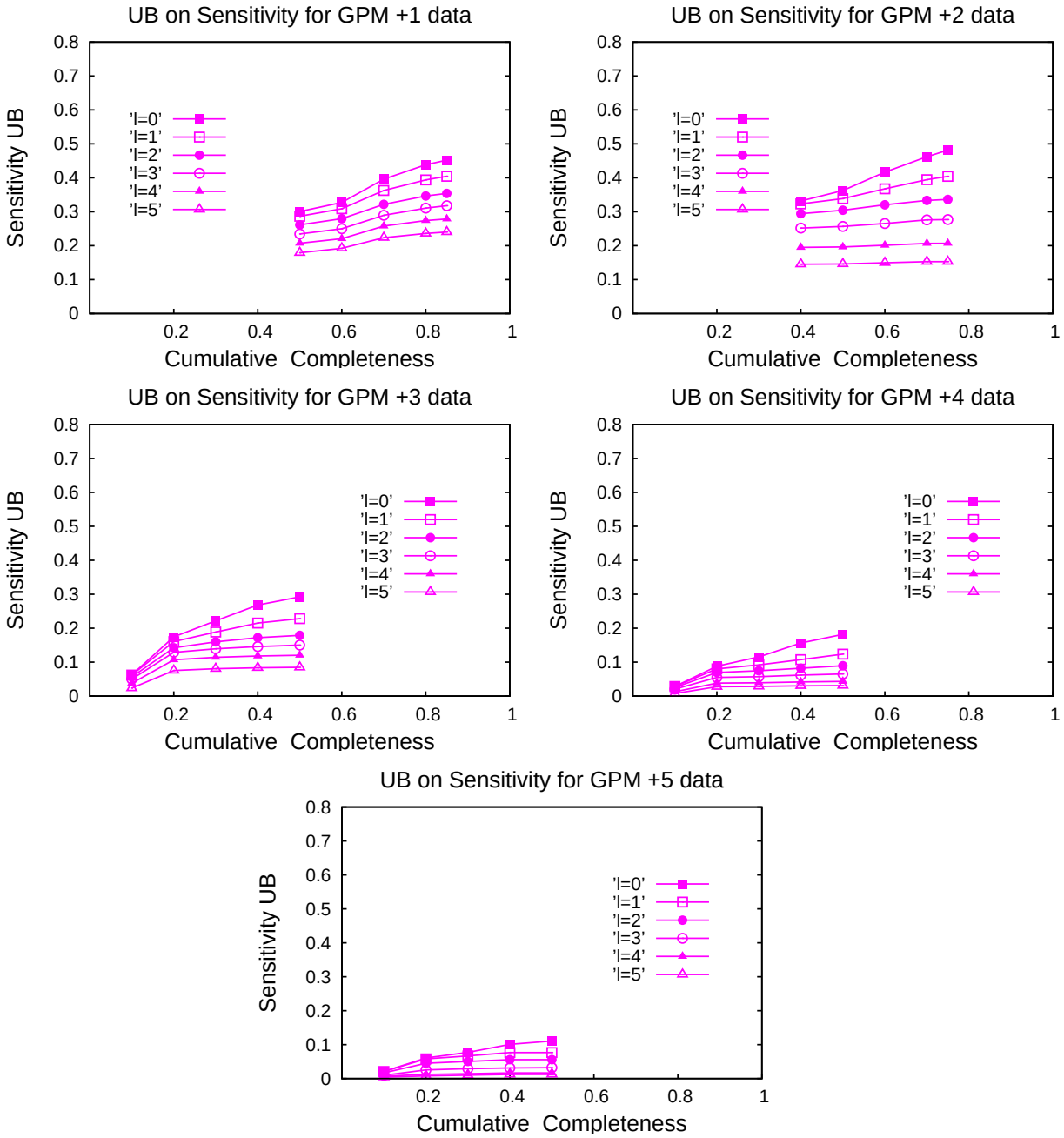


Figure 6.3: UB-Sensitivity for GPM Data. For +1 data, UB-Sensitivity does not increase much after completeness = 0.7. (average increase in UB-Sensitivity of 0.035 from completeness = 0.7 to completeness = 0.85). There is also a noticeable drop in UB-Sensitivity going from $l = 0$ to $l = 5$, especially at the higher completeness values. For +2 data, the plots are relatively flat for $l = 2$ to 5. As we go from $l = 2$ to 5 there is a step decrease in the UB-Sensitivity value of ~0.05. For +3,+4 and +5 data, UB-Sensitivity value plateau at completeness = 0.2 for $l > 2$ (capped at UB-Sensitivity = 0.3 for +3 data, that is 30% of the peptide can be sequenced at most).

4. *UB-Sensitivity*

of $G(S^{\alpha}_{\beta})$ and $G_b(S^{\alpha}_{\beta})$, and show in a separate row, the % reduction in each of the measures listed above when comparing $G_b(S^{\alpha}_{\beta})$ to $G(S^{\alpha}_{\beta})$.

ISB data. Table 6.3 shows the results for ISB data. We see a huge reduction of $> 60\%$ for +1,+2 and +3 data in the size of the graph (# of nodes, # of edges, connectivity) when going from $G(S^{\alpha}_{\beta})$ to $G_b(S^{\alpha}_{\beta})$. There is however only a small drop in UB-Sensitivity of (7%~18%).

More specifically, we see that there is a 66%-76% reduction in the average number of nodes in the extended spectrum graph $G_b(S^{\alpha}_{\beta})$ when compared to $G(S^{\alpha}_{\beta})$. This reduction increases as we go from +1 to +3 data. For the edges there is a reduction on average of 87%-92% when going from $G(S^{\alpha}_{\beta})$ to $G_b(S^{\alpha}_{\beta})$. This reduces the number of edges from the thousands to the hundreds. For the connectivity, this value goes down from around 11-12 edges per node to about 4 edges per node.

However we see that the drop in the UB-Sensitivity due to the removal of nodes and edges in $G_b(S^{\alpha}_{\beta})$ is capped at 18%, with the +1 data only suffering a loss of 7%. This allows for potentially more than 75% of the canonical peptide to be recovered with a good sequencing algorithm, except for the +2 data (65%) where the original UB-Sensitivity is not too high to begin with (79%).

Note that the huge drop in the connectivity ($> 60\%$) greatly reduces the number of possible paths found in $G_b(S^{\alpha}_{\beta})$, as there are $O(h^c)$ where h is the average number of nodes in a path, and c the average connectivity. While reducing the nodes and edges affect h, reducing the connectivity affects c which makes a bigger impact to the reduction in the number of possible paths.

ISB2 data. From Table 6.4, we see the same pattern as for ISB data, where there is a huge reduction in the average number of nodes, edges and the connectivity. Even though the UB-Sensitivity value drops by 23% for +3 data, the absolute value still allows potentially 70% of the peptide to be recovered with a good sequencing algorithm.

Conclusions on Comparing Size of $G(S^{\alpha}_{\beta})$ and $G_b(S^{\alpha}_{\beta})$. We can conclude that $G_b(S^{\alpha}_{\beta})$ is a good spectrum graph for our MCPS algorithm to work with, as there is (i) a great reduction in

the size of the problem and the amount of noise to deal with, and (ii) there is enough information (good UB-Sensitivity) for us to recover a good portion of the peptide. With a good sequencing algorithm, 65% or more of the peptide can be sequenced. If the denovo sequencing results were used as tags for database search, 65% matching amino acids is more than enough to get a unique hit in the database (usually a tag of 3 or more matching amino acids will be sufficient to get a unique hit 80% of the time [22]).

6.2.3 Sequencing Using Different Suffix-k

After setting the parameters for the pruning and bridging step. We now determine the parameters for the DP algorithm used in generating the candidate peptide in step 7 of MCPS. The parameter to be set in this step is k, the length of the suffix to be considered in $MCScore$. We have tried $k = 0, 1, 2, 3$. Bigger values of k did not finish in a reasonable amount of time and were not considered. Note that the case of $k = 0$ reduces $MCScore$ to $SScore$ (with the added constraint of not considering supporting edges) and the problem to finding the longest path in a DAG. For λ the multiplier, we have determined empirically that $\lambda(1) = 1$, $\lambda(2) = 2$ and $\lambda(3) = 4$ gives the best result. We tabulated the sensitivity values of the top result generated by MCPS. The k value that gave the best sensitivity overall is selected for each dataset, and is indicated by a * next to it in the table. We also bold and underlined the best sensitivity result for each of the charge categories.

Tuning for GPM Data. From Table 6.5, we see that $k = 1$ gives the best result overall. Even though $k = 3$ gave a very slight improvement for +3 data over $k = 2$, there is a corresponding greater drop in sensitivity for charge 1 and 2 data. This is consistent with our UB-Sensitivity plot in Section 6.2.2, where we note that there are rarely any mono-chromatic tag lengths of > 1 for any of the considered ion-types. In general however we note that the sensitivity is low for GPM data.

Tuning for ISB2 Data. We see from Table 6.6 that $k = 3$ gives the best result. There is a big improvement going from $k = 0$ to $k = 1$ but only a minor improvement going from $k = 1$ to $k = 3$.

	Spec. Graph	# nodes	# edges	Connectivity	UB-Sensitivity
Charge 1	$G(S^{\alpha}_{\beta})$	452	4982	10.9	0.99
	$G_b(S^{\alpha}_{\beta})$	153	636	4.1	0.92
	% reduction	66%	87%	62%	7%
Charge 2	$G(S^{\alpha}_{\beta})$	673	7287	10.8	0.79
	$G_b(S^{\alpha}_{\beta})$	197	768	3.9	0.65
	% reduction	71%	89%	64%	18%
Charge 3	$G(S^{\alpha}_{\beta})$	867	10157	11.7	0.93
	$G_b(S^{\alpha}_{\beta})$	209	824	3.9	0.76
	% reduction	76%	92%	67%	18%

Table 6.3: Comparing $G_b(S^{\alpha}_{\beta})$ and $G(S^{\alpha}_{\beta})$ for ISB Data. The table shows the #nodes, # edges, connectivitiy and Ub-Sensitivity for $G(S^{\alpha}_{\beta})$ and $G_b(S^{\alpha}_{\beta})$ generated for +1,+2 and +3 data. In general $G_b(S^{\alpha}_{\beta})$ shows> 60% reduction from the size of $G(S^{\alpha}_{\beta})$ (# edge, # nodes and connectivity), but with only a small drop in UB-Sensitivity of (7%~18%). UB-Sensitivity of +2 data drops to 0.65, but was not big to begin with (0.79).

	Spec. Graph	# nodes	# edges	Connectivity	UB-Sensitivity
Charge 1	$G(S^{\alpha}_{\beta})$	585	5211	8.9	0.93
	$G_b(S^{\alpha}_{\beta})$	192	648	3.4	0.78
	% reduction	67%	88%	62%	16%
Charge 2	$G(S^{\alpha}_{\beta})$	845	10003	11.9	0.88
	$G_b(S^{\alpha}_{\beta})$	248	1028	4.1	0.74
	% reduction	71%	90%	66%	16%
Charge 3	$G(S^{\alpha}_{\beta})$	1215	14812	12.2	0.90
	$G_b(S^{\alpha}_{\beta})$	298	1206	4.0	0.69
	% reduction	75%	92%	67%	23%

Table 6.4: Comparing $G_b(S^{\alpha}_{\beta})$ and $G(S^{\alpha}_{\beta})$ for ISB2 Data. we see the same pattern in reduction of the graph size going from $G(S^{\alpha}_{\beta})$ to $G_b(S^{\alpha}_{\beta})$. For +3 data, even though UB-Sensitivity drops by 23%, the absolute value is maintained at 0.69, that is potentially we can still recover ~70% of the peptide on average.

Analysis of ISB Data. Even though we did not use ISB data for our parameter tuning, we ran the experiment for ISB as a comparison with ISB2 data the From Table 6.7, yet again there is a clear improvement going from $k = 0$ to $k = 1$ with minor improvement after that. $k = 2$ and $k = 3$ gives the same result for +1 data, with $k = 2$ slightly better for +3 and $k = 3$ slightly better for +2.

Conclusions on using different suffix-k. In general, we note that increasing k beyond 1 does not result in much improvement (there is slight improvement for ISB and ISB2 data). For the GPM data, this is due to the fact that the information present lack peaks which map to long mono-chromatic tags of any ion-type.

For the ISB and ISB2 data, the reason could be the exact opposite. That is, the abundant and high ranking ion-types dominate the spectrum graph, and even fictitious paths contains long mono-chromatic tags. Trying to differentiate between real and false candidate peptides then becomes more than just considering the length of mono-chromatic tags.

Based on our analysis, we have set $k = 1$ for GPM data, and $k = 3$ for ISB and ISB2 data in order to maximize peptide recovery.

6.2.4 The Effect of Post-Processing on MCPS Results

After sequencing, we apply the last step the post-processing step to the result. We compare the results of MCPS before performing post-processing and the results after post-processing. We first compare the sensitivity and specificity results of the top ranked result, termed *Top-1*. The change in sensitivity and specificity values after post-processing is indicated in brackets () after their values in the tables.

We then compare among the top 100 results, termed *Top-100*, the *average ranking* of the 1st candidate peptide that gives a correctly predicted tag of length ≥ 3. We refer to such a candidate peptide as a **pep-3** candidate, and these are good quality candidates. The reason we choose pep-3 instead of the best candidate out of the Top-100 is (i) even though the ranking of the best candidate peptide might be high, the quality of the best prediction could be poor. A correctly predicted tag of length ≥ 3 is good enough for use in database search to generate

GPM	Sensitivity		
	Charge 1	Charge 2	Charge 3
$k = 0$	0.087	0.092	0.035
$*k = 1$	**0.114**	**0.156**	0.057
$k = 2$	0.108	0.154	0.057
$k = 3$	0.096	0.145	**0.058**

Table 6.5: GPM Sensitivity Results For Different k Values. k=1 gives the best result. There is a jump in the sensitivity results going from $k = 0$ to $k = 1$, but there is not much change going from $k = 1$ to $k = 2$. There is a drop in sensitivity for charge 1 and 2 for $k = 3$ compared to $k = 1$ and 2.

ISB2	Sensitivity		
	Charge 1	Charge 2	Charge 3
$k = 0$	0.186	0.157	0.075
$k = 1$	0.259	0.296	0.136
$k = 2$	0.254	0.305	0.143
$*k = 3$	**0.267**	**0.310**	**0.145**

Table 6.6: ISB2 Sensitivity Results For Different k Values. Big improvement using $k = 1$ compared to $k = 0$. there is consistent but very small improvement going from $k = 1$ to $k = 3$.

ISB	Sensitivity		
	Charge 1	Charge 2	Charge 3
$k = 0$	0.227	0.110	0.112
$k = 1$	0.433	0.237	0.206
$k = 2$	**0.459**	0.236	**0.216**
$*k = 3$	**0.459**	**0.241**	0.208

Table 6.7: ISB Sensitivity Results For Different k Values. Pattern of improvement is not as clear as ISB2 data after $k = 1$.

unique hits, and thus ensures a certain quality to the candidate peptides thus ranked. (ii) Even when there could be better candidate peptides that are lower in rank, these can be ignored if we can find a good enough candidate higher in the rank.

If there are no candidate peptides with a correct tag of length ≥ 3, the ranking is set to 101 for that case. A higher average ranking will indicate that post-processing helped to improve the sequencing result in ranking better candidates higher even though the sensitivity and specificity might remain the same when considering only the top ranked result.

De novo sequencing using MCPS in the rest of the thesis after this section will henceforth refer to that after post-processing.

6.2.4.1 Sensitivity and Specificity Results

We compare the results for Sensitivity and Specificity before and after post-processing. We label this as **Sensitivity/Specificity** (before post-processing) and **Sensitivity$^+$/Specificity$^+$** (after post-processing) respectively. We compare using Top-1, as well as the best candidate in Top-100.

ISB Data. For the Top-1 result, we see from Table 6.8 that sensitivity for +1 data dropped by quite a bit (0.083). This would seem strange since the sensitivity for ISB2 +1 data actually went up (Table 6.9). When we look at the Top-100 results, there is actually a slight increase in the sensitivity value of the best candidate peptide after post-processing by 0.007. This indicates that post-processing was able to generate new candidate peptides which gave a better match with the canonical peptide, but the scoring was not sensitive enough to make them the top ranking peptide. On the other hand the top ranking candidate peptide before post-processing suffered in their score after post-processing and was replaced by a worse candidate, resulting in the drop in sensitivity. However, since there are only 16 +1 ISB data, it is hard to conclude if this is indeed the case. The specificity values are similar to sensitivity values before and after post-processing.

The sensitivity of charge 2 data after post-processing remains the same, while that for charge 3 data shows very slight improvement both for Top-1 and Top-100 results. Specificity values

remains about the same after post-processing. Yet again we see that specificity values are similar to sensitivity values for charge 2 and 3 data.

ISB2 Data. Table 6.9 shows the sensitivity and specificity results for ISB2 data before and after post-processing. There is no change sequencing results for charge 2 data for both Top-1 and Top-100 results. For charge 1 and 3 data, there are improvements in both the sensitivity and specificity for Top-1 results (> 0.025 for charge 1 and > 0.01 for charge 3). Both charge 1 and 3 data shows improvement in sensitivity and specificity even for Top-100 results. This indicates clearly that mono-chromatic sub-paths of different ion-types representing the same peptide fragment can sometimes compete with each other, affecting the sequencing result adversely. Boosting up the scores of such mono-chromatic edges at the post-processing step helps in improving the results, while not unnecessarily improving the scores of spurious paths if this were to be done before the 1st round of sequencing.

GPM Data. Table 6.10 shows the sensitivity and specificity of GPM data before and after post-processing. In general, sensitivity and specificity values are low for GPM data both before and after post-processing (< 0.2 for sensitivity and specificity).

The sensitivity results after post-processing shows a huge improvement relative to the sensitivity before post-processing for charge 1 and 2 data (improvement in Top-1 result of 0.029 for both). There is also noticeable improvement in sensitivity at the Top-100 results for charge 1 and 2 (0.026 and 0.013 respectively). There is no noticeable change in sensitivity value for both Top-1 and Top-100 results for charge 3,4,5 data.

The specificity values for charge 1,2 and 3 data follows the same pattern as their sensitivity values, and are in fact similar in value to their sensitivity values. An interesting pattern to note for the specificity value of charge 4 and 5 data is that the Top-100 specificity values are 2-4 times bigger than the Top-100 sensitivity values. This indicates that the candidate sequences generated are very short in length (in terms of the amino acid content) compared to the canonical peptides. Values in Table 7.3 for charge 4 and 5 data corroborates this claim. This backups the claim on GPM data in 6.2.2, that generally multi-charge (≥ 3) GPM data is missing a lot of information in the experimental spectrum for peaks corresponding to ion-types in our ion-type

set $\triangle$ used.

6.2.4.2 Average Ranking of First Matching Candidate Peptide

We label the ranking of pep-3 candidates before post-processing **Ranking** and that after **Ranking$^+$.**

ISB and ISB2 Data. Table 6.11 and 6.12 shows the ranking before and after post-processing for ISB and ISB2 data respectively. Overall we note that the rank of pep-3 is closer to the top 1 result than the top 100 result before and after post-processing. Next we see that there is noticeable improvement in the ranking of pep-3 for +1 and +3 data relative to their original ranking. Both ISB and ISB2 +3 data improved their pep-3 ranking by 5 places. Thus even though there was not much change in sensitivity and specificity for ISB +3 data (refer to Section 6.2.4.1), the ability to get pep-3 candidates have improved. This signifies that post-processing has improved the ranking of good quality candidate peptides of multi-charge data in general.

The improvement in the pep-3 candidate ranking for ISB +1 data is contrary to the drop in ranking of the best candidate due to the drop in the average sensitivity (refer to previous section). This indicates that post-processing helps improve the ranking of good quality candidates in general. Ranking of +2 data for both ISB and ISB2 does not improve much, but did not go down either.

The above observations show that mono-chromatic sub-paths of different ion-types representing the same peptide fragment does compete with each other, affecting the sequencing result adversely. Boosting up the scores of such mono-chromatic edges at the post-processing step helps in improving the results (especially for pep-3 candidates), while not improving the scores of spurious paths if this were to be done before the 1st round of sequencing.

GPM Data. Overall we see from Table 6.13 that the rankings of pep-3 candidates for GPM data after post-processing does not improve noticeably relative to their original rankings, which are quite bad (most > 60 except for +2 data). However ranking did not go down either. The bad average ranking is due to the fact that there are actually very few pep-3 candidates (correctly predicted tag length ≥ 3) generated by MCPS. This is corroborated by results in Section 7.2.2.

ISB Data	Sensitivity	Sensitivity$^+$	Specificity	Specificity$^+$
Charge 1 (Top-1)	0.459	0.376 (*-0.083*)	0.447	0.361 (*-0.086*)
Charge 1 (Top-100)	0.790	0.797 (*+0.007*)	0.788	0.804 (*+0.016*)
Charge 2 (Top-1)	0.241	0.241 (*0.0*)	0.262	0.261 (*-0.001*)
Charge2 (Top-100)	0.396	0.394 (*-0.002*)	0.468	0.466 (*-0.002*)
Charge 3 (Top-1)	0.208	0.216 (*+0.008*)	0.225	0.231 (*+0.006*)
Charge 3 (Top-100)	0.406	0.410 (*+0.004*)	0.462	0.463 (*+0.001*)

Table 6.8: Comparing Before and After Post-Processing for ISB Result. +1 sensitivity drop by quite a bit (0.083). Sensitivity and Specificity for +2 and +3 data remains about the same.

ISB2 Data	Sensitivity	Sensitivity$^+$	Specificity	Specificity$^+$
Charge 1 (Top-1)	0.246	0.273 (*+0.027*)	0.225	0.253 (*+0.028*)
Charge 1 (Top-100)	0.447	0.477 (*+0.03*)	0.434	0.468 (*+0.034*)
Charge 2 (Top-1)	0.310	0.312 (*+0.002*)	0.337	0.337 (*0.0*)
Charge2 (Top-100)	0.469	0.466 (*-0.003*)	0.536	0.528 (*-0.008*)
Charge 3 (Top-1)	0.145	0.157 (*+0.012*)	0.159	0.170 (*+0.011*)
Charge 3 (Top-100)	0.249	0.270 (*+0.021*)	0.285	0.305 (*+0.020*)

Table 6.9: Comparing Before and After Post-Processing for Top-1 ISB2 Result. Clear improvement in the +1 and +3 sequencing results (improvement of Top-1 sensitivity result of 0.027 and 0.012 respectively and Top-100 sensitivity result of 0.03 and 0.021). No change to +2 sequencing results.

GPM Data	Sensitivity	Sensitivity$^+$	Specificity	Specificity$^+$
Charge 1 (Top-1)	0.094	0.123 (*+0.029*)	0.102	0.130 (*+0.028*)
Charge 1 (Top-100)	0.243	0.269 (*+0.026*)	0.272	0.299 (*+0.027*)
Charge 2 (Top-1)	0.151	0.180 (*+0.029*)	0.166	0.196 (*+0.030*)
Charge 2 (Top-100)	0.268	0.281 (*+0.013*)	0.309	0.322 (*+0.013*)
Charge 3 (Top-1)	0.049	0.045 (*-0.004*)	0.061	0.053 (*-0.008*)
Charge 3 (Top-100)	0.107	0.107 (*0.0*)	0.160	0.163 (*-0.003*)
Charge 4 (Top-1)	0.023	0.027 (*+0.004*)	0.037	0.052 (*+0.015*)
Charge 4 (Top-100)	0.057	0.057 (*0.0*)	0.130	0.135 (*+0.005*)
Charge 5 (Top-1)	0.004	0.005 (*+0.001*)	0.004	0.013 (*+0.009*)
Charge 5 (Top-100)	0.034	0.034 (*0.0*)	0.138	0.140 (*+0.002*)

Table 6.10: Comparing Before and After Post-Processing for Top-1 GPM Result. Big improvement to the Top-1 sensitivity values for +1 and +2 data (improvement of 0.029 for both). Slight improvement in sensitivity for +4 and +5 data, but the actual values are bad (recovers 2.7% of +4 data and virtually none of the +5 data for Top-1 results). Specificity results are around 4 times better than sensitivity results for +4 and +5 data indicating that most of the candidate peptides are very short in amino acid length compared to the canonical peptides.

ISB	Ranking	Ranking$^+$
Charge 1	17	14
Charge 2	34	33
Charge 3	25	20

Table 6.11: Ranking of Pep-3 Candidate for ISB Data. Rank of the pep-3 candidate is closer to the Top-1 rather than the Top-100 result. Noticeable drop in ranking of +1 and +3 data after post-process.

ISB2	Ranking	Ranking$^+$
Charge 1	33	29
Charge 2	19	19
Charge 3	41	36

Table 6.12: Ranking of Pep-3 Candidate for ISB2 Data. Similar to ISB data, rank of the pep-3 candidate is closer to the Top-1 rather than the Top-100 result. There are also improvements to ranking after post-processing for +1 and +3 data.

GPM	Ranking	Ranking$^+$
Charge 1	55	53
Charge 2	49	48
Charge 3	72	72
Charge 4	72	70
Charge 5	61	60

Table 6.13: Ranking of Pep-3 Candidate for GPM Data. Very slight to no improvement in ranking for all GPM data. In general ranking of pep-3 candidate is bad being closer to Top-100 then Top-1.

6.2.4.3 Conclusions on sequencing results before and after post-processing

Comparing MCPS sequencing results before and after post-processing, we conclude that post-processing did not degrade the ranking of good quality candidate peptides for all datasets in general.

Post-processing improved the ranking of good quality candidates (pep-3) for multi-charge data (+3) ISB and ISB2 data, even though the sensitivity of the best candidate did not improve much. Ranking of good quality candidates improved noticeably for +1 ISB data even though sensitivity of best candidate dropped by a lot.

Even after post-processing, the sensitivity and ranking of good quality candidates for GPM data is bad. Most of the good candidate peptides do not have correctly predicted tags of length ≥ 3.

6.2.5 Conclusion and Parameter Settings Used

From our experiments, we have shown that GPM and ISB/ISB2 data have vastly different characteristics which prompts us to use two sets of parameter settings for them.

We have used the following values for the different parameters for ISB/ISB2 and GPM data.

ISB/ISB2 data. For +1 data, $\triangle_x$ is set to be the ion-types that gives a completeness of 0.6, $l = 5$ and $k = 3$.

For +2 and +3 data, $\triangle_x$ is set to be the ion-types that gives a completeness of 0.6, $l = 3$ and $k = 3$.

GPM data. For GPM data, we do not restrict $\triangle_x$, but instead uses the whole of $\triangle$ and skip both the pruning and bridging step. We use $k = 1$ for the lookback length of $MCScore$.

Chapter 7

Comparing MCPS with Other Algorithms

In this chapter we compare and analyze the results of MCPS against that of Lutefisk, PEAKS and PepNovo based on various evaluation criteria set out in Section 7.1. Next we measure how much multi-charge ions, namely +3 ions affect the results of MCPS. The datasets we have used for sequencing are the ISB, ISB2 and GPM test sets defined in Chapter 6. Pevtsov et al. [50] have also performed a comparison of several de novo sequencing algorithms based on QSTAR and LCQ/ESI data, and is a good reference for further comparison among de novo sequencing algorithms.

7.1 Evaluation Criteria

To evaluate the performance of MCPS, we need metrics that measures how much of the canonical peptide ρ is recovered by a given candidate peptide P, and also metrics that measure the "*goodness*" of the recovered portions. The following metrics are used to evaluate our algorithm.

Sensitivity. *Sensitivity* measures how much of the peptide ρ is matched in the candidate peptide P. It indicates the quality of the candidate sequence with respect to the canonical peptide sequence and a high sensitivity means that the algorithm recovers a large portion of ρ. It is defined as

$$sensitiviy = \frac{\#correct}{|\rho|}$$

where #correct is the "number of correctly sequenced amino acids". An amino acid is *correctly sequenced* only if it does not differ from its position in the canonical peptide ρ by more than ε. In our experiments, we have used $\varepsilon = 2.5Da$, since the precursor parent mass given in the mass spectrum often differs by up to this amount from the canonical peptide mass (refer to [21]).

Specificity. *Specificity* measures how much "noise" is present in the candidate peptide P. Even though P can have a high sensitivity, it could also have predicted a lot of mismatching amino acids too. This specificity measures the accuracy of our algorithm. It is defined as

$$specificity = \frac{\#correct}{|P|} \tag{7.1}$$

Predictions with Correct Tags of Length $\geq x$. Even though Sensitivity measures the amount of recoverable peptide, it does not measure the quality of the recovered portions. For example the canonical peptide **A**AG**G**DD**FF**QT**R** can have the amino acids in bold and underlined matched in a candidate peptide. This amount to about ~45% recovery, and is considered quite good. However the recovered portions are very fragmented and do not give us a good idea of the actual canonical peptide. AAG**GDDF**FQTR even though recovering less of the peptide (4 amino acid compared to 5) is better, because the matching portion is a long tag which give a good idea of ρ, and when used as a tag in database search (hybrid approach) can result in a unique hit if the canonical peptide is found in the database. In fact Frank et al. [22] showed that peptide tags which are correctly predicted with a tag of ≥ 3 amino acids result in an 80% hit rate when used in database search. This *quality of matching* is captured by this metric, which computes the ratio of instances, where candidate peptides matches the canonical peptide with a tag of at least length x. The higher the ratio for larger x, the better the quality of the predictions.

7.2 Comparing Results of MCPS with other Algorithms

Comparing MCPS Top-1 results against Lutefisk, PEAKS and PepNovo, we find that MCPS performs well for multi-charge spectra relative to the others. It does the best for +3 ISB data and second best for ISB2 data (PepNovo is best). It does the best for +4 GPM data (on a par with Lutefisk) and second best for +3 and +5 GPM data (Lutefisk is best). However even though MCPS Top-1 does well relative to the others for GPM data, the quality of predictions are very low, with very few peptides having correctly predicted tags of length ≥ 3.

MCPS can generate peptides with correctly predicted tags of length≥ 3 more than 50% of the time for +3 ISB data. Moreover, it generates the most number of unique predictions (not predicted by the other algorithms) with correct tags of length ≥ 3 other than PepNovo. This shows that MCPS is good for generating peptide tags for database search of +3 ESI/low energy CID-based spectra either on its own or together with PepNovo.

MCPS Top-100 results are better than the other algorithms for +2 and +3 ISB and ISB2 data. It is also on a par with PepNovo for +1 ISB2 data. It performs the best for all GPM data, but quality of predictions for +4 and +5 data are still very low ($< 10\%$ peptide can be recovered). This indicates that further enchancements to MCPS algorithm need only look at a narrow band of solutions, namely the top 100.

7.2.1 Sensitivity and Specificity Results

We plot the sensitivity and specificity of MCPS Top-1, MCPS Top-100, Lutefisk, PEAKS and PepNovo for GPM, ISB and ISB2 data based on the different charge categories.

GPM Data. Sequencing results for GPM data is given in Figure 7.1. When comparing the MCPS Top-1 results, we observe that MCPS performs better for all data compared to PepNovo and PEAKS and is on a par with Lutefisk for +4 data (Sensitivity of 0.026 vs 0.027). Lutefisk performs the best on the whole.

We also observe in general that the sensitivity of the algorithms are not high on GPM data and in fact drops as the spectrum charge goes from +3 to +5, with +2 sequences having the best results. Sensitivity for +4 and +5 spectrum is on average **0.03-0.05** (recover **3-5%** of the

peptide).

We see however that when considering the best candidate out of the Top-100 candidates (MCPS Top-100), there is marked improvement in MCPS over the sequencing results of the other algorithms. This suggest that MCPS need only consider a narrow band of candidates in order to push up the rank of the best candidate during further post-processing of the results.

The specificity results show the same pattern as the sensitivity results and have about the same value for all algorithms except MCPS Top-100, where there is a huge difference between the specificity and sensitivity values (specificity values are about 2-4 times that of sensitivity values). This has been analyzed in Sections 6.2.2 and 6.2.4.1, and explains the general poor sensitivity values for all the algorithms for GPM data.

ISB2 Data. For +3 data, MCPS Top-1 is slightly better than Lutefisk and PEAKS (by **0.01**~**0.02**), with PepNovo better than MCPS by ~**0.035** (3.5% more peptide recovered) as shown in Figure 7.2. The sequencing result of MCPS Top-1 on ISB2 data is not as strong as PEAKS or PepNovo especially for +2 data (difference of **0.09**~**0.11**). For +1 data, MCPS Top-1 is better than Lutefisk and on a par with PEAKS (**0.28** for PEAKS and **0.27** for MCPS Top-1), with PepNovo having the best result.

As with GPM data, +2 ISB2 data shows the best sequencing results for each of the algorithms (Sensitivity difference of ~**0.11** between GPM +2 and +3 data, and ~**0.16** between ISB2 +2 and +3 data) compared to the other charges. When we compare the UB-Sensitivity values for +2 and +3 GPM and ISB2 data, we find that they do not differ by much (UB-Sensitivity difference of **0.08** between GPM c+2 and +3 data, and **0.01** between ISB2 +2 and +3 data, as obtained from results in Section 6.2.2). This indicates that in most mass spectrometers, +2 data have the **strongest fragmentation patterns** which can be exploited by the algorithm's scoring function.

Once again if we consider the MCPS Top-100 results, MCPS has a better performance than the other algorithms for all spectra. This indicates that post-processing will only once again have to focus on a narrow band of possible candidate peptides to improve their ranking.

Specificity results shows the same pattern and values as the sensitivity results.

ISB Data. Sequencing results for ISB is given in Figure 7.3. For +3, MCPS Top-1 has a higher sensitivity than all of the algorithms (0.215 compared to 0.144 for PepNovo which is the second best). This coupled with the ISB2 charge 3 results indicates that MCPS is suitable for sequencing higher charge data (charge 3) for ESI based data.

For +2 data, MCPS Top-1 performs on par with Lutefisk (0.23) while PepNovo has the best sequencing results (0.36). However, MCPS Top-1 results perform badly for +1 data. This is to be expected as the other algorithms are highly tuned for +1 and +2 ESI data.

When looking at the MCPS Top-100 results, we see that MCPS is slightly worse than the best algorithm PepNovo for +1 data and is slightly better for +2 data (~0.03 better sensitivity than PepNovo). An interesting pattern to note is that at Top-100, the sequencing result for +3 data is slightly better than for +2 (0.02 better) which is different from that of GPM and ISB2 data (sensitivity results are always worse for +3 data compared to +2 data). This suggest that MCPS scoring function is able to make use of +3 spectra fragmentation patterns better than +2 spectra patterns.

We see that specificity results again follows the same pattern as sensitivity results for all algorithms.

Conclusion on sensitivity and specificity results. From our analysis of Sensitivity and Specificity values of the different algorithms, we see that MCPS Top-1 does the best for +3 ISB data and second best for +3 ISB2 data. This indicates that the MCPS algorithm is well suited for sequencing multi-charge data. Next, we conclude that due to the many missing real peaks in the GPM data, all algorithms perform badly, with Lutefisk being the best, followed closely by MCPS Top-1.

In general due to the similarity of the specificity and sensitivity results for ISB and ISB2 data for all algorithms, we conclude for all tested algorithms (i) that a candidate peptide that can recover more of the canonical peptide will also contain less noise (non-matching amino acids) and vice versa, and (ii) both candidate and canonical peptide are about the same length.

MCPS Top-100 has the best result for all data sets and charge categories except for +1 ISB2 data. Thus we conclude that in further refinements of the MCPS algorithm, we need only focus

on a narrow band of the top 100 candidates.

7.2.2 Predictions with Correct Tags of Length $\geq x$

Here we compare the ratio of predictions with correct tags of length $\geq x$ for the different algorithms. In the tables generated, bold values indicate they are the best for that x value (MCPS (Top-100) values are not bolded even when they are the best). Since tag based database search only requires correct tags of length ≥ 3, we italicize values for $x = 3$. Note that when used for database search, an algorithm A with a higher ratio of correct predictions for $x = 3$ compared to another algorithm B is better than B even when B has a better ratio than A at higher x values.

ISB Data. In Table 7.1, we see the results for +1,+2 and +3 ISB data. For +3 data, MCPS Top-1 does the best for $x = 1..4$, and is slightly worse than PepNovo for $x = 5..10$. We see that MCPS Top-1 generates candidates with correct tags of length ≥ 3 ($x = 3$) more than half the time (0.51), and this is much better than the second best PepNovo (0.31). This indicates that MCPS Top-1 is much better than the rest for use with database search for +3 ESI data. The ratios are generally very low for $x \geq 8$.

For +2 data, PepNovo generally gives the best result for higher x values. MCPS Top-1 does better than Lutefisk for $x = 1..4$, and is slightly worse for higher x values. MCPS Top-1 has the second highest value for $x = 3$ and together with PepNovo would be a good candidate for use with database search.

For +1 data, we see that PepNovo has highest ratio for all x values, and this is especially apparent for bigger x values (≥ 5, except for 10), while MCPS (Top-1) has the worst result.

MCPS (Top-100) gives the best result overall for +2 and +3 data except at high x values, where PepNovo and PEAKS does better. This could be due to the fact that such long tags have a large mass. This causes the measurement of such tags to be inaccurate due to machine precision issues and due to mass shift caused by isotopes [76]. Gaps results which cannot be bridged in the spectrum graph due to difference in the PRMs of nodes exceeding the tolerance. This requires a post-processing step in bridging such gaps be developed for MCPS.

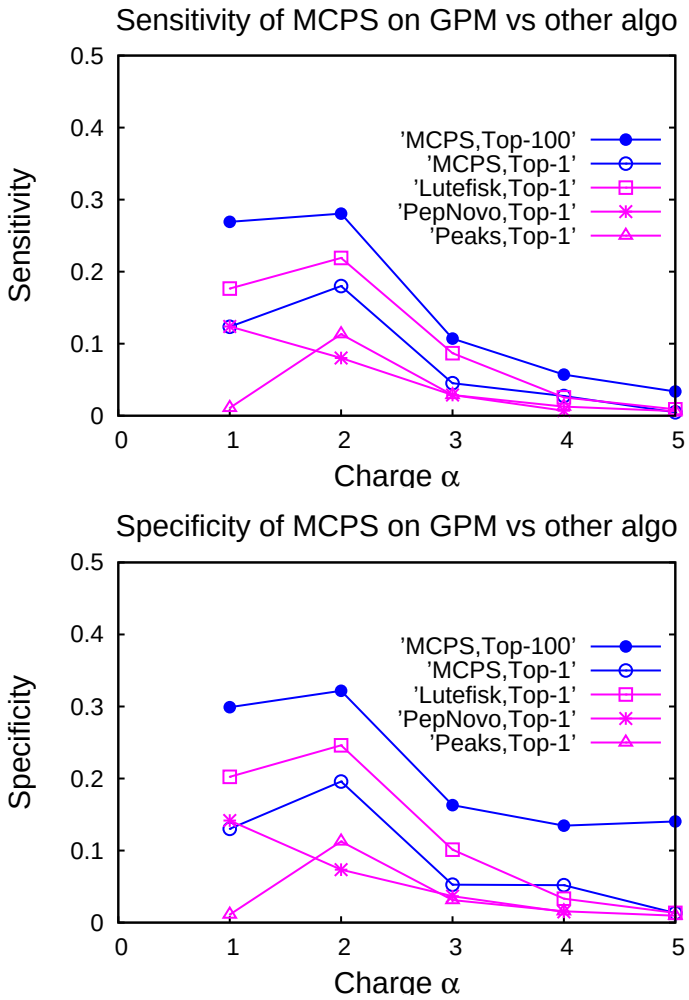


Figure 7.1: GPM sensitivity and specificity results for MCPS vs other algorithms. We plot for data in each charge category (+1,+2,+3,+4,+5 for GPM) the sensitivity and specificity using MCPS Top-1, MCPS Top-100, Lutefisk, PEAKS and PepNovo. For sensitivity results, comparing MCPS Top-1 result, we see that MCPS has better peptide recovery for +2,+3,+4 data except against Lutefisk. All algorithms do badly for +4 and +5 data. Using the best result in the Top-100 candidate peptides for MCPS (MCPS Top100), we see that MCPS has a marked improvement over all the other algorithms.

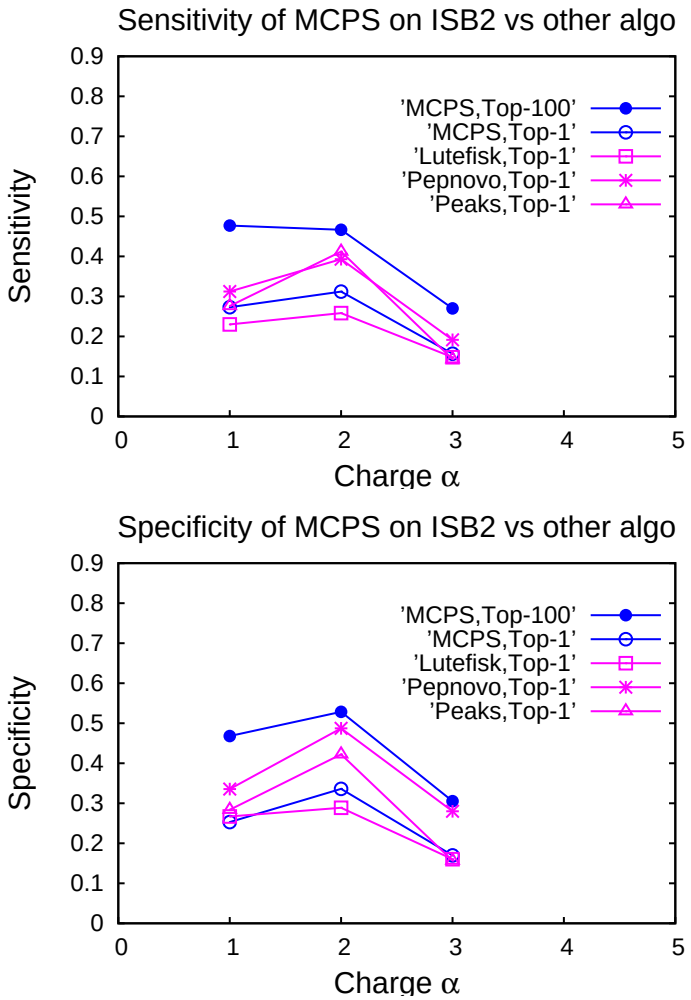


Figure 7.2: ISB2 sensitivity and specificity results for MCPS vs other algorithms. For sensitivity results, MCPS Top-1 does not do as well as PEAKS or PepNovo for +2 data. However, it is better or on a par with Lutefisk in all 3 charge categories. For +3 data, PepNovo does the best (3.5% more recovered peptide than MCPS Top-1), with MCPS slightly better than the rest ($1 \sim 2\%$ more recovered peptide). MCPS Top-100 does better in all charge categories compared to the other algorithms. For specificity results, specificity values follows that for sensitivity values for all the algorithms.

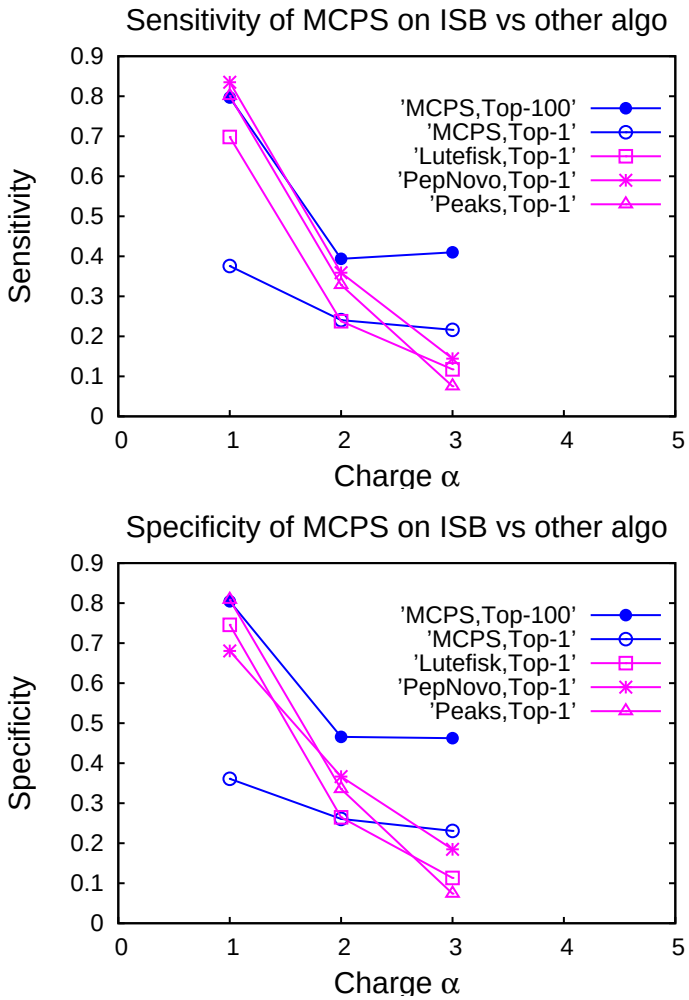


Figure 7.3: ISB sensitivity and specificity results for MCPS vs other algorithms. For sensitivity results, we see that MCPS Top-1 has better peptide recovery for +3 data against all other algorithms. It has comparable sequencing with Lutefisk for +2 data, but does badly for +1data. MCPS Top-100 however has a marked improvement in +1 data (comparable with PepNovo) and +2 data (~3% better sensitivity than PepNovo). An interesting pattern to note is that for MCPS Top-100, the sensitivity result for +3 data is slightly better than for +2 (2% better). For specificity results, again specificity values follow the same pattern as sensitivity values.

ISB2 Data. Table 7.2 shows the result for ISB2 data. +3 results indicate that MCPS Top-1 is on a par with PepNovo for $x = 1..3$, coming in second for $x = 3$. PepNovo does the best for $x \geq 4$.

For +2 and +1 data, PepNovo is generally the best. MCPS Top-1 is better or on a par with PEAKS for $x = 1..4$, and is second to PepNovo for $x = 3$.

MCPS (Top-100) does the best overall for all charge categories and indicate that improvement to the MCPS algorithm should focus on the top 100 candidates and study their properties closely.

GPM Data. Table 7.3 contains the results for GPM +1,+2,+3,+4, and +5 data. We observe that ratios are very low for + 4 and +5 data (no correct tags of length > 5). MCPS Top-1 does better than Lutefisk for +4 data, while Lutefisk is the best for the other charges. MCPS Top-1 is the next best for +2 and +3 data. PepNovo is slightly better than MCPS Top-1 for charge 1. In general we note that PepNovo and PEAKS does badly for charge 2 to 5 (PepNovo does no run for charge 5 data and thus could not be compared with), with PEAKS not having any correct tags of length > 1.

Conclusion. The quality of MCPS predictions are good for +3 ISB and ISB2 data, being the best in this category for ISB data and second best for ISB2 data. In both cases, the ratio of prediction of correction tags of length ≥ 3 is $> 40\%$. This implies that MCPS results can be useful as tags for database search of charge 3 ESI based data. All algorithms do badly for GPM data.

7.2.3 Distribution of Predictions with Correct Tags of Length ≥ 3

There are instances where using one de novo sequencing algorithm will not get the best result, especially for generating tags for database search. In this case, it would be good to use another sequencing algorithm which can generate hits for the cases where the former could not. It would thus be informative to measure the distribution of cases where predictions were correct with tags of length ≥ 3 among the algorithms.

We perform this for +3 ISB and ISB2 cases, and compare MCPS with PepNovo, where

ISB (+1 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	0.82	0.63	*0.50*	0.31	0.25	0.25	0.20	0.09	0.11	**0.20**
PEAKS	0.94	0.88	*0.88*	0.81	0.75	0.69	**0.73**	**0.72**	0.44	**0.20**
PepNovo	**1.0**	**1.0**	***1.0***	**1.0**	**0.88**	**0.88**	**0.73**	**0.72**	**0.56**	0.0
Lutefisk	**1.0**	**1.0**	***1.0***	0.69	0.31	0.25	0.27	0.36	0.22	0.0
MCPS Top-100	1.0	1.0	*0.94*	0.88	0.88	0.63	0.53	0.45	0.44	0.20

ISB (+2 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	**0.78**	**0.63**	*0.48*	0.35	0.22	0.14	0.09	0.06	0.05	0.02
PEAKS	0.61	0.50	*0.46*	0.42	0.37	0.31	0.23	0.19	**0.16**	**0.12**
PepNovo	0.65	0.60	***0.55***	**0.50**	**0.43**	**0.38**	**0.29**	**0.21**	0.15	0.11
Lutefisk	0.59	0.44	*0.35*	0.28	0.19	0.15	0.11	0.08	0.06	0.06
MCPS Top-100	0.96	0.87	*0.75*	0.62	0.47	0.36	0.26	0.16	0.11	0.05

ISB (+3 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	**0.84**	**0.69**	***0.51***	**0.32**	0.19	0.10	0.06	0.04	0.04	0.01
PEAKS	0.40	0.23	*0.16*	0.11	0.06	0.04	0.03	0.02	0.01	0.0
PepNovo	0.44	0.36	*0.31*	0.25	**0.20**	**0.17**	**0.14**	**0.09**	**0.05**	**0.02**
Lutefisk	0.55	0.35	*0.20*	0.12	0.07	0.04	0.02	0.02	0.01	0.0
MCPS Top-100	0.99	0.97	*0.89*	0.75	0.54	0.39	0.23	0.16	0.12	0.05

Table 7.1: % of Predictions with Correct tags of Length $\geq x$ for ISB Data. For +3 data, MCPS (Top-1) does best for $x = 1..4$, while doing slightly worse than PepNovo for $x = 5..10$. MCPS (Top-1) correct tags of length ≥ 3 more than 50% of the time (0.51), which is much better than PepNovo (31% of the time). For +1 and +2 data, PepNovo gives the best result for all x values except $x \geq 9$. MCPS Top-1 does the worst for +1 data, while MCPS Top-100 is more or less on a par with PEAKS. For +2 data, MCPS Top-1 does better than Lutefisk for $x = 1..4$. MCPS Top-1 has the second highest value for $x = 3$ while MCPS Top-100 gives the best result overall except for $x = 8..10$.

ISB2 (+1 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	**0.89**	0.69	*0.44*	0.27	0.15	0.10	0.05	0.03	0.01	0.0
PEAKS	0.79	0.57	*0.39*	0.27	0.19	0.14	0.09	0.06	0.04	**0.02**
PepNovo	0.85	**0.72**	***0.55***	**0.40**	**0.28**	**0.22**	**0.11**	**0.07**	0.04	0.01
Lutefisk	0.75	0.49	*0.32*	0.21	0.15	0.11	0.08	0.06	**0.05**	**0.02**
MCPS Top-100	0.99	0.92	*0.80*	0.64	0.48	0.33	0.23	0.15	0.10	0.05

ISB2 (+2 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	**0.92**	0.81	*0.67*	0.50	0.36	0.25	0.16	0.10	0.05	0.03
PEAKS	0.85	0.76	*0.65*	0.56	0.47	0.40	0.31	**0.26**	**0.20**	**0.15**
PepNovo	0.90	**0.83**	***0.75***	**0.67**	**0.56**	**0.47**	**0.33**	0.24	0.18	0.11
Lutefisk	0.78	0.56	*0.41*	0.30	0.23	0.17	0.13	0.08	0.07	0.05
MCPS Top-100	0.99	0.95	*0.87*	0.77	0.66	0.54	0.42	0.31	0.21	0.12

ISB2 (+3 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	**0.83**	**0.59**	*0.41*	0.28	0.17	0.10	0.04	0.03	0.01	0.01
PEAKS	0.67	0.46	*0.32*	0.23	0.16	0.10	0.06	0.03	0.01	0.01
PepNovo	0.71	0.61	***0.49***	**0.39**	**0.28**	**0.20**	**0.11**	**0.06**	**0.04**	**0.02**
Lutefisk	0.73	0.49	*0.29*	0.18	0.09	0.07	0.04	0.02	0.0	0.0
MCPS Top-100	0.98	0.87	*0.73*	0.55	0.41	0.24	0.15	0.11	0.07	0.03

Table 7.2: % of Predictions with Correct tags of Length $\geq x$ for ISB2 Data. For +3,+2 and +1 data, PepNovo is the best overall. For +3 data, MCPS Top-1 is on a par with PepNovo for $x = 1..3$ and is second for $x = 3$. For +2 data, MCPS Top-1 is on a par with PEAKS for $x = 1..4$, and is second for $x = 3$. For +1 data, MCPS Top-1 is comparable to PEAKS for $x = 1..4$. MCPS Top-1 comes in second for $x = 3$.

GPM (+1 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	**0.54**	0.29	*0.18*	0.11	0.05	0.04	0.01	0.01	0.01	0.0
PEAKS	0.12	0.0	*0.0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PepNovo	0.46	0.25	*0.19*	0.14	0.10	0.06	0.04	0.02	0.02	0.0
Lutefisk	0.49	**0.35**	***0.26***	**0.20**	**0.17**	**0.12**	**0.08**	**0.05**	**0.04**	**0.02**
MCPS Top-100	0.82	0.61	*0.41*	0.32	0.21	0.14	0.10	0.06	0.05	0.04

GPM (+2 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	0.55	**0.37**	*0.26*	0.19	0.11	0.08	0.06	0.04	0.03	0.02
PEAKS	0.07	0.01	*0.0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PepNovo	0.22	0.14	*0.10*	0.07	0.05	0.04	0.02	0.01	0.0	0.0
Lutefisk	0.56	**0.37**	***0.27***	**0.20**	**0.16**	**0.12**	**0.09**	**0.07**	**0.05**	**0.04**
MCPS Top-100	0.76	0.54	*0.40*	0.30	0.21	0.17	0.13	0.08	0.05	0.05

GPM (+3 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	0.34	0.14	*0.06*	0.03	0.02	0.01	0.0	0.0	0.0	0.0
PEAKS	0.08	0.0	*0.0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PepNovo	0.13	0.07	*0.05*	0.03	0.03	0.02	0.02	0.02	0.01	0.0
Lutefisk	**0.40**	**0.22**	***0.13***	**0.11**	**0.07**	**0.06**	**0.05**	**0.04**	**0.03**	**0.02**
MCPS Top-100	0.73	0.31	*0.15*	0.09	0.06	0.03	0.03	0.02	0.01	0.0

GPM (+4 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	**0.33**	**0.12**	***0.05***	0.01	0.0	0.0	0.0	0.0	0.0	0.0
PEAKS	0.11	0.0	*0.0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PepNovo	0.09	0.02	*0.0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Lutefisk	0.27	0.11	*0.04*	**0.02**	**0.01**	0.0	0.0	0.0	0.0	0.0
MCPS Top-100	0.66	0.24	*0.12*	0.03	0.0	0.0	0.0	0.0	0.0	0.0

GPM (+5 data)	$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x \geq 10$
MCPS Top-1	**0.11**	0.0	*0.0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PEAKS	0.03	0.0	*0.0*	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PepNovo	N/A	N/A	*N/A*	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Lutefisk	**0.11**	**0.03**	***0.03***	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MCPS Top-100	0.54	0.16	*0.03*	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 7.3: % of Predictions with Correct tags of Length $\geq x$ for GPM Data. In general, quality of predictions are bad for GPM data. Lutefisk being the best for GPM data, predicts $< 26\%$ of the time, correct tags of length ≥ 3. Virtually no predictions of correct tags of length ≥ 3 for +4 and +5 data for all algorithms excluding MCPS Top-100.

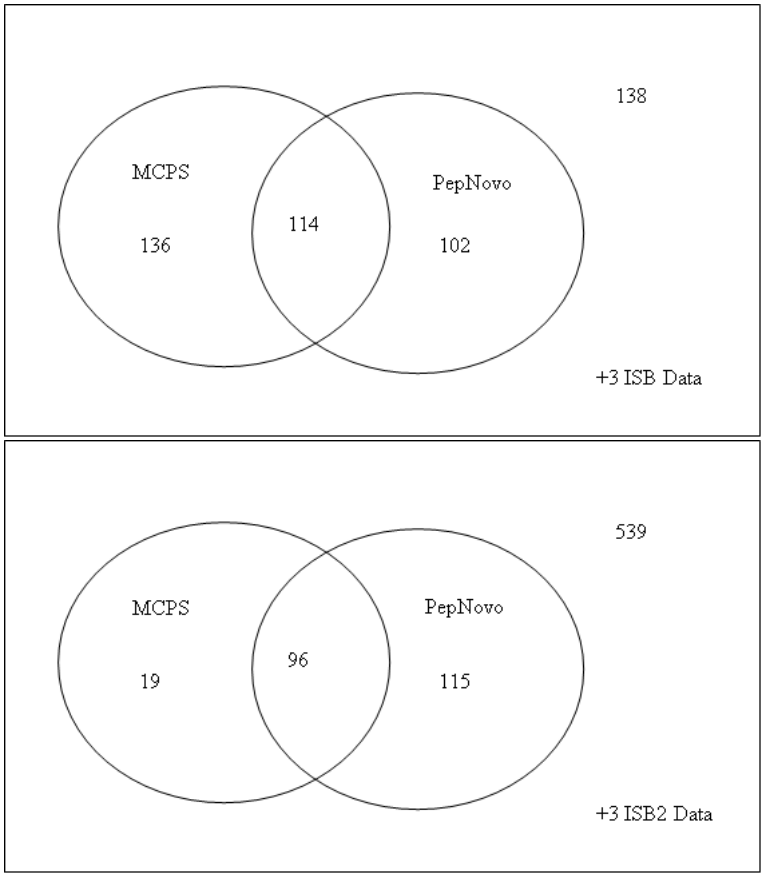


Figure 7.4: Distribution of Predictions with Correct Tags of Length ≥ 3 between MCPS and PepNovo: The venn diagram for +3 ISB data shows that 136 cases of predictions with correct tags of length ≥ 3 were uniquely generated by MCPS. This covers 27.8% of the total correct predictions, while the unique cases covered by PepNovo is 20.8%. This shows that both MCPS and PepNovo can be used to tandem to generate tags for database search. For +3 ISB2 data, the unique cases covered by MCPS accounts for around 8.6% of the cases.

we then draw the distribution of cases in a form of a venn diagram. We compare MCPS and PepNovo since they are the best two algorithms for +3 ISB and ISB2 data.

From Figure 7.4, we find that for +3 ISB data, 136 cases were uniquely predicted by MCPS. This accounted for 27.8% of all the total cases. PepNovo uniquely predicted 20.8% (102) of the cases. They co-predicted 23.3% of the cases. The remaining 28.1% failed to have a prediction with a correct tag of length ≥ 3.

For +3 ISB2 data, even though MCPS only uniquely covered 8.6% (19) of the cases, this is much more than the unique cases covered by PEAKS (2%) and Lutefisk (2.7%) which is not shown in the figure. Thus it makes sense to use MCPS in tandem with PepNovo for generation of peptide tags for database search of multi-charge ESI data.

7.3 Sequencing Using +3 ion-types vs not Using +3 ion-types

In our sequencing, we have used some +3 ion-types in our ion-type sets which is currently not used by any other algorithm. In order to confirm whether these ion-types have an impact on MCPS, we measure the sensitivity of our sequencing result with and without using the +3 ion-types for +3 and higher data.

GPM Data. Table 7.4 shows the sensitivity with and without the +3 ion-types. We see that there is a very slight improvement for the MCPS Top-1 results, with a slight drop for the MCPS Top-100 results. This could indicate that +3 peaks have too low a likelihood of occurrence in the path of good candidate peptides to affect sensitivity by much.

ISB2 Data. Table 7.5 indicates that there is an improvement in the sensitivity of MCPS Top-1 and MCPS Top-100 results when we consider +3 ion-types. This shows that they could be important as a bridge (used in the bridging step) between good quality mono-chromatic sub-paths in the spectrum graph, since none of the +3 ion-types are used in the pruning step. It could also indicate that they are useful as supporting ions to improve the score of good quality paths in the spectrum graph.

ISB Data. ISB data also shows the same improvement in sensitivity as seen from Table 7.6. In the characterization study (Chapter 4) we have concluded that +3 ion-type for ISB do not help to recover additional amino acids by much. Thus most of the improvement comes from increasing the rank of good quality paths by using these as supporting ions.

Conclusion on using +3 ions. Using +3 ions helps in improving the sequencing results for ISB and ISB2 data. They do so mainly as supporting ions. +3 ions do not help in sequencing GPM data, indicating they occur too infrequently in the paths of good candidate peptides to affect sensitivity by much.

GPM Results	**Sensitivity** (no +3 ions)	**Sensitivity** (with +3 ions)
Charge 3 (Top-1)	0.044	0.045 (*+0.001*)
Charge 3 (Top-100)	0.109	0.107 (*-0.002*)
Charge 4 (Top-1)	0.024	0.027 (*+0.003*)
Charge 4 (Top-100)	0.059	0.057 (*-0.002*)
Charge 5 (Top-1)	0.003	0.005 (*+0.002*)
Charge 5 (Top-100)	0.035	0.034 (*-0.001*)

Table 7.4: Comparison of Sensitivity between using +3 ions and not using +3 ions. Very slight improvement in Top-1 sensitivity when using +3 ion-types compared to without using them. There is also a very slight drop in the Top-100 sensitivity.

ISB2 Top-1 Results	**Sensitivity** (no +3 ions)	**Sensitivity** (with +3 ions)
Charge 3 (Top-1)	0.146	0.157 (*+0.011*)
Charge 3 (Top-100)	0.249	0.270 (*+0.021*)

Table 7.5: Comparison of Sensitivity between using +3 ions and not using +3 ions. Improvement in the sensitivity results for both Top-1 and Top-100 when using +3 ion-types.

ISB Top-1 Results	**Sensitivity** (no +3 ions)	**Sensitivity** (with +3 ions)
Charge 3 (Top-1)	0.185	0.216 (*+0.031*)
Charge 3 (Top-100)	0.406	0.410 (*+0.004*)

Table 7.6: Comparison of Sensitivity between using +3 ions and not using +3 ions. Improvement in sensitivity for Top-1, and a very slight improvement for Top-100 results when using +3 ion-types.

Chapter 8

Conclusion

8.1 Summary

In summary, we have developed a generalized model of multi-charge spectra for sequencing purposes. We have applied this model in a characterization study of datasets with multi-charge spectra. We conclude that for GPM data, inclusion of multi-charge ions will affect improve the upper bound on the amount of recoverable peptide. We conclude that for ISB and Orbitrap data, inclusion of multi-charge ions will not improve the upper bound on the amount of recoverable peptide but can improve the scores of good candidate peptides and this can help in better ranking of such peptides.

We have also developed a novel de novo sequencing algorithm which makes use of multi-charge ion-types (based on our conclusion in our characterization study) and the observation that monochromatic tags of abundant ion-types represent strong signals. Both are exploited in the building of our extended spectrum graph. Most importantly, the monochromatic tags are used in a scoring function that boosts up the scores of such tags by performing a suffix-k lookback when scoring a path that contains them.

From the sequencing result of GPM data, we conclude that all the tested de novo sequencing algorithm suffers from a poor recovery of the canonical peptide (especially PEAKS and PepNovo for charge 2 and above spectra). This is due to missing peaks corresponding to ion-types in the ion-type set used. Also, a lack of strong fragmentation patterns for GPM data results in inability

of all algorithms tested to differentiate between good and bad candidates. As a consequence, the use of multi-charge ion-types do not have much impact to the sequencing results. However among all algorithm, MCPS still does better than PEAKS and PepNovo, losing out only to Lutefisk.

Sequencing results for ISB and ISB2 data suggests that MCPS is better for ISB charge 3 data than PEAKS, PepNovo and Lutefisk, and loses only to PepNovo for ISB2 charge 3 data. The fact that $> 50\%$ of the candidate peptides generated by MCPS for charge 3 ISB data contains correct tags of over length 3, and that it generates a good number of such predictions not generated by the other algorithms for charge 3 ISB2 data, suggest that MCPS can be used together with other de novo sequencing algorithms especially PepNovo to generate solutions as tags for database search of charge 3 ESI-based spectra.

Comparing using charge 3 ion-types to not using charge 3 ion-types, we see that there is an improvement to the sensitivity and this shows that charge 3 ion-types are useful as supporting ions in improving the score and thus the ranking of good candidate peptides.

8.2 Future Work

For future work in improving MCPS, the parent mass correction procedure can be applied to spectra before sequencing.

Another area that can be explored is the region based boosting of the MCScore function. The idea here is that monochromatic tags do not have their scores boosted uniformly based on length but depending on which region of the peptide they reside, the tags will be boosted differently, maybe even different parts of the tag boosted with a different multiplier if those parts lie in different regions of the peptide. This is inspired by the fact that different ion-types are abundant in different regions of the peptide. This can also be applied to the selection of the ion-type sets for different datasets.

Research can also be done in modifying and applying MCPS in generating of peptide tags for database search especially for PTM peptides.

Bibliography

[1] V. Bafna and N. Edwards. On de novo interpretation of peptide mass spectra. *RECOMB 2003*, pages 9–18, 2003.

[2] R. Bakhtiar and F. L. S. Tse. Biological mass spectrometry: a primer. *Mutagenesis*, 15(5): 415–430.

[3] C. Bartels. Fast algorithm for peptide sequencing by mass spectroscopy. *Biomedical and Environmental Mass Spectrometry*, 19:363–368, 1990.

[4] R. C. Beavis and D. Fenyö. Database searching with mass spectrometric information. *Proteomics*, pages 22–26, 2000.

[5] K. Biemann, C. Cone, B. R. Webster, and G. P. Arsenault. Determination of the amino acid sequence in oligopeptides by computer interpretation of their high-resolution mass spectra. *Journal of the American Chemical Society*, 88(23).

[6] T. Chen, M. Y Kao, M. Tepel, J. Rush, and G. M. Church. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 8:325–337, 2001.

[7] K. F. Chong and H. W. Leong. Mcps : A mono-chromatic peptide sequencer for multi-charged mass spectra. *RECOMB Satellite Conference on Computational Proteomics 2011*, pending review, 2011.

[8] K. F. Chong, N. Kang, and H. W. Leong. Characterization of multi-charge mass spectra for peptide sequencing. *Asia Pacific Bioinformatics Conference*, pages 109–119, 2006.

[9] K. F. Chong, K. Ning, and H. W. Leong. Characterization of multi-charge mass spectra for peptide sequencing. *Journal of BioInformatics and Computational Biology*, 4:1329–1352, 2006.

[10] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. Introduction to algorithms. *Proceedings of the Asia BioInformatics Conference*, 1989.

[11] K. A. Cox, S. J. Gaskel, M. Morris, and A. Whiting. Role of the site of protonation in the low-energy decompositions of gas-phase peptide ions. *Journal of the American Society for Mass Spectrometry*, 7:522–531, 1996.

[12] R. Craig, J.P. Cortens, and R.C. Beavis. Open source system for analyzing, validating and storing protein identification data. *Journal of Proteome Research*, 3:1234–1242, 2004.

[13] V. Dancik, T. Addona, K. Clauser, J. Vath, and P.A. Pevzner. De novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 6:327–342, 1999.

[14] J. Fernández de Cossío, J. Gonzales, and V. Besada. A computer pogram to aid the sequencing of peptides in collision-activated decomposition experiments. *Computer Applications in Biosciences*, 11:427–434, 1995.

[15] A. R. Dongre, J. L. Jones, A. Somogyi, and V. H. Wysocki. Influence of peptide composition, gas-phase basicity, and chemical modification on fragmentation efficiency: Evidence for the mobile proton model. *Journal of the American Chemical Society*, 118:8365–8374, 1996.

[16] J. E. Elias, F. D. Gibbons, O. D. King, F. P. Roth, and S. P. Gygi. Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nature Biotechnology*, 22:214–219, 2004.

[17] J. K. Eng, A. L. McCormack, and J. R. Yates III. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *JASMS*, 5(11): 976–989, 1994.

[18] B. Fischer, V. Roth, F. Roos, J. Grossmann, S. Baginsky, P. Widmayer, W. Gruissem, and J. M. Buhmann. Novohmm: A hidden markov model for de novo peptide sequencing. *Analytical Chemistry*, 77:7265–7273, 2005.

[19] A. Frank. Predicting intensity ranks of peptide fragment ions. *Journal of Proteome Research*, 8(5), .

[20] A. Frank. A ranking-based scoring function for peptide-spectrum matches. *Journal of Proteome Research*, 8(5), .

[21] A. Frank and P. Pevzner. Pepnovo: De novo peptide sequencing via probabilistic network modeling. *Analytical Chemistry*, 77:964–973, 2005.

[22] A. Frank, S. Tanner, and P. Pevzner. Peptide sequence tags for fast database search in mass spectrometry. *RECOMB 2005*, 2005.

[23] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Anals of Statistics*, 28:337–407, 2000.

[24] M. Gashler, C. Giraud-Carrier, and T. Martinez. Decision tree ensemble: Small heterogeneous is better than large homogeneous. *Seventh International Conference on Machine Learning and Applications (ICMLA 08)*, pages 900–905, 2008.

[25] A. Gavin, M. Bösche, and R. Krause. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.

[26] S. Gay, P. A. Binz, D. F. Hochstrasser, and R. D. Appel. Modeling peptide mass fingerprinting data using the atomic composition of peptides. *Electrophoresis*, 20:3527–3534, 1999.

[27] J. Grossmann, F. F. Roos, M. Cieliebak, Z. Lipták, L. K. Mathis, M. Müller, W. Gruissem, and S. Baginsky. Audens: A tool for automatic de novo peptide sequencing. *Journal of Proteome Research*, 4(5):1768–1774, 2005.

[28] C.W. Hamm, W.E. Wilson, and D.J. Harvan. Peptide sequencing program. *Computer Applications in Biosciences*, 2:115–118, 1986.

[29] Y. Han, B. Ma, and K. Zhang. Spider: software for protein identification from sequence tags with de novo sequencing error. *Journal of Bioinformatics and Computational Biology*, 3(3):697–716, 2005.

[30] M. Havilio, Y. Haddad, and Z. Smilansky. Intensity-based statistical scorer for tandem mass spectrometry. *Analytical Chemistry*, 75:435–444, 2003.

[31] Y. Ho, A. Gruhler, and A. Heilbut. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415:180–183, 2002.

[32] R.J Johnson and K. Biemann. Computer program (seqpep) to aid in the interpretation of high-energy collision tandem mass spectra of peptides. *Biomedical and Environmental Mass Spectrometry*, 18:945–957, 1989.

[33] E. A. Kapp, F. Schütz, and G. E. Reid et. al. Mining a tandem mass spectrometry database to determine the trends and global factors influencing peptide fragmentation. *Analytical Chemistry*, 75:6251–6264, 2003.

[34] A. Keller, S. Purvine, A. Nesvizhskii, S. Stolyar, D.R. Goodlett, and E. Kolker. Experimental protein mixture for validating tandem mass spectral analysis. *OMICS*, 6:207–212, 2002.

[35] S. Kim, N. Bandeira, and P. A. Pevzner. Spectral profiles, a novel representation of tandem mass spectra and their applications for de novo sequencing and identification. *Molecular and Cellular Proteomics*, 8(6).

[36] S. Kim, N. Gupta, N. Bandeira, and P. A. Pevzner. Spectral dictionaries: integrating de novo peptide sequencing with database search of tandem mass spectra. *Molecular and Cellular Proteomics*, 8:53–69, 2009.

[37] J. Klimek, J. S. Eddes, L. Hohmann, J. Jackson, A. Peterson, S. Letarte, P. R. Gafken, J. E. Katz, P. Mallick, H. Lee, A. Schmidt, R. Ossola, J. K. Eng, R. Aebersold, and D. B.

Martin. The standard protein mix database: A diverse data set to assist in the production of improved peptide and protein identification software tools. *Journal of Proteome Research*, 7:96–103, 2008.

[38] X. W. Liu, B. Z. Shan, L. Xin, and B. Ma. Better score function for peptide identification with etd ms/ms spectra. *BMC Bioinformatics*, 11 Suppl 1:S4, 2010.

[39] B. Lu and T. Chen. A suboptimal algorithm for de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 10:1–12, 2003.

[40] B. W. Lu and T. Chen. Algorithms for de novo peptide sequencing using tandem mass spectrometry. *BIOSILICO*, 2:85–90, 2004.

[41] 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 ms/ms. *Rapid Communications in Mass Spectrometry*, 17:2337–2342, 2003.

[42] B. Ma, K.Z Zhang, and C.Z. Liang. An effective algorithm for peptide de novo sequencing from ms/ms spectra. *Journal of Computer and System Sciences*, 70:418–430, 2005.

[43] J. M. Malard, A. Heredia-Langner, D. J. Baxter, K. H. Jarman, and W. R. Cannon. Constrained de novo peptide identification via multi-objective optimization. *IEEE International Workshop on High Performance Computational Biology (HICOMB)*, 2004.

[44] M. Mann and M. Wilm. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Analytical Chemistry*, 66:4390–4399, 1994.

[45] A. L. McCormack, A. Somogyi, A. R. Dongre, and V. H. Wysocki. Fragmentation of protonated peptides: surface-induced dissociation in conjunction with a quantum mechanical approach. *Analytical Chemistry*, 65(20):2859–2872, 1993.

[46] E. Nathan and L. Ross. Generating peptide candidates from amino-acid sequence databases for protein identification via mass spectrometry. *Lecture Notes In Computer Science*, 2452: 68–81, 2002.

[47] K. Ning and H. W. Leong. Algorithm for peptide sequencing by tandem mass spectrometry based on better preprocessing and anti-symmetric computational model. *Computational Systems Bioinformatics Conference*, 6:19–30, 2007.

[48] K. Ning, K. F. Chong, and H. W. Leong. De novo peptide sequencing for mass spectra based on multi charge strong tags. *Proceedings of the Asia BioInformatics Conference*, pages 287–296, 2007.

[49] D. N. Perkins, D. J. C. Pappin, D. M. Creasy, and J. S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20:3551–3567, 1999.

[50] S. Pevtsov, I. Fedulova, H. Mirzaei, C. Buck, and X. Zhang. Performance evaluation of existing de novo sequencing algorithms. *Journal of Proteome Research*, 5:3018–3028, 2006.

[51] P. Pevzner. Personal communication. 2005.

[52] P. A. Pevzner, V. Dancik, and C. L. Tang. Mutation-tolerant protein identification by mass spectrometry. *Journal of Computational Biology*, 7(6):777–787, 2000.

[53] P. A. Pevzner, Z. Mulyukov, V. Dancik, and C. L. Tang. Efficiency of database search for identification of mutated and modified proteins via mass spectrometry. *Genome Research*, 11:290–299, 2001.

[54] M. J. Polce, D. Ren, and C. Wesdemiotis. Dissociation of the peptide bond in protonated peptides. *Journal of Mass Spectrometry*, 35:1391–1398, 2000.

[55] M. Pollack. The kth best route through a network. *Operations Research*, 9:578, 1961.

[56] T. Sakurai, T. Matsuo, H. Matsuda, and I. Katakuse. Paas 3: A computer program to determine probable sequence of peptides from mass spectrometric data. *Biomedical Mass Spectrometry*, 11:396–399, 1984.

[57] S. W. Sun, C. G. Yu, and Y. T. Qiao et. al. Deriving the probabilities of water loss and

ammonia loss for amino acids from tandem mass spectra. *Journal of Proteome Research*, 7:202–208, 2008.

[58] D. Tabb, A. Saraf, and J. Yates Jr. Gutentag: high-througput sequence tagging via an empirically derived fragmentation model. *Analytical Chemistry*, 75:2594–2604, 2003.

[59] D. L. Tabb., Y. Y. Huang, V. H. Wysocki, and J. R. Yates III. Influence of basic residue content on fragment ion peak intensities in low-energy collision-induced dissociation spectra of peptides. *Journal of Analytical Chemistry*, 76(5):1243–1248, 2004.

[60] J. S. Tan and H. W. Leong. Least-cost path in public transportation systems with fare rebates that are path- and time-dependent. *IEEE Intelligent Transportation Systems Conference*, 7:1000–1005, 2004.

[61] H. Tang. *Private Communications*, 2006.

[62] X. J. Tang, P. Thibault, and R. K. Boyd. Fragmentation reactions of multiply-protonated peptides and implications for sequencing by tandem mass spectrometry with low-energy collision-induced dissociation. *Analytical Chemistry*, 65:2824–2834, 1993.

[63] S. Tanner, H. J. 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*, 77(14):4626–4639, 2005.

[64] S. W. Tanner. Efficient and accurate bioinformatics algorithms for peptide mass spectrometry. *PhD. Dissertation*, 1997.

[65] J.A. Taylor and R.S. Johnson. Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry*, 11:1067–1075, 1997.

[66] J.A. Taylor and R.S. Johnson. Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Analytical Chemistry*, 73:2594–2604, 2000.

[67] D. Tsur, S. Tanner, E. Zandi, V. Bafna, and P. A. Pevzner. Identification of post-translational modifications by blind search of mass spectra. *Nature Biotechnology*, 23(12): 1562–1567, 2005.

[68] V. H. Wysocki, G. Tsaprailis, L. L. Smith, and L. A. Breci. Mobile and localized protons: a framework for understanding peptide dissociation. *Journal of Mass Spectrometry*, 35: 1399–1406, 2000.

[69] B. Yan, Y. Qu, F. Mao, V. Olman, and Y. Xu. Prime: A mass spectrum data mining tool for de novo sequencing and ptms identification. *Journal of Computer Science and Technology*, 20(4):483–490, 2005.

[70] J. Yates, P. Griffin, L. Hood, and J. Zhou. Computer aided interpretation of low energy ms/ms mass spectra of peptides. *Techniques in Protein Chemistry II*, pages 477–485, 1991.

[71] J. Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17: 712–716, 1971.

[72] A. L. Yergey, J. R. Coorssen, P. S. Backlund Jr, P. S. Blank, G. A. Humphrey, J. Zimmerberg, J. M. Campbell, and M. L. Vestal. De novo sequencing of peptides using maldi/tof-tof. *Journal of American Society for Mass Spectrometry*, 13:784–791, 2002.

[73] N. Zhang, R. Aebersold, and B. Schwikowski. Probid: A probabilistic algorithm to identify peptides through sequence database searching using tandem mass spectral data. *Proteomics*, 2002.

[74] Z. Q. Zhang. Prediction of low-energy collision-induced dissociation spectra of peptides. *Journal of Analytical Chemistry*, 76(14):3908–3922, 2004.

[75] Z. Q. Zhang. Prediction of low-energy collision-induced dissociation spectra of peptides with three or more charges. *Journal of Analytical Chemistry*, 77(19):6364–6373, 2005.

[76] Z. Q. Zhang, S. H. Guan, and A. G. Marshall. Enhancement of the effective resolution of mass spectra of high-mass biomolecules by maximum entropy-based deconvolution to

eliminate the isotopic natural abundance distribution. *Journal of American society for Mass Spectrometry*, 8:659–670, 1997.

[77] D. Zidarov, P. Thibault, M.J. Evans, and M.J. Bertrand. Determination of the primary structure of peptides using fast bombardment mass spectrometry. *Biomedical and Environmental Mass Spectrometry*, 19:13–16, 1990.

Appendix A

A.1 Parent Mass Correction

One problem faced when doing peptide sequencing by tandem mass spectrometry is the fact that a lot of the time, the precursor parent mass given in the mass spectrum is not accurate. There are a few possible reasons for this. Not including PTMs which will cause the actual peptide mass to be shifted, the accuracy of the mass spectrometry and isotopic atoms will cause the measurement of the parent peptide mass to be inaccurate. In the case of mass spectrometer accuracy, an instrument with an accuracy of 500 PPM (parts per million) can cause an error in the mass measurement of 0.5 Da per 1000Da of peptide mass. However, most modern mass spectrometers have an accuracy of 1-50 PPM, making errors by instrument inaccuracy negligible even for peptides masses of up to 5000Da. This leaves most of the error in parent mass to be due to the case of isotopic atoms. Zhang et al. [76] shows that at 10KDa the isotopic distribution ranges over 16Da, and this approximates to a 4Da range at 2.5KDa which is a reasonable mass for peptides. Moreover, based on the probability of occurrence of each of the isotopes of each of the organic atoms making up the amino acid, it has been shown that for peptides of length from 1-60 amino acids, the most prevalent isotopic atom will be C13, and there is a 50% chance of getting a C13 atom every 10 amino acids. Since most peptides have lengths of up to a max. of 40 amino acids, there is a 50% chance of a shift of up to 4Da. This concurs with the study of Zhang et al. [76]. Gay et al. [26] also shows empirically that the intensity of the M+1 isotopic peak for the peptide will exceed the intensity of the M+0 (mono-isotopic peak) at peptide masses above 1500Da. M+2 will do so at masses above 2000Da, M+3 at masses above

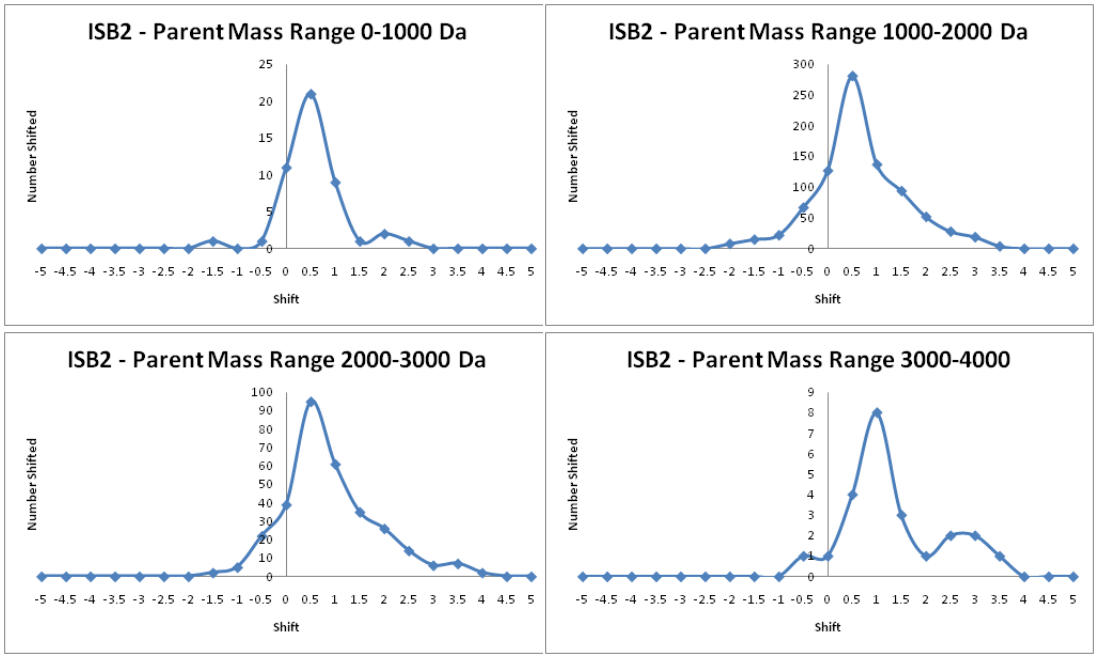


Figure A.1: Parent Mass Shifts for ISB2 data. We split the peptide masses into 4 range 0-1000, 1000-2000, 2000-3000 and 3000-4000 Da. The number of spectra with mass shift as indicated in the x-axis is then plotted for each mass range. We see that for masses 0-3000 Da, the majority of the peptides are shifted by 0.5 Da (the major peak in the diagrams). For masses 3000-4000 Da, the peptides are shifted by 1.0 Da. The range of mass shifts is [-2.0,4.0]. This indicates that ISB and ISB2 data are relatively accurate and error tolerances of 0.5Da will be able to deal with most spectra.

2500Da. Thus it is likely for the mass spectrometer to pick the isotopic peak instead of the mono-isotopic one. Also dynamic exclusion by mass spectrometers can cause the instrument to select the same peptide (possibly isotopically-enriched) for fragmentation resulting in incorrectly reported parent/precursor mass Tanner [64]. Incorrect parent mass causes a lot of problems when performing peptide sequencing because the calculation of the PRM from the C-terminal ions requires a correct parent mass to be assumed. Otherwise the scoring can be drastically off and result in poor sequencing results.

In Figure A.1, we show the parent mass shift for different parent mass ranges for ISB2 training data. We see that for mass range 0-3000 Da the majority of spectra have mass shift of 0.5 Da, while those from 3000-4000 Da , most have a mass shift of about 1.0 Da. The range of mass shifts is [-2.0Da, 4.0Da]. This suggest that ISB and ISB2 data are quite accurate and error tolerances of 0.5Da will be able to deal with most spectra.

ISB2	Top1	Top1-2	Top1-3
% of correct prediction	0.23	0.59	0.81

Table A.1: % of corrected parent masses for ISB2 using self-convolution. From the table, we see that using only the top result (mass bin with the most number of complimentary peaks) only result in 23% of the spectra being corrected, whereas using the top1-3 results, 81% of the spectra are corrected.

ISB2	Top1	Top1-2	Top1-3
% of correct prediction	0.38	0.77	0.89

Table A.2: % of corrected parent masses for ISB2 using self-convolution 2.0. From the table, we see that using only the top result (mass bin with the most number of complimentary peaks) only result in 23% of the spectra being corrected, whereas using the top1-3 results, 81% of the spectra are corrected.

A.1.1 Self-Convolution

The usual method in correcting the precursor parent mass is by computing the self-convolution of the fragments. This is done by assuming each peak to be both a y-ion and b-ion, and compute the total mass of each unique pair of peaks. The assumption here is that the mass bin which contains the actual peptide mass will also contain the most number of peak pairs (these peaks pairs are complimentary peaks which add up to be the actual peptide mass) as opposed to the mass bins which do not correspond to the peptide mass. A good bin size to work with here would be 0.5 Da. An experiment using the above self-convolution method obtained the following result in Table A.1 when performed on the ISB2 data. In the experiment only the top 50 peaks were used for the self-convolution, and only those mass bins with a [-4.0,1.0] Da range (maximum shift by isotopic atoms) from the experimental precursor mass is considered. It is usually unlikely for the actual peptide mass to be bigger than the experimental precursor mass. A prediction is correct when the actual mass between the correct parent mass and the predicted mass (based on the average mass in the mass bin) is < 0.5 Da.

A.1.2 Self-Convolution 2.0

Instead of using the number of complimentary peaks, we use the total intensity contributed by the complimentary peaks. This improves the results vastly as shown in Table A.2, especially for the Top1 (the best) result (15% improvement) and Top1-2 (best among the top 2) result (18% improvement).

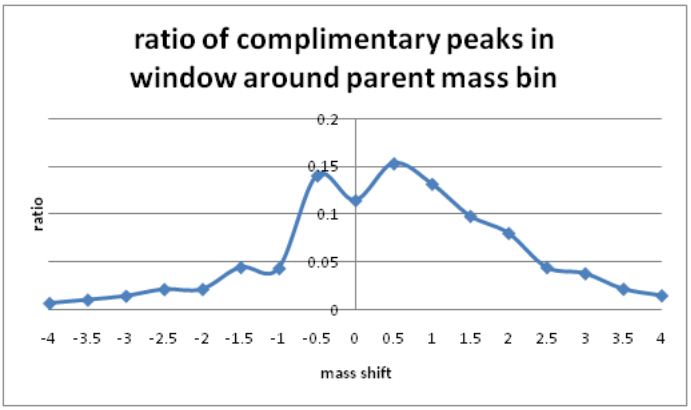


Figure A.2: Ratio of complimentary peaks in window around parent mass bin. In the figure, we see that using an error range of [-4.0,4.0] Da for the fragments themselves, most of the (87%) of the complementary peaks add up to masses that reside within the [-2.5,2.5] window around the actual parent peptide mass, but not all in the mass bin corresponding to the putative peptide mass, which only contributes ~11%.

A.1.3 Parent Mass Correction using Boosting Classifier

A few factors could possibly improve the results. First is the assumption that the b and y-ion peaks themselves are mono-isotopic and thus the real complimentary peaks should add up to the mono-isotopic parent mass itself. This assumption might not be correct. We test the hypothesis that the fragment themselves can be shifted by allowing a sufficiently large error range of [-4.0, 4.0] Da in the mass of each real fragment. Doing this, we find the distribution of parent masses formed by the summation of real complimentary peaks given in Figure A.2.

The second assumption that the mass bin which contains the highest number of complimentary pairs will be the correct parent mass bin, might not be true. Isotopic shifts of the individual b and y peaks as explained above, coupled with noise may cause mass bins outside of the parent mass bin to have a higher complimentary pair count. This is clearly seen in Figure A.2 where the bins to the right and left of the parent mass bin has a higher complimentary peaks count.

The observation that the fragment peaks themselves might not be mono-isotopic and that the actual parent mass bin itself might not have the highest complimentary pair count presents us with a possible way of improving the self-convolution method. Instead of finding the mass-bin with the most number of complimentary peak pairs, we instead find the *"score"* of the

mass-bins within a window of size $[-2.5, 2.5]$ Da centered around the candidate mono-isotopic parent mass bin, and choose the top few mass bins in this way.

Instead of just relying on a simple complimentary pair count, a way of scoring the mass bins within the window of a candidate parent mass bin is to use properties related to the complimentary peaks in the mass bins within the window. In our experiments we have tried including all the following properties:

Assuming a bin size of 0.5 Da, a window is of size n (where n is the number of bins and is always an odd number) centered around the the candidate parent mass bin $parbin$,

1. Experimental parent mass range – 0-1000,1000-2000,2000-3000,3000-4000,4000-5000
2. Number of complimentary peaks/fragmentation points in bin_x where $x \in [parbin - 5, parbin + 5]$
3. Number of y-ion peaks at bin_x with intensity level = y for $y \in [1, 6]$ where
   (a) y = 1 is for peaks at intensity rank 1-10
   (b) y = 2 is for peaks at intensity rank 11-20
   (c) y = 3 is for peaks at intensity rank 21-30
   (d) y = 4 is for peaks at intensity rank 31-40
   (e) y = 5 is for peaks at intensity rank 41-50
   (f) y= 6 is when there are no peaks in bin_x
4. Number of y-ion peaks at bin_x with intensity level = y representing SRM fragmentation points = z for $z \in [1, 5]$ where
   (a) z = 1 is for SRM fragmentation mass between 0-500 Da
   (b) z = 2 is for SRM fragmentation mass between 500-1000 Da
   (c) z = 3 is for SRM fragmentation mass between 1000-1500 Da
   (d) z = 4 is for SRM fragmentation mass between 1000-2000 Da
   (e) z = 5 is for SRM fragmentation mass between 2000-2500 Da

ISB2	Top1	Top1-2	Top1-3
% of correct prediction	0.59	0.78	0.87

Table A.3: % of corrected parent masses for ISB2 using LogitBoost. From the table, we see that using only the Top1 result (mass bin with the most number of complimentary peaks) has improved from Self-Convolutions 2.0's 23% to 59%.

5. Same as 3. But for b-ion peaks

6. Same as 4. But for b-ion peaks and fragmentation points representing PRM

7. Number of fragmentation points in bin_x that have a consecutive fragmentation point in bin_x (that is forming a mass difference corresponding to an amino acid mass)

8. Number of fragmentation points in bin_x without a consecutive fragmentation point in bin_x.

Having determined the above attributes, we then made use of **LogitBoost** Friedman et al. [23]a boosting classifier which is basically an ensemble machine learner. The advantage of using LogitBoost is that it is resistant to redundant attributes, in fact making use of all possibly good attributes in building the final classifier, thus there is no necessity for attribute selection. Secondly it is an ensemble machine learner where it iteratively builds weak learners based on some of the attributes and adds them to the final strong classifier, thus making them akin to decision forests which have been shown to be superior to single decision trees Gashler et al. [24]. Finally, instead of simply predicting a class label, it gives a score representing the confidence of the given sample being in the predicted class. This is useful for our case since we can sort the confidence score for each of the candidate parent masses and pick the top few.

Result of using the boosting classifier for parent mass correction on the test data is show in Table A.3. LogitBoost was run for 50 iterations on the training data (ISB2 and GPM) where the accuracies asymptotes. Since we are only concerned about the score of the positive class (the correct parent mass bin) on each data, we sort said scores and picked the Top1-3 mass bins.

ISB2	Top1	Top1-2	Top1-3
% of correct prediction	0.63	0.79	0.90

Table A.4: % of corrected parent masses for ISB2 using LogitBoost with improved attributes. From the table, we see that using only the Top1 result (mass bin with the most number of complimentary peaks) has improved from the previous attempt from 59% to 63% (an improvement of 4%). There are also slight improvement for both Top1-2 and Top1-3 (both slightly better than using Self-Convolution 2.0).

GPM	Top1	Top1-2	Top1-3
% of correct prediction	0.38	0.52	0.57

Table A.5: % of corrected parent masses for GPM using LogitBoost with improved attributes. From the table, we see that using only the Top1 result (mass bin with the most number of complimentary peaks) only corrects 38% of the parent masses. There is not as much improvement from Top1-2 to Top1-3 (5%) as compared to the ISB2 data.

A.1.4 Improvement to Attributes

In our prediction, we have a fixed number of possible parent mass bins to use. However, in our attributes we do not capture the relationship among these parent mass bins. A possible improvement can be obtained by using the rank of a parent mass bin compared to the other potential mass bins for each of the attributes listed, instead of simply using a number that is only associated with the bin itself. In this improvement, we first compute the attributes as before, then sort the mass bins in non-descending order of the attribute value. Each attribute of each mass bin will then take the rank instead of the attribute value itself. Using this change, Table A.4 below shows the experimental results.

Results of parent mass correction for GPM data is given in Table A.5. The results are not as good as for ISB2 (Top1 result only corrected 38% of the parent masses). This is in line with the fact that GPM has a wider range of parent mass shifts (due to PTMs) and these are hard to correct for.

Appendix B

B.1 Analysis of Probability of Observation of Mono-Chromatic Tag of length $\geq l$

Given an ion-type δ with a probability of observation q, let r_l be the probability of observation of a mono-chromatic tag of length $\geq l$, where $l \geq 1$ can be explained as the canonical peptide ρ being fragmented in such a way that $\geq l+1$ consecutive fragmentation points of ρ generated ions of type δ. This probability can be analyzed as follows.

Assuming that ρ on average has t fragmentation points, the probability of δ explaining $l+1$ of these points is q^{l+1}. There are $\binom{t}{l+1}$ number of ways to pick $l+1$ positions. The probability of picking exactly one of these combination is then $\frac{1}{\binom{t}{l+1}}$ and the probability of δ explaining such a combination is then $\frac{q^{l+1}}{\binom{t}{l+1}}$. Out of the combinations of $l+1$ positions, $t-l$ of them have the positions consecutive to each other (a tag). Therefore the probability of picking either one of these combination and having δ explain it is $\frac{q^{l+1}*(t-l)}{\binom{t}{l+1}}$, which is $\frac{q^{l+1}*(t-l)*(l+1)!*(t-(l+1))!}{t!}$. This is exactly the probability of finding a mono-chromatic tag of δ of length $= l$. The probability r_l of finding a mono-chromatic tag of δ of length $\geq l$ is then defined by the function

$$r_l = \sum_{x=l+1}^{x=t} \frac{q^{l+1} * (t-(x-1)) * (x)! * (t-(x))!}{t!}$$

We note that the value of the function is dominated by the first term that is $x = l+1$ and the last term $x = t$, since for each of these terms, the numerator has a factorial that cancels or almost cancels the denominator. In fact the values of the term drops as x goes further away

from $l+1$ and rises again as it approaches t. On average the length t of a peptide is around 15, and thus $t!$ is a huge value. We can practically ignore all terms except for $x = l+1$, and $x = t$. Thus we can simply the function to

$$r_l = \frac{(q^{l+1} * (t-(l)) * (l+1)! * (t-(l+1))!)}{t!} + q^t$$

Now for For δ with $q = 0.1$ (a rare ion-type), the probability r_1 of observation of a mono-chromatic tag of length at least 1 is ≈ 0.0013. As q decreases or l increases, this probability decreases as well (exponentially as l increases). Thus we see that for rare ion-types, it is highly unlikely to see mono-chromatic tags of any length.