Algorithms for Peptide and PTM Identification using Tandem Mass Spectrometry

Kang Ning

A DISSERTATION SUBMITTED
FOR THE DEGREE OF
DOCTOR of PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE, SINGAPORE

2007

To my wife Bai Hong, mother and father

You all deserve the pride!

Acknowledgements

I would like to first thank my family, especially my parents and my wife, for their endless support every day, month and year during my pursuit of PhD.

I would like to take this opportunity to thank Prof. Leong Hon Wai for his patience, constant guidance and countless insightful suggestions throughout my entire PhD candidature. He is a great supervisor, who not only supervises me on research projects and research methodologies (授之与渔), but also teaches me the principles of being a right man. He is also a gentleman, allowing me to initiate many interesting research projects on my own, and provided assistance when I needed it. These virtues will be inherited in me, and help me in my whole life.

I would also like to thank Prof. Zhang Louxin for his great guidance on many projects, and for inspiring me in research, as well as setting a role model for doing careful and thoughtful research. His influence on me will be priceless to my future career and life.

I would also wish thank my friends, especially Dr. Chua Hon Nian; as well as alumni and current members of the RAS group leaded by Prof. Leong Hon Wai. And I am also grateful to many collaborators that co-operated with me during my PhD candidature.

Table of Contents

Summary

This dissertation focuses on my work in the analysis of biological sequences, with special concentration on algorithms for peptide and PTM identification using tandem mass spectrometry.

The main concern for algorithms in peptide identification is achieving fast and accurate peptide identification by mass spectrometry. The main results of this study is a set of database search and *De Novo* algorithms for peptide identification based on "extended spectrum graph" and machine learning techniques such as SOM.

I have designed a set of heuristic algorithms for identification of peptide sequences from mass spectrometry, with focus on multi-charge spectrum. I have first introduced and analyzed the extended spectrum graph computational model. Based on this model, I have defined the "best strong tags" which are highly accurate. Then I have proposed the GBST algorithm based on best strong tags. After this, I have extended the best strong tags to "multi-charge strong tags", and proposed the GMST and GST-SPC algorithms. The GST-SPC algorithm is also based on computing the SPC of the candidate sequences and experimental spectrum. A fast database search algorithm, PSP, is also proposed based on multi-charge strong tags.

Then I have described peptide identification algorithms that are based on transformation of spectra to high dimensional vectors. Using the SOM and MPRQ technique, these algorithms then transformed the peptide sequence similarity to 2D point similarity on SOM map, and performed multiple simultaneous queries for candidate peptides

efficiently. The first algorithm, PepSOM, empirically proved the effectiveness of using SOM and MPRQ for efficient peptide identification. The second algorithm further improved PepSOM by scoring and ranking the candidate peptides by comparing them with tags generated by GST-SPC algorithm. The improved version of this algorithm, the TagSOM algorithm, went further by using the information contained in these candidate peptides and tags for the purpose of PTM identification.

These algorithms are fast and accurate, especially when compared to other algorithms on multi-charge spectra. Some of these algorithms can also detect post translational modifications (PTMs) in spectra with high accuracy.

I have also performed research on the analysis of multiple sequences. These researches include the analysis of Longest Common Subsequence (LCS) and Shortest Common Supersequence (SCS) of multiple sequences based on multiple alphabets.

List of Figures

List of Tables

Chapter 1

Introduction

People have been wondering about the complex nature of living beings on this planet from ancient times. The advance in biology science has little by little fed our curiosity, and this process is accelerated after the invention of computers. In the past few years, more and more computational methods have been used on large scale analysis of biological units (based on molecules) of every living being. This latest development of computational analysis of biological systems has given birth to the new era of bioinformatics.

Bioinformatics is a science that refers to the creation and advancement of algorithms, computational, statistical techniques, and theory to solve formal and practical problems inspired from the management and analysis of biological data. In bioinformatics, we bioinformaticians are provided with a huge amount of raw data that are generated by various experiments on different biological samples. Bioinformaticians have to (a) identify and analyze these samples, and from them, (b) discover complex relationships between them. In this process, we aim to ultimately understand Life itself.

Biological sequences are critical in bioinformatics. Since biological sequences are the basis for other biological units, the analysis of biological sequences is fundamental to virtually every aspect of bioinformatics. Gusfield [1] wrote:

> "*The area of approximate matching and sequence comparison is central in computational molecular biology both because of the presence of errors in*

molecular data and because of active mutational processes that sequence comparison methods seek to model and reveal."

This dissertation concentrates on analysis of biological sequences, with special focus on *algorithms for peptide sequence identification by mass spectrometry*. Traditionally, there are two classes of algorithms for for peptide identification by mass spectrometry problem aim to identify peptide sequences from high-throughput mass spectra data – *database search algorithms* and *de novo sequencing algorithms*. They are useful to biologists to verify known peptides or to discover new peptides [3, 4, 22-24, 30, 32, 33]. The algorithms that I have designed in this dissertation are both accurate and efficient, with superior performance on multi-charge spectra. In addition, I have also carried out research in heuristic algorithms for multiple sequence analysis and algorithms for some other problems related to sequences analysis [20, 28, 29, 31, 34].

1.1 Peptide identification problem

Peptide identification from mass spectrometry is important, since it provides data for further research such as protein sequence analysis. However, while high-throughput spectrometers have generated a huge number of spectra, peptide identification algorithms are slow and inaccurate. I have analyzed and designed efficient and accurate algorithms for peptide identification problems.

1.1.1 Algorithms Based on Tags

I have designed *De Novo* peptide identification algorithms that are based on **multi-charge strong tags**. The simple algorithm **GBST, which** only utilized the "best strong tags" on **extended spectrum graph,** showed that considering multi-charges in multi-

charge spectrum can help to improve identification accuracies [2, 3]. The improved **GST-SPC** algorithm not only use multi-charge strong tags (**GMST** algorithm), but also **optimize SPC**, so that it has improved accuracies [4]. Further improvement includes a better **preprocess** computational model and a better computational model for **anti-symmetric problem** [5]. These new models can also be applied on other *De Novo* algorithms to improve their accuracies.

Based on "best strong tags", I have also designed an efficient database search algorithm (**PSP**) for peptide identification [6]. The algorithm is based on **linear time pattern matching strategy which allows mismatches**, so it is both accurate and fast.

These projects have utilized the information in multi-charge spectra that have not been investigated before. The algorithms that I have proposed for these problems have improved the peptide identification accuracies.

1.1.2 Algorithms Based on Tags, SOM and MPRQ

Apart from peptide identification algorithm only based on tags, I have also designed peptide identification algorithms based on **transforming both experimental and theoretical spectra to high-dimensional vectors**. These vectors are then transformed to 2D points on plane, followed by **SOM and MPRQ** query to quickly get the candidate peptides. These candidate peptides are then **validated by comparing with tags and experimental spectrum** for accurate peptide identification. In this way, no spectrum comparison is needed, while the spectrum similarity is preserved through vector similarity and neighborhood relationships between points on the 2D plane.

The first attempt (**PepSOM**) by us involves **binning** the spectra according to mass/charge values to get vectors, and using SOM and MPRQ techniques to get candidate peptide sequences. This is followed by SPC for validation, and the results are already quite accurate [7]. Subsequently we proposed an improved algorithm that used SPC together with multi-charge strong tags for candidates' validation, and also incorporated a module in this algorithm to **identify Post Translational Modifications (PTMs)**. Results are satisfactory on real spectra with real PTMs [8]. Furthermore, we have recently designed a novel algorithm (**TagSOM**) that **used biologically meaningful features to transform spectra to vectors**, as well as **an improved scoring function** in the validation stage to identify PTMs. The peptide and PTM identification accuracies are expected to be further improved [9].

These projects have empirically proved the effectiveness of peptide identification by transforming spectra to vectors in high-dimensional space using spectrum features. The advantage of these set of algorithms is accurate identification of peptides and PTMs, and show the power of combination of tags, SOM and MPRQ techniques for peptide and PTM identifications.

The overall outline of my PhD dissertation is illustrated in Figure 1.

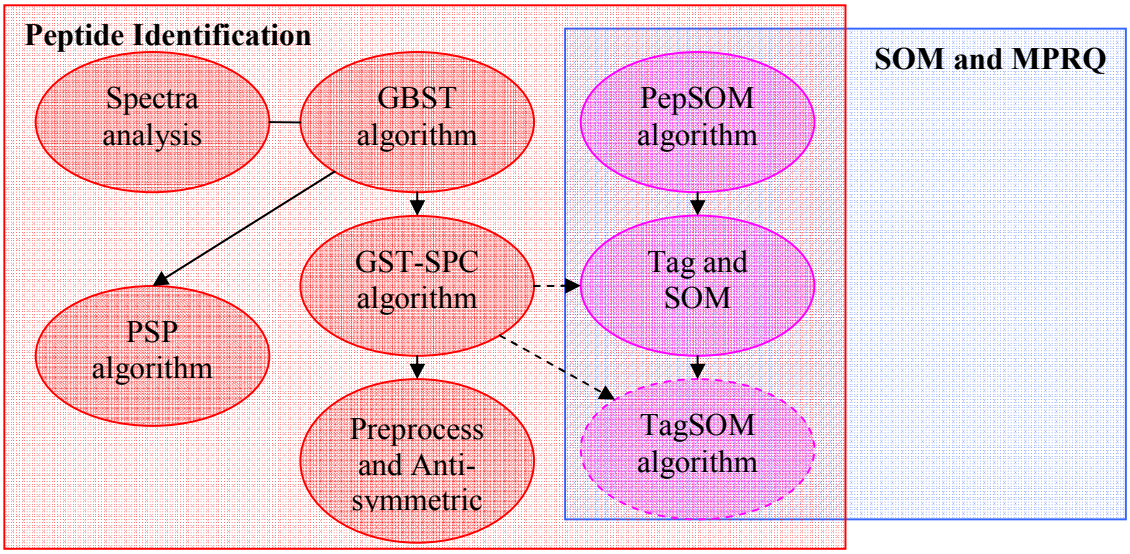


Figure 1. The illustrated outline of my PhD dissertation. Solid arrows indicate "improvement" or "extension" relationships; dashed arrows indicate "using results of" relationships; and lines with no arrows indicate "highly related subjects" relationships. Solid ovals indicate "completed" projects, while dashed ones indicate projects "in progress".

1.2 Multiple sequences analysis

In addition to peptide identification, I have also performed research on multiple sequences analysis. Given a great amount of biological sequences, I have analyzed the common properties of these sequences, and designed a set of heuristic algorithms to compare them and discover their common parts, namely, their Longest Common Subsequence (LCS), Shortest common Supersequence (SCS) and patterns [20, 28, 29, 31, 34]. The heuristic algorithms that I have designed are superior to other algorithms in both the quality of the results and computational time, especially for many long sequences. Since these are not the focus of this dissertation, I will not go into details of these research, but a summary of these results can be found in **Appendix A**.

Chapter 2

Survey of Peptide Identification Problems and Algorithms

Proteomics is the large-scale study of proteins, particularly their sequences, structures and functions. In proteomics, the identification of peptide sequences is very important. This is because: (i) we do not know the full set of proteins that cells produce; (ii) it is important to identify which specific proteins interact in a biological system; and (iii) it is important to identify proteins that are present in biological tissues under different conditions. Currently, peptide identification is mainly done on spectra data generated by mass spectrometry (MS) or tandem mass spectrometry (MS/MS).

The advance in tandem mass spectrometry (MS/MS) technology has made high-throughput mass spectra generation possible. A protein can be digested into peptides by proteases such as trypsin. In a very short time, a tandem mass spectrometer breaks a peptide into smaller fragments, and measures the mass/charge ratio of each. The *mass spectrum* of a peptide is a collection of mass/charge ratios of these fragments.

In an ideal fragmentation process, where every fragment of a peptide is generated in an ideal mass spectrometer, the peptide identification problem is simple. However, peptide identification is a non-trivial problem because these ideal conditions are never met in experiments. The spectrum obtained from MS/MS usually contains a lot of noise, introduced by impurities in the peptide sample, and biases inherent in mass spectrometers. The existence of PTMs further complicates the problem [10]. Post Translational Modifications (PTMs) are chemical modifications to a protein after its translation. This

makes the problem becomes more difficult since a known peptide sequencemay not exactly match the actual peptide fragments used to generate the spectrum.

There are two types of computational problems in peptide identification. The first type of problem, which we refer to the problem as peptide identification, are algorithms that identify peptide sequences in database. The second type of problem, which we refer to as *De Novo* peptide sequencing, is the interpretation of peptide sequences in cases when peptide sequences are either not present in database, or different from canonical form present in a database (such as with post-translational modifications).

2.1 Problem Statement

2.1.1 Peptide Identification Problem

To introduce the *peptide identification problem*, we first define some general terms. In tandem mass spectrometry (MS/MS), a peptide sequence $\rho = (a_1a_2\ldots a_l)$ is fragmented into a spectrum *S*. The parent mass of the peptide ρ is given by $M = m(\rho) = \sum_{j=1}^{l} m(a_j)$. A peptide prefix fragment is $\rho_k = (a_1a_2\ldots a_k)$, for $k \leq l$, and has mass $m(\rho_k) = \sum_{j=1}^{k} m(a_j)$. Suffix masses are defined similarly. We always express a fragment mass in experimental spectrum using its *PRM* (prefix residue mass) representation, which is the mass of the prefix fragment. In mathematical notation, given a fragment ρ_k with mass $m(\rho_k)$, we define $PRM(\rho_k) = m(\rho_k)$ if ρ_k is a prefix fragment. Similarly, we define $PRM(\rho_k) = M - m(\rho_k)$ if ρ_k is a suffix fragment ({*y*-ion}). By calculating the *PRMs* for all fragments, we can treat all fragment masses uniformly.

A spectrum S is composed of many peaks. Each of the peaks p_i is represented by its *intensity*(p_i) and mass-to-charge ratio *mz*(p_i). If peak p_i is not noise, then it represents a fragment ion of ρ. Each peak p_i can be characterized by the ion-type, specified by (z, t, h) $\in (\Delta_z \times \Delta_t \times \Delta_h) = \Delta$, 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 ρ_k (prefix or suffix fragment) will produce an observed peak p_i in the experimental spectrum S that has a mass-to-charge ratio of *mz*(p_i) and intensity *int*(p_i). The mass of ρ_k, $m(\rho_k)$ can be computed using a shifting function, *Shift*, defined as follows:

$$m(\rho_k) = Shift(p_i, (z,t,h)) = mz(p_i) \cdot z + (\delta(t) + \delta(h)) - (z-1) \quad (1)$$

where $\delta(t)$ and $\delta(h)$ are the mass differences associated with the ion-type t and the neutral loss h, respectively. We say that peak p_i is a *support peak* for the fragment ρ_k and we say that the fragment ρ_k is supported by the peak p_i. A peak p_j is a *support peak* for the peak p_i if both of them are support peaks for the same fragment ρ_k.

In the problem of peptide identification by tandem mass spectrometry, the input includes the mass spectrum S, the set of possible ion types Δ and the parent mass M (and for database search algorithms, a database of peptides). The output is the putative peptide sequence P that matches with S better than any other peptides.

2.1.2 Extended Spectrum Graph

The *match* between a peptide and an experimental spectrum is always represented by the number of common peaks between the theoretical spectrum of P and the experimental spectrum S. This is often referred to as the *shared peaks count* (SPC). In reality, peptide identification algorithms use more complicated scoring function than SPC.

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

Extended Spectrum: Conversely, the *real* peaks (in contrast to noise) in an experimental spectrum $S = \{p_1, p_2, \ldots p_n\}$ of maximum charge α, may have come from different ion-type of different fragments (may be prefix or suffix fragment, depending on the ion-type). We do not know, *a priori*, the ion-type $(z, t, h) \in \Delta$ of each peak p_i, we can not even distinguish *real* peaks from noise. Therefore, We "extend" each peak p_i by generating a set of $|\Delta|$ pseudo-peaks (or guesses), one for each of the different ion-types $(z, t, h) \in \Delta$. More precisely, in the extended spectrum S_α^α, for each peak $p_i \in S$ and an ion-type $(z, t, h) \in \Delta$, we generate a pseudo-peak, denoted by $(p_i, (z, t, h))$, with an "assumed" (uncharged) fragment mass computed using the *Shift* function (1). Only one of these pseudo-peaks can be a *real* peak, while the others are "introduced" noise.

An example of an extended spectrum is illustrated in Figure 2. For simplicity, we only consider ion-types $\Delta_t = \{b$-ions, y-ions$\}$ and $\Delta_h = \{\emptyset\}$. The figure depicts the extended spectrum for a peptide ρ = GAPWN with parent mass $M = m(\rho) = 525.2$, and an experimental spectrum $S = \{113.6, 412.2, 487.2\}$ with maximum charge 2. The first peak "113.6" is a (2, b-ion, Ø)-ion of the prefix fragment GAP; the peak 412.2 is a (1, b-ion, Ø))-ion of the prefix fragment GAPW; and "487.2" is a (1, y-ion, Ø)-ion for the fragment

G. In Figure 2 (a), only charge 1 is considered and S_1^2 = {112, 430, 411, 132, 486, 57}. The entries in the table are the *PRM* values. For example, the possible fragment masses of 112 and 430 correspond to the extension of the first peak for ion-types (1, *b*-ion, Ø) and (1, *y*-ion, Ø), respectively. However, if charge 2 is also considered, then S_2^2 = {112, 430, 225, 31, 411, 132, 486, 57} as shown in Figure 2 (b).

Modeling Current *De Novo* Algorithms: To take into account the fact that some algorithms consider only ion-types of charge up to β (usually β = 2), we extend the definition to $TS_\beta^\alpha(\rho)$ which is defined to be the subset of $TS_\alpha^\alpha(\rho)$ for which the charge $z \in \{1,2,\ldots, \beta\}$. The case β=1 reflects the assumption that all peaks are of charge 1, and makes use of the extended spectrum S_1^α. Algorithms such as PepNovo and Lutefisk work with a subset of the extended spectrum S_2^α, even for spectra with charge $\alpha > 2$. In general, $TS_\beta^\alpha(\rho)$ does not account for peaks that correspond to ion-types with higher charges $z=\beta+1, \ldots, \alpha$ ($\alpha > \beta$). Since $TS_1^\alpha(\rho) \subseteq TS_2^\alpha(\rho) \ldots \subseteq TS_\alpha^\alpha(\rho)$, higher accuracy can be attained when higher charge values are taken into account.

The Extended Spectrum Graph: We also introduce the extended spectrum graph, denoted by $G_d(S_\beta^\alpha)$, where *d* is the "connectivity". Each vertex *v* 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. Thus v = (p_i, (z, t, h)). Therefore, 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 mass 0 and the end vertex v_M corresponding to the parent mass *M*.

In the "standard" spectrum graph, we have a directed edge (u, v) from vertex u to vertex v if $PRM(v)$ is larger than $PRM(u)$ by the mass of a single amino acid. In the extended spectrum graph of connectivity d, $G_d(S_\beta^\alpha)$, we extend the edge definition to mean "*a directed path of no more than d amino acids*". Thus, we connect vertex u and vertex v by a directed edge (u, v) if $PRM(v)$ 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 20^d and increased exponentially with d. In this dissertation, I use d=2, unless otherwise stated.

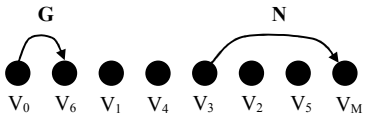
Two extended spectrum graphs (with d=2) are shown in Figure 2. The spectrum graph $G_2(S_1^2)$ is shown in Figure 2 (c). We can see that only the edges (v_0, v_6) for amino acid G and (v_3, v_M) for amino acid N can be obtained. The subsequence APW is more than 2 amino acids long and so $G_2(S_1^2)$ is unable to elucidate this information. By considering S_2^2 (in (a) and (b)), we obtain the graph $G_2(S_2^2)$ shown in (d). New edges can be obtained: edge (v_6, v_7) for path AP of length 2 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. However we also note that more noise may be introduced in $G_2(S_2^2)$, which can result in the formation of fictitious edges . One example is shown in (d) using dashed line to denote the fictitious edge (v_4, v_8). Many such fictitious edges can result in fictitious paths from v_0 to v_M, thus yielding a higher rate of false positives.

z	$mz(p_1)$= 113.6		$mz(p_2)$= 412.2		$mz(p_3)$=487.2	
	B	***Y***	***B***	***Y***	***B***	***Y***
1	V_1 112.6	V_2 430.6	V_3 411.2	V_4 132	V_5 486.2	V_6 57

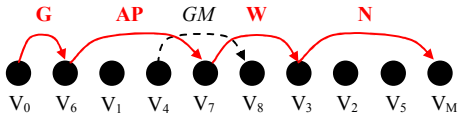
(a) The spectrum S_1^2 (only B and Y ions considered)

z	$mz(p_1)$= 113.6		$mz(p_2)$= 412.2		$mz(p_3)$=487.2	
	B	***Y***	***B***	***Y***	***B***	***Y***
2	V_7 225.2	V_8 318	- -	- -	- -	- -

(b) Extending the peaks for charge 2 ions.



(c) The spectrum graph $G_2(S_1^2)$



(d) The extended spectrum graph $G_2(S_2^2)$

Figure 2. Example of extended spectrum graph for mass spectrum generated from peptide "GAPWN".

2.2 Peptide identification algorithms

Approaches for peptide identification can be categorized into database search algorithms [9, 35, 37, 40], *De Novo* algorithms [1, 2, 8, 10, 15, 16, 38, 39] and combined algorithms [11-14]. Database search algorithms usually return the peptide sequences that match the parent mass of the experimental spectrum via some scoring functions. Apparently, the accuracy of these approaches depends largely on the completeness of the database, and the process is slow (usually at least a few minutes). An analysis of an LC/LC/MS/MS experimental dataset using the popular BioWorks program by ThermoFinnigan on a computer with a single processor typically takes several hours (approximately 30,000 scans against the *Escherichia coli* database).

Moreover, the accuracy of these methods are generally mediocre for peptide sequences not available in database (i.e. peptides not already known), as well as for peptides with PTMs. For such peptide sequences, *De Novo* algorithms are the methods of choice. These algorithms interpret peptide sequences from spectrum data purely by analyzing the intensity and correlation of the peaks in the spectrum. They can identify tags (highly

reliable fragments) with high accuracy [15], and the process is fast (always within one minute), but their performance deteriorates quickly with the presence of noise and PTMs.

2.2.1 Database Search Algorithms

Database searching algorithms [9, 35, 40] for peptide identification by mass spectrometry rely primarily on good scoring. The peptide that scores the highest or has a lowest p-value is the one that best explains the spectrum. The success of these algorithms relies on the completeness of peptide databases, and the selection of an appropriate scoring mechanism.

Database search in mass-spectrometry has been investigated by many researchers [9, 35, 40]. Database search algorithms exhibit good performance in the identification of peptides already in the peptide database. However, these algorithms rely heavily on the presence of the target peptide (or similar ones) in the protein database. Generally, these algorithms search a sequence database for peptide sequences which would produce ions of the mass observed for a particular spectrum, then score these candidate sequences against the observed spectrum.

Traditional database search algorithms are established on a common principle: the experimental spectrum is compared with the theoretical spectrum for each of the peptide in the database, and the peptide from the database with best match is likely to match the sequence of the experimental spectrum. The most widely used database search algorithms for analyzing the mass spectra of peptides includes the software Sequest [16, 17]. Sequest extracts a list of sequences that match the experimentally determined peptide mass from the database. The best match between the spectrum and the database-derived peptide

sequences is made via a combination of an ion intensity-based score plus a cross-correlation routine. The main advantage of this approach is that it is highly automated and requires little human intervention. Its disadvantage lies in its inability to make non-identical matches between query peptides and database homologs due to the use of the peptide-mass pre-filter,.

One problem with these algorithms is that they only compared the ions of the mass observed for a particular spectrum against the peptide, so they can work well for peptide sequences already in the database, but perform badly for spectrum with noise and peptides with post-translational modifications (PTMs).

2.2.2 De Novo Algorithms

De Novo algorithms [1, 2, 8, 10, 38, 39] are used to predict sequences or partial sequences for novel peptides or for peptides that are not found in the protein database. Many *De Novo* sequencing algorithms [8, 10, 38, 39] uses a spectrum graph approach to reduce the search space of possible solutions. Given a mass spectrum, the spectrum graph [18] is a graph where each vertex corresponds to some ion type interpretation of a peak in the spectrum. Edges represent amino acids which can interpret the mass difference between two vertices. Each vertex in this spectrum graph is then scored using some scoring function (i.e., Dancik scoring) based on its supporting peaks in the spectrum (see [18] for details). Given such a scoring function the predicted peptide represents the optimal weighted path from the source vertex v_0 (of mass 0) to the end vertex v_M (of mass M).

PepNovo [19] uses a spectrum graph approach similar to [18], but uses an improved scoring function based on a probability network of different factors which affect the peptide fragmentation and how they conditionally affect each other (represented by edges from one vertex to another). The PEAKS algorithm [20] does not explicitly construct a spectrum graph but builds up an optimal solution by finding the best pair of prefix and suffix masses for peptides of small masses until the mass of the actual peptide is reached. A fast dynamic programming algorithm is then used in PEAKS for peptide identification.

2.2.3 Combined Algorithms

For database search algorithms, it is well known that it is almost impossible to find a peptide whose theoretical spectrum *matches exactly* (100% match) with the experimental spectrum. However, *De Novo* algorithms can only output highly reliable peptide fragments (tags). Therefore, for the sake of accuracy and completeness of the results, many algorithms rely on matching peptides with much shorter and reliable *tags* [11, 14] generated from spectrum by *De Novo* algorithms.

In [11], tags are used for the search of peptide sequences. A fragmentation spectrum usually contains a short, easily identifiable series of sequence ions, which yields a partial sequence (tag). This partial sequence divides the peptide into three parts - regions 1, 2, and 3 - characterized by the added mass m_1 of region 1, the partial sequence of region 2, and the added mass m_3 of region 3. The construct, *m_1 partial sequence m_3*, is called a "peptide sequence tag" and it is a highly specific identifier of the peptide. The algorithm then uses the sequence tag to find the peptide in a sequence database. The main problem of this approach is that the model used in this algorithm is too simple. A 3-segment

peptide sequence tag is used, but can not utilize more than one highly-confident fragment. The database search may return several candidate peptide sequences, but further discriminations are very limited.

Recently there are some research interests on this issue that combine database search with *De Novo* techniques [12, 14]. The GutenTAG algorithm [12] automates the process of inferring "partial sequence tags" directly from the spectrum and efficiently examines a sequence database for peptides that match some of these tags. When multiple candidate sequences result from the database search, the algorithm evaluates the best match by a rapid examination of spectral fragment ions. More recently, the InsPecT [21] algorithm is proposed, which first generates a set of highly accurate tags from spectrum, and then use these tags to filter peptide sequences in database. Because *De Novo* is imperfect, multiple tags are produced for each spectrum to ensure that at least one tag is correct. The accuracy of this algorithm depends on the quality of the tags but even in the context of up to a dozen modifications, they perform reasonably well. Another interesting aspect of InsPecT is that it uses automata to search for peptide sequences in linear time. For a batch of spectrum data, the process can be very quick (about 10 ms per spectrum). Another database search algorithm based on a set of tags is SPIDER [22]. However, based on our analysis [2, 3], these algorithms still have a lot of room for improvement.

2.2.4 Our algorithms

I have worked on peptide identification problem, and proposed algorithms for accurate and fast peptide sequence identification. Essentially, there are two categories of peptide identification algorithms examined, the first category centered on *De Novo* and database

search algorithms based on tags, while the second category of algorithms are based on tags, SOM and MPRQ techniques for peptide and PTM identification.

Chapter 3

Peptide Identification Algorithms Based on Tags

In this section, I will focus on a series of projects on peptide identification algorithms based on tags, with special concern on multi-charge spectra. I will first introduce the notation of strong tags in the context of extended spectrum graph. Based on the extended spectrum graph and strong tags, I will then describe our analysis of the characteristics of multi-charge spectrum datasets. Then I will introduce the *De Novo* algorithm, GBST, for peptide identification (sequencing) from multi-charge spectrum based on "best strong tags". Next I will extend the "best strong tags" to "maximal multi-charge strong tags", and proposed the GMST and GST-SPC algorithms. I have also designed a database search algorithm, PSP algorithm, based on patterns generated by a set of "best strong tags", for peptide identification. Finally, I will touch on two issues in peptide identification; namely, preprocessing to remove noise, and the anti-symmetric problem. I have also proposed new computational models to address these issues, which can further improve accuracy in peptide identification.

3.1 Brief Review and my work

Multi-charge spectra are spectra with parent charge larger than 1. Because of the vast use of electrospray source in mass spectrometry, multi-charge spectra data are very abundant. However, the analysis of these multi-charge spectra is rare. Most peptide sequencing algorithms currently handle spectra of charge 1 or 2 and have not been designed to handle multi-charge spectra. PEAKS [20] perform a conversion of multi-charge peaks to their single-charge equivalent before sequencing. Lutefisk [23] works with single-charge ion

only, while Sherenga [18] and PepNovo [19] works with single- and double-charge ions. In [2, 3], we have analyzed the characteristics of multi-charge spectrum data. We proposed a characterization of multi-charge spectra by generalizing existing models. Using these new models, we analyzed spectra with charges 1-5 from the GPM datasets. Our analysis shows that higher charge peaks are present and they contribute significantly to the prediction of the complete peptide. They also help to explain why existing algorithms do not perform well on multi-charge spectra.

Based on these analyses, we proposed a novel *De Novo* algorithm (GBST) for dealing with multi-charge spectra based on tags in the context of extended spectrum graph models. Experimental results show that it performs well on all spectra, especially so for multi-charge spectra.

In [4], we analyzed current *De Novo* algorithms, and proposed a novel algorithm (GST-SPC) for peptide sequencing. In this project, we have analyzed some of the shortcomings of GBST. We also present a new algorithm GST-SPC, by extending the GBST algorithm in two directions. First, we use a larger set of multi-charge strong tags and show that this improves the theoretical upper bound on performance. Second, we proposed an algorithm that finds a peptide sequence which is optimal with respect to *shared peaks count* (*SPC*) from among all sequences that are derived from strong tags. Experimental results demonstrate the improvement of GST-SPC over GBST and other *De Novo* algorithms for multi-charge mass spectra.

In [6], we proposed a database search algorithm for peptide identification. The Peptide Sequence Pattern (PSP) algorithm first generates the peptide sequence patterns (PSPs) by connecting the strong tags with mass differences. A linear time database search process is then used to search for candidate peptide sequences by PSPs, and the candidate peptide sequences are then scored by shared peaks count (SPC). The PSP algorithm is designed for peptide identification from multi-charge spectra, but it is also applicable for single-charge spectra. Experiments have shown that the PSP algorithm can obtain better identification results than some current database search algorithms on many multi-charge spectra; and also obtain comparative results on single-charge spectra against these algorithms.

I also noticed that although peptide sequencing problem is extensively investigated by researchers recently, and peptide sequencing results are becoming more accurate, many of these algorithms are using computational models based on some assumptions, and these unverified assumptions may be the obstacles for further improvement.

In [24, 25], I first investigated the simple model for peptide sequencing without preprocessing the spectrum, and I have shown that by introducing preprocessing to remove noise in spectrum, the peptide sequencing can be faster, easier and more accurate. I then investigated one of the most important assumptions, the anti-symmetric assumption in the peptide sequencing problem. From my studies, I have proven empirically that approached that do not consider anti-symmetry or simply remove anti-symmetric instances may be oversimplifying the peptide sequencing problem. I then proposed a more realistic model that takes anti-symmetry into account. I also proposed a novel

algorithm which incorporate preprocessing and the new model, and showed though experiments that this algorithm can achieve further improvement in performance.

3.2 Strong Tags

Tandem mass spectrum data analysis shows that peaks in many mass spectra can be grouped into closely-related sets, especially when the peptide is multi-charge. Within each set, the peaks can be interpreted as the same ion type (*b*-ions or *y*-ions), and the mass differences between "successive" peaks are such that they can form ladders (partial sequences). An example is shown in Figure 3, where we have computed the theoretical spectrum (the table) and the peaks from an experimental spectrum *S* are shown in bold. Several peaks are grouped together into ladders of *y*-ions and *b*-ions of charge 1.

bond	$^{+1}$y	$^{+1}$y*	$^{+1}$b	$^{+1}$b*
S1	1807.0	1790.0	130.0	113.0
I2	1693.9	1676.9	243.1	226.1
R3	1537.8	1520.8	**399.2**	382.2
V4	1438.8	1421.7	**498.3**	481.3
T5	1337.7	1320.7	**599.3**	582.3
Q6	1209.7	1192.6	**727.4**	710.4
K7	1081.6	1064.5	**855.5**	838.5
S8	994.5	977.5	942.5	925.5
Y9	831.5	814.4	1105.6	1088.6
K10	**703.4**	686.3	1233.7	1216.7
V11	**604.3**	587.3	1332.8	1315.7
S12	**517.3**	500.2	1419.8	1402.8
T13	**416.2**	**399.2**	1520.8	1503.8
S14	**329.2**	312.2	1607.9	1590.8
G15	**272.2**	255.1	1664.9	1647.9
P16	**175.1**	158.1	1761.9	1744.9

Figure 3. Theoretical spectrum for the peptide sequence "SIRVTQKSYKVSTSGPR", with parent mass of 1936.05 Da. "y" and "b" indicates y- and b-ions, "+1", "+2" indicates charge 1 and 2, and "*" indicates ammonia loss. Bold numbers are mass-to-charge ratios of peaks present in experimental spectrum.

This motivates us to call these contiguous sequences of strong ion-types (*b*-ions and *y*-ions of charge 1) "*strong tags*". More formally, they are defined as follows: Consider the extended spectrum graph, $G_1(S_1^\alpha)$, namely, only charge 1 ion-types. We define a *strong tag T of ion-type* (1, *t*, Ø) to be a maximal path (v_1, v_2, ..., v_r) in $G_1(S_1^\alpha)$ where each vertex $v_i \in T$ has the same ion-type (1, *t*, Ø) and (v_i, v_{i+1}) is an edge in the graph if the mass difference of v_i and v_{i+1} is the mass of one amino acid. (We consider only *b*-ions and *y*-ions, namely, *t* = *b*-ions or *y*-ions and strong tags must have at least 2 edges.)

Figure 4 shows the two strong tags obtained for the spectrum given in Figure 3.

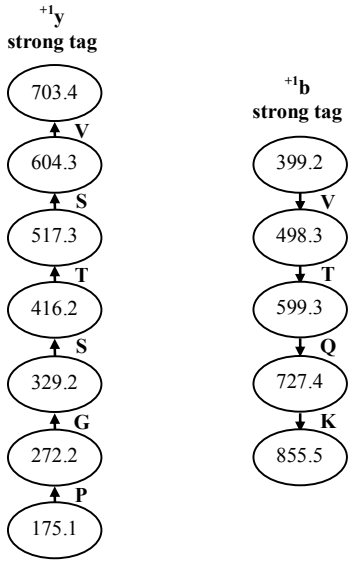


Figure 4. Example of strong tags in the spectrum graph for spectrum in Figure 3. There are 2 strong tags. Vertices (small ovals) represent mass-to-charge ratios, and edges (arrows) represent amino acids whose mass are the same (within tolerance) as the mass difference of the vertices.

3.3 Evaluating Mass Spectra

In this project, we have used the extended spectrum graph model that better describes multi-charge spectra. We have also proposed quality measures for multi-charge spectra based on the new model. Our evaluation of multi-charged spectra from GPM with the

new model shows that the theoretically attainable accuracy increases as we consider higher charge ions, meaning that multi-charge ions are significant. In addition, we show that any algorithm that considers only charge 1 or 2 ions will suffer from low prediction accuracy. Our experiments show that the accuracy (accuracy measure defined later) of these algorithms on multi-charge spectra is very low (less that 35%), and this accuracy decrease as the charge of the spectra increases (for charge 4 spectra, the accuracy of Lutefisk is less than 7%).

3.3.1 Quality measures for evaluating mass spectra

We have extensively analyzed many multi-charge spectra using extended spectrum graph model. We define two *quality measures* of a multi-charge spectrum

$$Specificity(\alpha, \beta) = |TS_{\beta}^{\alpha}(\rho) \cap S| \,/\, |S| \quad (2)$$

$$Completeness(\alpha, \beta) = |TS_{0}(\rho) \cap PRM(S_{\alpha}^{\alpha})| \quad (3)$$

Specificity measures the proportion of true peaks in the experimental spectrum *S*, and it can also be consider the signal-to-noise ratio of *S*. The completeness measure computes the proportion of the fragment masses that are explained by support peaks. By using completeness measurement, multiple support peaks for the same fragments are not double-counted.

3.3.2 Experimental data and analysis

The data being used for analysis is the Amethyst data set from GPM (Global Proteome Machine) [26] (obtainable from ftp://ftp.thegpm.org/quartz). The GPM system is an open-source system for analyzing, storing, and validating proteomics information derived from tandem mass spectrometry. One feature of the Amethyst dataset is that there are lots

of multi-charge spectra (up to charge 5). These data are MS/MS spectra obtained from QSTAR mass spectrometers. Both MALDI and ESI sources were included.

Using the $G_d(S_\beta^\alpha)$ extended spectrum graph model (with d=2), we measured the average *Specificity*(α,β) and *Completeness*(α,β) on the entire Amethyst datasets from GPM using our extended spectra S_β^α for $1 \le \alpha \le 5$, and $1 \le \beta \le \alpha$. A mass tolerance of 0.5 Da is used for matching. Since GPM datasets are of reasonably good quality, all the data in the Amethyst dataset (12558 datasets in total, with 4000, 4561, 2483, 1175, 339 for charge 1, 2, 3, 4, 5, respectively) has been used for this purpose.

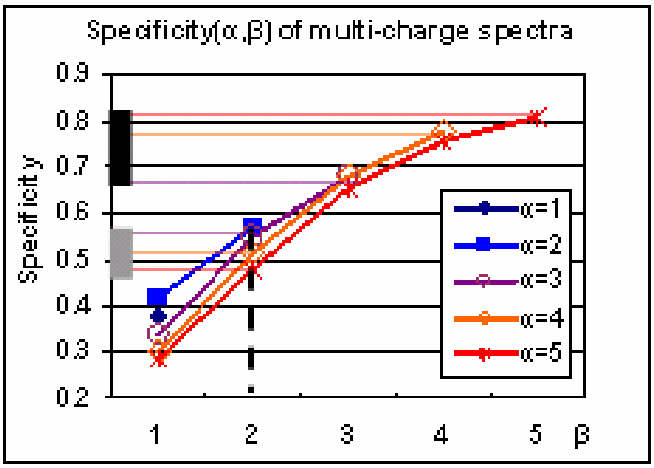


Figure 5. *Specificity(α,β)* of multi-charge spectra. Specificity increases as β increases. Most algorithms consider up to S_2^α (dashed black line). But considering S_α^α for spectra with $\alpha \ge 3$ improves the specificity (black line vs grey line).

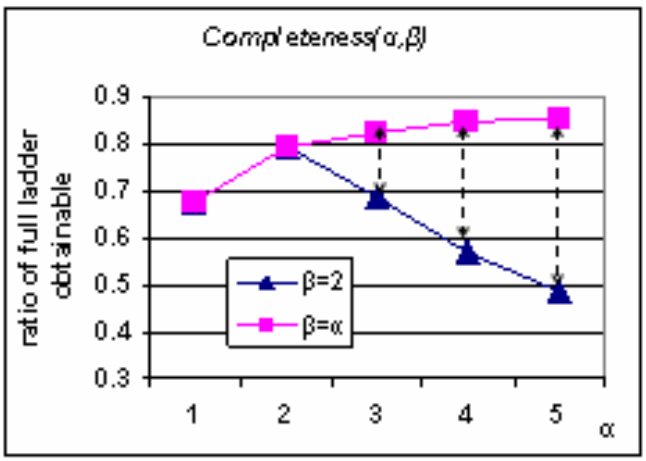


Figure 6. Completeness(α,β) of multi-charge spectra. We see that considering only S_2^α gives < 70% of the full ladder, which drops drastically as α gets bigger. On the other hand, considering S_α^α gives > 80% of full ladder.

The *Specificity*(*α*,*β*) results are shown in Figure 5. The results show that the GPM spectra contain an abundance of higher charged peaks in high-charge spectra. For a fixed *α*, as *β* increases, the specificity increases – meaning that more true peaks are discovered. Furthermore, the increase is significant. For *α*=5, the specificity increases from 0.49 with *β*=2, to 0.81 when *β*=5. Algorithms that uses *β* =2 considering only charge 1 and 2 (like LuteFisk and PepNovo) are limited to specificity values of between 0.48 to 0.56, as indicated by the dashed vertical line at *β*=2.

The *Completeness*(*α*,*β*) results are shown in Figure 6. In this graph, we compare the *Completeness*(*α*,*β*) results for (a) using the full extended spectrum S_α^α versus (b) using only S_2^α. Again, the results clearly show that significant improvement can be obtained by considering high-charge peaks. The disparity increases with *α*, as seen from the widening gap indicated by the vertical arrows.

3.4 GBST Algorithm for Multi-Charge Spectra

We also proposed a simple *De Novo* sequencing algorithm called *GBST* (Greedy Best Strong Tag) that considers high-charge ions based on extended spectrum graph model. Experimental results on GPM spectra show that *GBST* outperforms some of the other *De Novo* algorithms on spectra with charge ≥ 3.

3.4.1 Evaluate "best" strong tags

To help the search for good strong tags, we define a *weight function* that is used to score vertices and strong tags. The weight of vertex $v_i \in G_1(S_\beta^\alpha)$ is defined as

$$w(v_i) = \frac{f_{support}(v_i) + f_{loss}(v_i) + f_{intensity}(v_i)}{f_{tolerance}(v_i)} \quad (4)$$

- $f_{support\text{-}ion}(v_i)$ is a function of the number of v_j, with v_j having a different ion-type as v_i, but represent same PRM
- $f_{loss}(v_i)$ is a function of the number of v_j, with $(PRM(v_i) - PRM(v_j))$=17 or 18,
- $f_{intensity}(v_i)$ is a function of $(\log_{10}(\text{int}(v_i)))$,
- $f_{tolerance}(v_i) = (\sum \mid \text{PRM}(v_j) - \text{PRM}(v_i) - \text{m}(a_k) \mid)/N$, where N is the total number of incoming and outgoing edges for v_i, and a_k is the amino acid for the edge (v_i, v_j) or (v_j, v_i).

For a strong tag $T=(v_1, v_2, \ldots, v_r)$, the weight $W(T)$ of the strong tag T is just the sum of weight of the vertices in T, namely, $W(T) = \sum_{v_i \in T} w(v_i)$. The spectrum graph $G_1(S_\beta^\alpha)$ is a DAG that may consist of several disjoint components. Obviously, we are interested in finding a set of "best" strong tags, namely, tags that optimizes the weight $W(T)$ in a

component. We let *BST* denote the set of "best" strong tags from each of the components *C* in the spectrum graph.

3.4.2 The GBST algorithm

We developed a simple *De Novo* peptide sequencing algorithm that uses the best strong tags in the spectrum graph based on best strong tag, which we call the *Greedy Best Strong Tag (GBST)* algorithm.. The GBST algorithm first computes a set *BST*. To find best strong tags, the algorithm uses ion-types that appear most frequently, namely, charge 1, *b-ions* and *y-ions* with no neutral loss. The restricted set is given by $\Delta^R = (\Delta_z^R \times \Delta_t^R \times \Delta_h^R)$, where $\Delta_z^R = \{1\}$, $\Delta_t^R = \{b, y\}$, and $\Delta_h^R = \{\phi\}$. They also define $G_1(S_1^\alpha, \Delta^R)$, the spectrum graph $G_1(S_1^\alpha)$ where the ion types considered are restricted to those in Δ^R . Then, a *best strong tag T* of ion-type (*z*, *t*, *h*) $\in \Delta^R$ is a maximal path $\langle v_0, v_1, v_2, \ldots, v_r \rangle$ in the graph $G_1(S_1^\alpha, \Delta^R)$, where every vertex $v_i \in T$ is of a (*z*, *t*, *h*)-*ion*. In each component of this graph, GBST compute a "best" strong tag with respect to scoring function [2] described above. Then, the set *BST* is the set comprising the best strong tag for each component in the spectrum graph $G_1(S_1^\alpha, \Delta^R)$.

After the set of best strong tags, *BST*, is computed, the GBST algorithm then proceeds to find the best sequence that result from paths obtained by "extending" the tags from *BST* using all possible ion-types. It searches for paths in the graph $G_2(BST)$ defined as follows: the vertices are the best strong tags in *BST*, and we have a directed edge from the tail vertex *u* of a best strong tag T_1 to the head vertex *v* of another best strong tag T_2 if there is a directed edge (*u*,*v*) (or mass difference) in the graph $G_2(S_\alpha^\alpha)$. We note two major

difference between $G_2(BST)$ and the extended spectrum graph $G_2(S_\alpha^\alpha)$ – firstly, the number of vertices in $G_2(BST)$ is smaller; and secondly, the number of edges is also much smaller since only best strong tags are linked in a head-to-tail manner.

3.4.3 Upper bound on sensitivity

Given any spectrum graph G defined on an experimental spectrum S from a *known* peptide ρ, the notion of *theoretical upper bound on sensitivity* is defined as follows: Given G, we can compute the path in G that *maximizes* the number, p^*, of amino acids from the (*known*) peptide ρ. Then, $U(G) = p^*/|\rho|$ is an upper bound on the *sensitivity* for *any sequencing algorithm* based on the spectrum graph approach using the graph G. Then $U(G_d(S_\beta^\alpha))$ is the *theoretical upper bound on sensitivity* for the extended spectrum graph $G_d(S_\beta^\alpha)$, namely using the extended spectrum S_β^α with all ion types in Δ and a connectivity of d. PepNovo and Lutefisk which considers charge of up to 2 (and connectivity of up to 2) are bounded by $U(G_2(S_2^\alpha))$ and there is a sizeable gap between $U(G_2(S_2^5))$ and $U(G_2(S_5^5))$.

3.4.4 Experiments

Datasets and Experiment Settings

To evaluate the performance of GBST vis-à-vis the upper bounds, we used spectra that are annotated with their corresponding peptides – the GPM-Amethyst dataset [26] (Q-star data with good resolution) and the ISB dataset [27] (Ion-Trap data with low resolution). For each dataset, we selected subsets of spectra with annotated peptides validated by X-correlation score (Xcorr ≥ 2.5).

Table 1 lists the number of spectra and the number of peaks per spectrum for GPM and ISB spectra with different charges. In addition, peptides for GPM spectra have average lengths of 14.5 amino acids, and peptides for ISB spectra have average length of 15.0.

Table 1 : The number of spectra, and the number of peaks per spectrum. The results are based on the GPM and ISB datasets of different charges.

Charge	**No. Spectrum**		**No. peaks per spectrum**	
	GPM	**ISB**	**GPM**	**ISB**
1	756	16	48.2	149.6
2	874	489	46.9	144.5
3	454	490	42.6	145.1
4	207	-	46.8	-
5	37	-	46.1	-
Total	2328	995	46.5	144.9

Each GPM spectrum has between 20-50 peaks (usually high quality peaks) and an average of about 40 peaks. We use all of the peaks in our experiments. In contrast, each ISB spectrum has between 50~300 peaks and an average of 150 peaks.

We have applied the GBST algorithm on these spectrum data. For these spectra, we have also compared the results of GBST with those of the Lutefisk [23] and PepNovo [19]. For the comparison of prediction results, we defined two accuracy measures:

$$Sensitivity = \#\text{correct} / |\rho| \quad (5)$$

$$Specificity = \#\text{correct} / |P| \quad (6)$$

where #correct is the "*number of correctly sequenced amino acids*". The number of correctly sequenced amino acids is computed as the *Longest Common Subsequence* (LCS) of the correct peptide sequence ρ and the sequencing result P. *Sensitivity* indicates the quality of the result with respect to the correct peptide sequence and a high sensitivity means that the algorithm recovers a large portion of the correct peptide. For fair

comparison with algorithms like PepNovo that only outputs the highest scoring tags (subsequences), we also use the specificity measurement.

Comparison with Other Algorithms: In the experiments, we have only used GPM datasets, and run PepNovo on spectra with charge 1 and +2 (since it only handles spectra with charge 1 and +2), and compared the results with GBST algorithm.

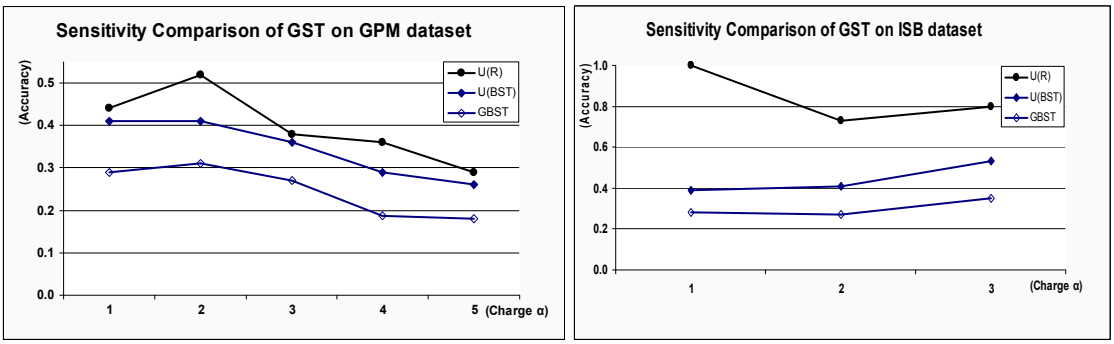
Table 2: Results of GBST, compared with Lutefisk and PepNovo on GPM spectra. Results show that GBST is generally comparable and sometimes better, especially for multi-charge spectra. The accuracy values are represented in a (specificity/sensitivity) format. (*based on spectra with +1 and +2).

Charge	Number of spectrum	Lutefisk	PepNovo	GBST
1	756	0.261 / 0.258	0.322 / 0.186	***0.296 / 0.315***
2	874	0.243 / 0.241	0.316 / 0.215	***0.297 / 0.326***
3	454	0.111 / 0.113	-	***0.262 / 0.285***
4	207	0.065 / 0.063	-	***0.190 / 0.222***
5	37	0 / 0	-	***0.165 / 0.223***
All	**2328**	**0.203 / 0.202**	**0.319 / 0.202***	***0.278 / 0.304***

Experiment results show that the GBST algorithm generally performs comparably to or better than Lutefisk [23] and PepNovo [19]. This is obvious for multi-charge spectra. The relatively high specificity of the results of GBST is comparable to the results of Lutefisk and PepNovo. The higher sensitivity shows that the GBST algorithm can identify more correct amino acids than Lutefisk and PepNovo.

Upper Bounds on Sensitivity for GBST: Since the GBST algorithm uses a restricted set of ion-types Δ^R in its search for best strong tags, we let $U(R)= U(G_1(S_1^{\alpha}, \Delta^R))$ be the upper bound on sensitivity *with ion-type restriction*. For the second phase, we define $U(BST) = U(G_2(BST))$, the upper bound on sensitivity *with best strong tags restriction*.

Comparison with Upper Bounds: We have computed the upper bounds on sensitivity for both the GPM and the ISB datasets and the results are shown in Figure 7, together with the actual sensitivity obtained by the GBST algorithm. The results in Figure 7 show that for GPM datasets, $U(BST)$ is near to $U(R)$, but the GBST results have sensitivities about 10% less than $U(BST)$. This indicates that GBST has not been able to fully utilize the power of *BST*. For the ISB datasets, even $U(BST)$ is far from $U(R)$. Therefore, it is natural the GBST algorithm can not perform well on ISB datasets.



(a) (b)

Figure 7: The comparison of sensitivity results of *GBST* with theoretical upper bounds. $U(R)$ and $U(BST)$ on (a) GPM dataset, and (b) ISB datasets.

However, since there is still a large gap between the accuracies of GBST sequencing results and U(BST) and U(R), we think that the algorithms based on using tags still have room for further improvement.

3.5 GST-SPC Algorithm

In this project, I present an improved *De Novo* algorithm called *GST-SPC* that extends on GBST algorithm. In the first phase, the GST-SPC algorithm computes a *larger* set of strong tags – the set of all "*maximal* multi-charge strong tags". We show that this improves the theoretical upper bound on the sensitivity. In the second phase, the GST-

SPC algorithm computes a peptide that is *optimal* with respect to *shared peaks count* (*SPC*) from among all peptides that are derived from strong tags. The SPC is computed as the number of shared peaks between experimental spectrum and theoretical spectrum of candidate peptides (within tolerance). Our evaluation shows that the GST-SPC algorithm improves on GBST, especially on multi-charge spectra.

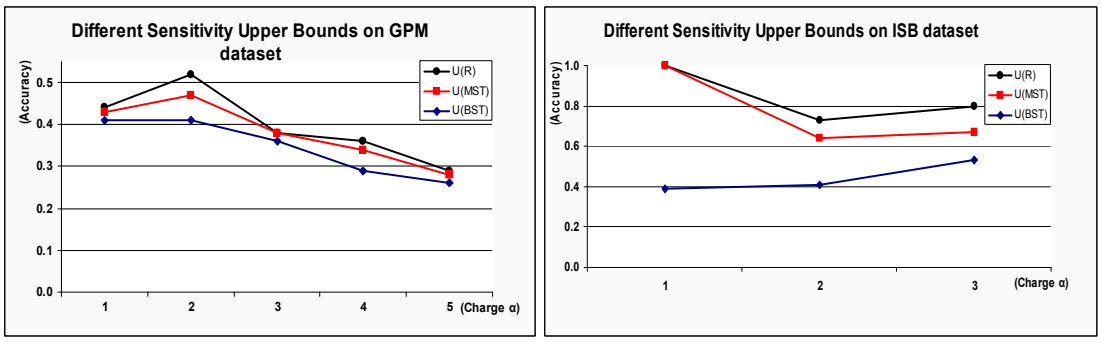
3.5.1 An improved algorithm – GST-SPC

(a) Using a Larger Set of Strong Tags: A straight-forward improvement of GBST [2, 3] is to expand the set of strong tags under consideration. We do this as follows: (i) when searching for strong tags, we use multi-charge ions (using S_α^α instead of just S_1^α), and (ii) instead of choosing *only one* "best" strong tag from each component of the graph $G_1(S_1^\alpha, \Delta^R)$, we allow a set of all *multi-charge strong tag* in each component of the graph $G_1(S_\alpha^\alpha, \Delta^R)$ to be chosen. Namely, a *multi-charge strong tags* of ion-type $(z^*, t, h) \in \Delta^R$ is a maximal path $\langle v_0, v_1, v_2, \ldots, v_r \rangle$ in $G_1(S_\alpha^\alpha, \Delta^R)$, where every vertex v_i is of a (z^*, t, h)-ion, in which t and h should be the same for all vertices, but z^* can be different numbers from $\{1, \ldots \alpha\}$. We let *MST* denote this set. The algorithm for computing the *MST* is almost identical to that for *BST* (a depth-first search), with slight modification to store the *MST* instead of *BST*. Running the GBST algorithm with the MST (GMST algorithm) improves the results slightly.

Theoretically, the size of the *MST* can be exponential. However, in practice, our experiments show that the MST does not exhibit exponential growth compared to BST. For GPM datasets (average of about 46 peaks) the increase in the average number of strong tags is from 10 to about 50. For ISB datasets (average of 145 peaks) the increase is

from 15 to about 90. As for tag length, the average length of strong tags in MST is 4.65 amino acids for GPM datasets, and 2.26 amino acids for ISB datasets.

We define $U(MST) = U(G_2(MST))$ the *theoretical upper bound on sensitivity with respect to the set MST*. The increase from $U(BST)$ to $U(MST)$ is shown in Figure 8. From Figure 8, it is easy to see that the introduction of MST has pushed up the theoretical upper bounds for both datasets. For GPM dataset, the best sequencing results obtainable from MST is about 5% higher in accuracy than BST. We also note that $U(MST)$ is very close to the $U(R)$, the theoretical upper bounds with Δ^R. For ISB datasets, the increase is more pronounced – partly because the ISB datasets have more peaks. The best sequencing results obtainable from *MST* is about 10%~60% higher in accuracy than *BST*, and within 20% of the theoretical upper bounds. This shows a great potential for sequencing algorithms based on *MST*.



(a) (b)

Figure 8. Comparing the theoretical upper bounds on sensitivity for MST and BST. Results are based on (a) GPM dataset, and (b) ISB datasets.

(b) Optimal Shared Peaks Count: While the GMST algorithm using MST is slightly better than GBST algorithm using BST, there is still a gap in performance compared to upper bounds. This motivates us to formulate the problem of *maximizing the shared peaks counts* (SPC) with respect to the set of multi-charge strong tags. The shared peak

count (SPC) is a commonly used and fairly objective criterion to compare experimental spectrum with theoretical spectrum of the peptides. We also show that we can solve this problem *optimally in polynomial time*.

Suppose that we are given the set, say MST, of strong tags. Define a *multi-charge strong tag path* Q to be a path from v_0 to v_M given by $Q = (q_0\ T_1\ q_1\ T_2\ q_2\ T_3\ q_3\ \ldots\ q_{k-1}\ T_k\ q_k)$ where each T_j is a strong tag in MST and each q_j is an edge of at most *two* amino acids, or mass difference that "links" the preceding tag to the succeeding tag in the usual head-to-tail fashion. A strong tag path Q gives rise to a peptide sequence $P(Q)$ obtained by interpreting the "gaps" in the path Q. A example of P(Q) is "[50]CGV[100]PK". Given the peptide sequence $P(Q)$, we can compute the shared peaks count of $P(Q)$. Then our problem can be stated as the following: Among all the possible strong tag paths, we want to find an *optimal multi-charge strong tag path* Q^* that *maximize the shared peak count* between the theoretical spectrum of peptide sequence $P(Q^*)$ and experimental spectrum.

Our solution to this problem is to form the graph $G_2(MST)$ defined in the same ways as the graph $G_2(BST)$. We first pre-compute the shared peaks count for each tag in *MST*. For each edge (u, v) connecting two tags T_u and T_v, we compute the path Q of length with at most two amino acids that locally maximizes that shared peak count of Q against experimental spectrum. Then we compute the path from v_0 to v_M with maximum shared peaks count in the graph $G_2(MST)$, which is a DAG. Additional processing has to be done if neither of the end vertices is connected to the first (or last) vertex in the path, or the sparse areas are not connectable - we connect this via mass difference. It is easy to see that this algorithm optimizes the shared peaks count among all peptide sequences

obtained by extending the multi-charge strong tags in MST via connectivity 2. Next, we present an algorithm that produces provably better result.

Improving the Spared Peaks Counts using *H*(*MST*): We can further improve the shared peaks count if we increase the maximum connectivity *d*. However, this will cause the running time to grow exponentially due to the number of paths to be searched. We propose a graph *H*(*MST*), a superset of G_2(*MST*) which is simple to define, and yet not too computationally expensive. In *H*(*MST*), we have an edge from the tail vertex *u* of T_u to the head vertex *v* of T_v if the mass difference (*PRM*(*v*)–*PRM*(*u*)) is in the range [57.02, 186.08] Da, where 57.02 Da and 186.08 Da are the minimum and maximum mass of any amino acid, respectively. In addition, we pre-compute the path from *u* to *v* that locally maximizes the shared peak count. We have fast procedure that solves this sub-problem efficiently. The length of the computed path from *u* to *v* varies depending on the mass difference. The rest of the algorithm is to interpret edges in *H*(*MST*).

Algorithm GST-SPC: Finally, our GST-SPC algorithm uses the multi-charge strong tag set *MST* and the graph *H*(*MST*) to compute a peptide with optimal *shared peaks count*.

3.5.2 Performance Evaluation of Algorithm GST-SPC

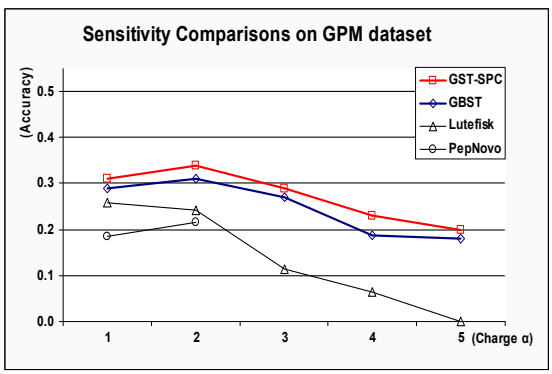
We have compared the performance of our algorithms with two other algorithms with freely available implementation, Lutefisk [23] and PepNovo [19]. For specific spectrum and algorithm, the sequencing results with best scores are compared. We have compared performance of GST-SPC with the GBST [2, 3], Lutefisk [23, 28], and PepNovo [19]. Except for formula (5) and formula (6), we have also used the following accuracy measures:

$$\text{Tag-Sensitivity} = \#\text{ tag-correct} / |\rho| \quad (7)$$

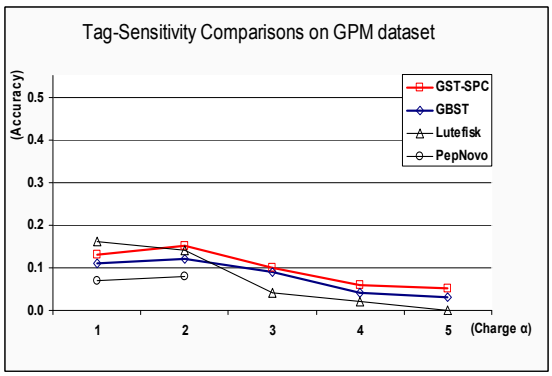
$$\text{Tag-Specificity} = \#\text{ tag-correct} / |P| \quad (8)$$

where #tag-correct is "*the sum of lengths of correctly sequenced tags* (*of length* > *1*)". Note that here "tag" only refers to subsequences, and not the "strong tag" that we have defined previously. The *tag-sensitivity* accuracy takes into consideration of the continuity of the correctly sequenced amino acids. For a fairer comparison with algorithms like PepNovo that only outputs the highest scoring tags (subsequences) we have also used tag-specificity, which measures how much of the results are correct.

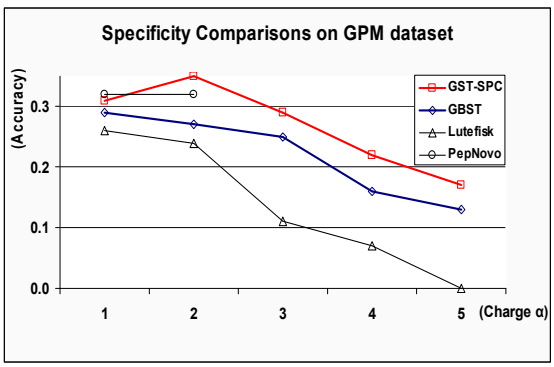
The comparison of the different algorithms based on these four accuracy measures is summarized in Figure 9 (for the GPM datasets) and Figure 10 (for the ISB datasets). Overall, the results obtained by our GST-SPC algorithm using the shared peaks count scoring functions are promising. On the GPM datasets, the GST-SPC outperforms the other algorithms. For example, it has higher sensitivity and tag-sensitivity than Lutefisk (by 10% for charge ≥ 2) and PepNovo (by about 10%). It has comparable specificity and tag-specificity to PepNovo for charge 1 and 2. It is constantly better than GBST and Lutefisk (for charge > 1) on all accuracy measures.

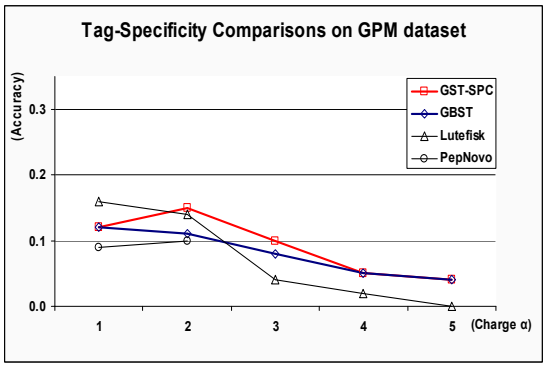


(a)



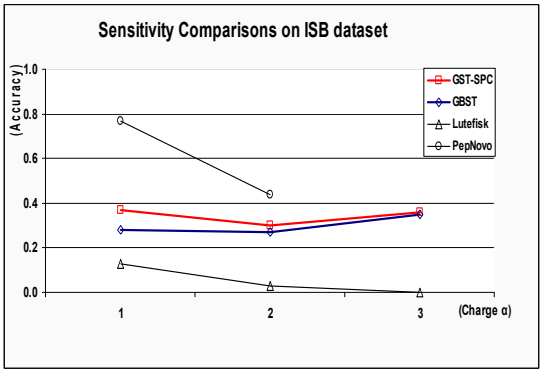
(b)

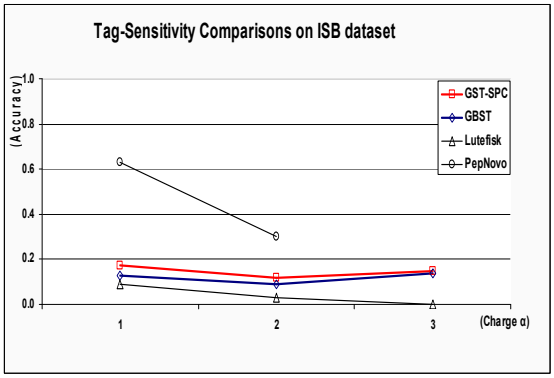




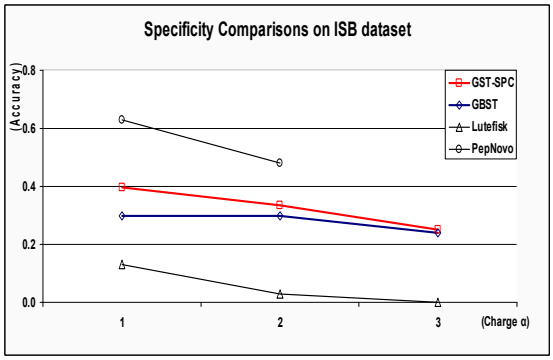
(c) (d)

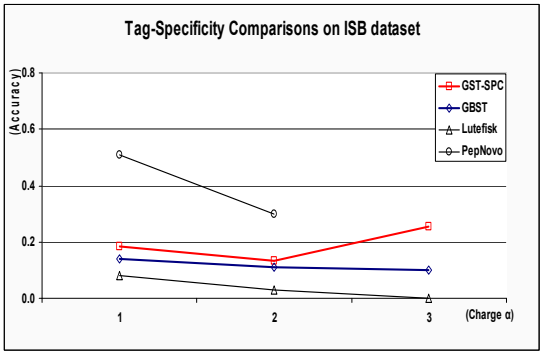
Figure 9. Comparison of different algorithms on GPM dataset – based on (a) sensitivity, (b) tag-sensitivity, (c) specificity and (d) tag-specificity. PepNovo only has results for charge 1 and 2.





(a) (b)





(c) (d)

Figure 10. Comparison of different algorithms on ISB dataset - based on (a) sensitivity, (b) tag-sensitivity, (c) specificity and (d) tag-specificity. PepNovo only has results for charge 1 and 2.

For the ISB dataset, the results show the ranking as follows: (PepNovo, GST-SPC, GBST, Lutefisk) for all the accuracy measures. The ISB datasets contains much noise and PepNovo has a sophisticated scoring function that may account for its best

performance, especially on datasets with charge 1. For spectra with charge 2, the difference in performance is not as big. However, since PepNovo do not (as yet) handle spectra with charge greater than 2, there was no way to compare results for charge 3. That comparison would be interesting given the apparent trend exhibited in the results.

We also compare the algorithm with respect to the number of *completely correct* identified peptide sequences. Our results (not shown here due to space limitations) show that the GST-SPC algorithm out-performs Lutefisk, but is slightly worse than PepNovo. We have also listed (in Table 3) a few sample "good" interpretations of the GST-SPC algorithm, on which Lutefisk does not provide good results. It is interesting to note that GST-SPC algorithm can identify more correct amino acids – illustrating the power of using multi-charge strong tags.

Table 3: The sequencing results of Lutefisk, PepNovo and GST-SPC algorithm on some spectra. The accurate subsequences are labeled in bold and italics. "-" means there is no result.

M/Z	Z	Real	Lutefisk	PepNovo	GST-SPC
1219.8	2	VAQLEQVYIR	[170.1]E***LEKVYLR***	GL***QLEQVYLR***	AVE***IEQVYIR***
1397.9	2	ELEEIVQPIISK	[242.1]***EEL***AVG[LP]***LSK***	***EELVKPLLSK***	***EIEEI***A[101.0]QH***ISK***
1644.9	2	PAAPAAPAPAEKTPVKK	[AP]***AAPA***[HS]AP[198.1]***PA***AA[CS]	***AAPA***DFEAMTNLPK	***APAAPAPA***[56.1]APAMTKVPK
1838.8	3	SSYSLSGWYENIYIR	[172.1]L[303.2][243.1][NP][MT]***LYLR***	-	***SS***IYI[27.3]IIEPCE***IYIR***
2000.2	4	PAAPAAPAPAEKTPVKKKAR	[323.1]R***PA***[AP]EKTN[LP]K[199.1]R	-	***APAAPA***MWNYNHKPYIR
1936.1	4	SIRVTQKSYKVSTSGPR	[199.1][PW][259.1]L[250.1]***KVSTSGPR***	-	VVIS***VTQK***[63.8]W***KVSTSGPR***
2101.1	4	KIETRDGKLVSESSDVLPK	[243.1]***LV***R[TY]YT***SESS***AE[PV]R	-	IKQHTHECY***SESSDVIPK***
2359.0	5	CDKDLDTLSGYAMCLPNLTR	-	-	AF***CDYA***[417.2]RNQKIRCP***TR***

3.6 PSP Database Search Algorithm

Extending the idea of using tags [11] on GBST algorithm, we have developed a new database algorithm, the PSP algorithm that concentrated on the multi-charge spectrum data. We have tried to utilize all of the tags information, and tried to get the best results based on this information. In PSP algorithm, we first find out best strong tags (BST) from the spectrum, and connect them by their mass differences; these are called *Peptide Sequence Patterns (PSPs)*, and the peptide sequences in the database that best match the

PSPs are selected for further processing. Then a linear time database search process is used to search candidate peptides sequences by PSPs. These candidate peptides are then scored and ranked by shared peaks count.

3.6.1 Peptide sequence patterns algorithm

The PSP algorithm first compute a set, *BST*, of "best" strong tags. The PSP algorithm then proceeds to find the PSPs that result from paths obtained by "connecting" the tags from *BST*. This is done by searching for paths in the graph $G_d(BST)$ in which the vertices are the strong tags in *BST*, and we have an edge from the tail vertex u of T_1 to the head vertex v of T_2 if $PRM(v)$ is larger than $PRM(u)$. Note that there is a different from this approach to that used in GBST algorithm. Since in PSP algorithm, the tags from *BST* are not extended before linking of the tags.

The peptide sequence patterns (PSPs) that represent the paths compose of the tags and mass fragments. Formally, $PSP_i = m_1 t_1 m_2 t_2 ... m_n t_n m_{n+1}$, in which m_i and t_i refer to mass difference and tag, respectively. Each tag in the sequence composes of those consecutive amino acids. Each mass is the sequence represents the mass difference between tags.

After PSPs are retrieved, the PSPs are scored and ranked according to shared peaks count of the theoretical spectrum of the PSP and the experimental spectrum. Some top PSPs are then selected for database search.

The database search algorithm is essentially an approximation pattern matching in the database, with PSPs (composed of tags and mass differences) as patterns. The detailed database search algorithm will be described later.

After database search based on PSPs, several candidate peptides are obtained. For each of candidate peptide sequences, the shared peaks count is computed by comparing the theoretical spectra of the candidate peptides against the experimental spectrum.

The scheme and the description of the PSP algorithm are illustrated in Figure 11 and Figure 12, respectively.

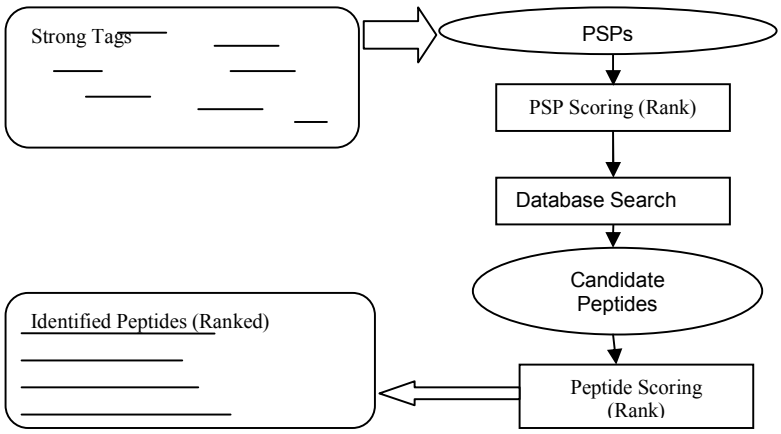


Figure 11: The scheme of the database search algorithm.

1. **Search for strong tags**
   - Transform spectrum to extended spectrum graph
   - Select all of the best strong tags (BST) in extended spectrum graph
2. **Generation of PSPs**
   - Connect BSTs by mass differences
   - Generate a graph G, every vertex is a BST, every edge is one mass difference
   - List all paths from start to end vertexes
   - For each of the path P_i, generate the peptide sequence pattern PSP_i
   - Score and rank PSPs by shared peaks count
3. **Database search by PSP**

Figure 12: The description of the PSP algorithm.

3.6.2 Approximate database search using PSP

The candidate peptides are obtained by searching in the database with PSP. By searching the database, we can identify those peptides that match with a certain number of tags (with 1 or 2 amino acids errors) in PSP.

The approximate pattern matching problem in the context of peptide sequencing is a pattern matching problem. It involves both approximate tags matching and approximate masses matching.

String matching has been investigated by many researchers, and there are many theories and algorithms on it. It is known that inexact string matching with errors can be done in linear time, and exact string matching with wildcard can be done in linear time [1, 29]. Moreover, the semi-numerical inexact string matching algorithms [1, 29] can be very efficient if the patterns are relatively short. In the PSP algorithm, we have used the semi-numerical inexact string matching algorithms, so the database search process is linear in computational time.

The problem definition and the procedure of approximate database search are listed in Figure 13. An illustration of approximate match of PSP to the peptide sequences in the database is in Figure 14.

Problem: Approximate database search using PSP

1. Input:
    1) *peptide sequence pattern* (PSP)
    $PSP_i = m_1t_1m_2t_2...m_nt_nm_{n+1}$ (m_i and t_i refer to mass and tag, respectively)
    2) database sequence, *Seq*
2. Output:
    3) Subsequence Seq_i (or subsequences) in *Seq* that fulfill the requirements
3. Constraints:
    1) Approximate match with tags t_i in Seq_i in order, with strict tolerance (every tag with $\leq$2 amino acids error); if at most $m<n$ tags are present for

---

every database sequences, then these m tags should be approximately matched
2) Approximate match with masses m_i in Seq_i in order, with loose tolerance (every mass with ≤50 Da mass error)
3) Efficient process

Procedure: Approximate database search using PSP

1. Select the top PSPs (depending on the total number of PSPs), search database for candidate peptides that approximately match with the tags and masses of these PSPs within certain tolerance.
2. Score and rank the candidate peptides by the shared peaks count between their theoretical spectrum and experimental spectrum.
3. Output these peptide sequences.

Figure 13: Description of the approximate pattern matching problem; and the procedure for the database search algorithm.

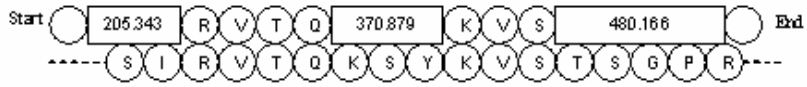


Figure 14: An example of the match of the peptide sequence pattern (first row) and the peptide sequence in the database (second row).

As illustrated in Figure 14, the PSP is "[205.343]RVTQ[370.879]KVS[480.166]" (numbers in brackets represent mass differences); and the matched peptide sequence is "SIRVTQKSYKVSTSGPR". In this example, the two tags "RVTQ" and "KVS" have matched the identical fragments in the peptide sequence (1 or 2 amino acids mismatches are tolerable). The three mass differences also match with the fragments having similar masses.

As for the running time, for one PSP with length of m and database size (total length of sequences) of n, the algorithm can operate in $O(m+n)$ time. This is much better than the naïve sequence matching method, which requires $O(m*n)$ time. Since there are thousands of peptide sequences in database, the efficiency improvement is very significant. If we

load the peptide database into memory once, and search several PSPs against it, the average processing time for a PSP can be even shorter.

3.6.3 Experiments

In these experiments, dataset being used is GPM (Global Proteome Machine) dataset [26] with different charges. The methods to be compared are PeptideSearch [11], SPIDER [22], the 2 typical database search methods based on tags; Mascot [30], one of the most popular database search methods; and the recent InsPecT [14] software.

Both of PeptideSearch and SPIDER need a tagged sequence (sequence composed of tags and masses) as input; we have used the PSP generated by our algorithm as such tagged sequence. For Mascot and InsPecT, the input is original spectrum data. The PeptideSearch algorithm uses the non-redundant database in FASTA format, which obtain the peptide sequences from various protein databases. SPIDER, Mascot, InsPecT and our algorithm used the Swiss-Prot protein database. The Swiss-Prot protein database that we have used is Swiss-Prot Release 45.5 of 04-Jan-2005, which contains 167089 protein sequence entries. The default parameters have been used for all of these algorithms, and the sequencing result with top rank is used for analysis.

We have compared PSP algorithm with Mascot [30] and InsPecT [14] in details, and explained the comparison results against PeptideSearch [11] and SPIDER [22] briefly.

The comparison with Mascot is meaningful. The Mascot algorithm is currently regarded as one of the most accurate database search algorithms. More important is that Mascot is not based on tags, and the input is the spectrum data, same as our algorithm's input.

Therefore such comparison is fair. The results of the comparisons are shown in Table 4. The "accurate subsequences" refer to the subsequences of the correct sequences, and at the appropriate position of the corresponding sequences.

Table 4: Comparisons of Mascot and PSP on selected spectra. The accurate subsequences are labeled in italics. A "-" means that there is no result.

M/Z	charge	correct	Mascot	PSP
1219.8	2	VAQLEQVYIR	*VAQLEQVYIR*	*VAQLEQVYLR*
1397.9	2	ELEEIVQPIISK	*ELEEIVQPIISK*	*ELEEIVQPIISK*
1644.9	2	PAAPAAPAPAEKTPVKK	LHGGNAIGFMTLEGTK	*AAPA*ETSDLEFA*VKK*
881.5	2	SPRLRPR	LVIVAL*PR*	*SP*IVRG*PR*
1448.7	2	LPGAYFFSFTLGK	MLRAMVASGSE*LGK*	LVRGQNTVHI*LGK*
1888.1	3	VTHAVVTVPAYFNDAQR	*VTHAVVTVPAYFNDAQR*	I*VVT*QPRRISAVSV*AER*
1934.1	3	DNHLLGTFDLTGIPPAPR	*DNHLLGTFDLTGIPPAPR*	KNVA*LIG*LTVE*TG*SALVPK
1934.3	3	DNNLLGKFELTGIPPAPR	*DN*H*LLG*TFD*LTGIPPAPR*	*DNNLLGKFELTGIPPAPR*
1838.8	3	SSYSLSGWYENIYIR	*SS*L*SIS*SMFCNYDETR	*SSYSLSGWYENIYIR*
1761.0	3	PAAPAPAEKTPVKKKAR	LFFAFEKQESVPYR	-
1932.8	4	HKVYACEVTHQGLSSPVTK	VFFDNNFQCILWFLK	TLKVDGNDETFALSNISK
2000.2	4	PAAPAAPAPAEKTPVKKKAR	GQYEPV*AE*IGVGAYGTVYK	*PAAP*K*AAPA*TPAAPAPVYLR
1936.1	4	SIRVTQKSYKVSTSGPR	EGEYTGRTPSGADVTLQR	*SIRVTQKSYKVSTSGPR*
2101.1	4	KIETRDGKLVSESSDVLPK	MVQPDSSSLAEVLDR*VL*DK	*KIETRDGKLVSESSDVLPK*
2140.2	4	KASGPPVSELITKAVAASKER	GER*PP*DVETTVILPESVFR	*KASGPPVSELITKAVAASKER*
1933.3	4	VTIAQGGVLPNIQAVLLPK	DPEDGRPAPGVEHSNGLGK	*VTIAQGGVLPNIQAVLLPK*
3292.8	5	LLILEAGHRMSAGQALDHPWVITMAAGSSMK	EP*LE*LEDIPIEIDNDDDEDDED*GS*GVEYD	[387.26]WCGG[12.55]GD[1438.93]PIDIY*MK*
3291.8	5	LEILLHLTSLSQTFNHFFPEEKFETLR	QPIYPYGSPMGAHVYYPPPVAQPPVRGPVR	SPKVPRTL*LT*LDEQVLSFQRKVGILYCR
3151.2	5	MGSMFRSEEVALVQLFLPTAAAYTCVSR	*GS*GLPDLVLD*VA*GEFYKFGLEGIGAVLLG*SR*	D*EEV*DELYREAPIDKKGNFNYIEFTR
3752.0	5	LPPGEQCEGEEDTEYMTPSSRPLRPLDTSQSSR	CTPFRPSAMSPDFVAQVPLAPDLLPLAELFQRAR	RVEKNALKSQLRSMQ*EQ*LAEMQQKYVQLC*SR*
2359.0	5	CDKDLDTLSGYAMCLPNLTR	LGVMLVGWGGNNGSTLTAGVIANR	[1655.89]AGVPC*TR*

It is obvious from the table above that in these cases; our algorithm is more accurate than Mascot. Mascot can find exact match in only 4 cases, and ours can find exact match in 8 cases. In other cases, the results of our algorithm also have comparable number of subsequences matched to the true peptide sequences.

It is known that recent Mascot has already incorporated the tags function similar to PeptideSearch [11], but our algorithm can still beat Mascot on some spectrum data. This shows that our PSP algorithm, which adopts the new strategies to find tags from spectra, as well as our database search techniques, is quite effective.

The comparisons of PSP algorithm against PeptideSearch [11] and SPIDER [22] (details not shown) show that the PSP algorithm has comparable or higher accuracies than these tag-based algorithms.

To evaluate the performance of PSP and InsPecT algorithm, we use the accuracy measures (5)-(8).

Results (Table 5) show that the PSP algorithm has comparable accuracy results to InsPecT based on our accuracy functions. Though the PSP algorithm has lower accuracies than InsPecT for spectrum data with charge 1 and 2, it has comparable or higher accuracies compared with InsPecT for spectrum with charge > 2. This shows the power of PSP for multi-charge spectrum data.

Table 5: The accuracy results of PSP and InsPecT on GPM datasets. The accuracies in cells are represented in a (specificity/sensitivity/[tag-specificity /tag-sensitivity]) format.

Charge	**Number of spectrum**	**PSP**	**InsPecT**
1	756	0.301/0.285[0.110/0.108]	0.448/0.446[0.287/0.289]
2	874	0.412/0.400[0.213/0.212]	0.460/0.455[0.305/0.305]
3	454	0.338/0.339[0.143/0.144]	0.360/0.362[0.193/0.194]
4	207	0.302/0.322[0.099/0.109]	0.276/0.292[0.102/0.109]
5	37	0.286/0.340[0.088/0.120]	0.241/0.279[0.077/0.093]
Total	2328	0.350/0.343[0.153/0.152]	0.417/0.417[0.256/0.257]

We have calculated the ratios that the completely correct peptides are sequenced by the algorithms. Results show that InsPecT has better performance than PSP algorithm based on this criterion.

We have also compared some of our sequencing results with those obtained from InsPecT, and listed the sequencing results in details (Table 6). From these results, we can see that both PSP and InsPecT can correctly predict a large portion of the peptide sequences.

The experiments on ISB datasets [27] are also performed. The results show that our results are not as accurate as the results of InsPecT, but comparable to Mascot's.

Comparison with PeptideSearch and SPIDER show that the accuracies of PSP algorithm are superior to PeptideSearch algorithm, and comparable to SPIDER algorithm.

Table 6: Comparisons of InsPecT and PSP on selected spectra. The accurate subsequences are labeled in italics. A "-" means that there is no result.

M/Z	charge	correct	InsPecT	PSP
1219.8	2	VAQLEQVYIR	*VAQLEQVYIR*	*VAQLEQVYLR*
1397.9	2	ELEEIVQPIISK	*ELEEIVQPIISK*	*ELEEIVQPIISK*
1644.9	2	PAAPAAPAPAEKTPVKK	*PAAPAAPAPAEKTPVKK*	*AAPA*ETSDLEFA*VKK*
881.5	2	SPRLRPR	PSIVG*RPR*	*SP*IVRG*PR*
1448.7	2	LPGAYFFSFTLGK	*LP*QSLKLHIIV*GK*	LVRGQNTVHI*LGK*
1888.1	3	VTHAVVTVPAYFNDAQR	*VTHAVVTVPAYFNDAQR*	I*VVT*QPRRISAVSV*AER*
1934.1	3	DNHLLGTFDLTGIPPAPR	*DNHLLGTFDLTGIPPAPR*	KNVA*LIG*LTVE*TG*SALVPK
1934.3	3	DNNLLGKFELTGIPPAPR	*DN*H*LLG*TFD*LTGIPPAPR*	*DNNLLGKFELTGIPPAPR*
1838.8	3	SSYSLSGWYENIYIR	SDGGLVMKRDPTE*YIR*	*SSYSLSGWYENIYIR*
1761.0	3	PAAPAPAEKTPVKKKAR	–	-
1932.8	4	HKVYACEVTHQGLSSPVTK	–	TLKVDGNDETFALSNISK
2000.2	4	PAAPAAPAPAEKTPVKKKAR	*PAAPAAPAPAEKTPVKKKAR*	*PAAP*K*AAPA*TPAAPAPVYLR
1936.1	4	SIRVTQKSYKVSTSGPR	YGKPFKLIFH*VST*LQR	*SIRVTQKSYKVSTSGPR*
2101.1	4	KIETRDGKLVSESSDVLPK	*KIETRDGKLVSESSDVLPK*	*KIETRDGKLVSESSDVLPK*
2140.2	4	KASGPPVSELITKAVAASKER	*KASGPPVSELITKAVAASKER*	*KASGPPVSELITKAVAASKER*
1933.3	4	VTIAQGGVLPNIQAVLLPK	*VAQLEQVYIR*	*VTIAQGGVLPNIQAVLLPK*
3292.8	5	LLILEAGHRMSAGQALDHPWVITMAAGSSMK	*ELEEIVQPIISK*	[387.26]WCGG[12.55]GD[1438.93]PIDIY*MK*
3291.8	5	LEILLHLTSLSQTFNHFFPEEKFETLR	*PAAPAAPAPAEKTPVKK*	SPKVPRTL*LT*LDEQVLSFQRKVGILYCR
3151.2	5	MGSMFRSEEVALVQLFLPTAAAYTCVSR	PSIVG*RPR*	D*EEV*DELYREAPIDKKGNFNYIEFTR
3752.0	5	LPPGEQCEGEEDTEYMTPSSRPLRPLDTSQSSR	*LP*QSLKLHIIV*GK*	RVEKNALKSQLRSMQ*EQ*LAEMQQKYVQLC*SR*
2359.0	5	CDKDLDTLSGYAMCLPNLTR	*VTHAVVTVPAYFNDAQR*	[1655.89]AGVPC*TR*

The processing time of PSP algorithm is moderate. Running on a PC with 3GHz of CPU and 1GB of RAM, it uses about 10 seconds for the sequencing of one spectrum (the average of 50 PSPs checked). This running time is comparable with typical methods such as Sequest [16, 17], but slower than InsPacT [14]. The running time is also dependent on the quality of the spectrum data, especially the accuracy of the parent mass, so high quality data may result in fast process as well as high accuracy. For example, the running time is about 60 seconds for a spectrum data, for which we have generated more than 300 PSPs in step 2 of the PSP algorithm (refer to Figure 13).

3.7 New Computational Models for Preprocess and Anti-symmetric Problem

Though current extensive research in peptide identification helps to improve the accuracies, there are still many obstacles for both *De Novo* and database search approaches, which make further improvement of the accuracies of peptide identification difficult. Among these obstacles, preprocess to remove the noise from spectrum before peptide identification, as well as the anti-symmetric problem, are two very important issues; and they are our focus in this project.

Preprocess to remove noise

A peak in spectrum is noisy if it is not the result of peptide fragmentation, but due to contaminant in mass spectrometers, experiment environments, etc. Since most of the spectra contain a significant amount of noise, and noisy peaks may mislead interpretation; therefore, preprocessing to remove noisy peaks from the spectrum is necessary.

The anti-symmetric problem

A peak p_i is anti-symmetric if there can be different ion type interpretations for p_i, otherwise, p_i is symmetric. There is an anti-symmetric problem in spectrum S if S has one peak p_i which is anti-symmetric. For the spectrum graph G [18] used to represent spectrum, a path in G is called anti-symmetric if there are no two vertices (ion interpretations) on this path which represent the same peak; otherwise, there is anti-symmetric problem. The anti-symmetric problem is common in spectrum. Currently there are generally two approaches to the anti-symmetric problem. One approach is not to consider the anti-symmetric problem [28]; and another is to apply the *"strict" anti-symmetric rule* that require each peak to represent at most one fragment ion [29, 43, 79].

The "strict" anti-symmetric rule is used by many in peptide sequencing, but whether applying this rule is realistic is doubtful.

In this project, we addressed preprocess computational model to remove noise peaks from spectrum. This model also includes the method for introduction of "pseudo peaks" into the spectrum to improve peptide sequencing accuracies. We have also proposed the *restricted anti-symmetric model* for the anti-symmetric problem. We have then proposed a novel peptide sequencing algorithm which incorporate these two computational models.

3.7.1 Analysis of problems and current algorithms

Datasets

All of the experiments use the spectra selected with different charges from GPM dataset [31] and ISB datasets [27] as described previously in Table 1.

Problem Analysis

Since binning is generally the prerequisites for spectra data preprocessing, in this section, we first analyze the methods for binning of the peaks in the spectrum, and then discuss on using preprocessing to remove noisy peaks and introduce "pseudo peaks" into spectrum, followed by the analysis of anti-symmetric problem.

- **Binning of peaks in spectrum**

The binning idea is already embedded in [32, 33] for the purpose of mass spectrum alignment. In [32, 33], the peaks of the spectrum are packed into many bins of same size, and the spectrum is translated into sequences of 0s and 1s. More recently, a database search algorithm COMET [34] is proposed which uses the bins (usually of size 1 Da) for

their correlations and statistical analysis (Z-score) for accurate peptide identification by database search (spectrum comparison).

The important parameters considered in binning include the size of the bins, the interpretation of supporting peaks (bins), as well as the peaks (bins) intensity.

Lemma 1. Given the mass range m_{bin} for bin, and mass tolerance of m_t without binning. If we increase tolerance to $m_t^*=m_{bin}+m_t$ after binning, then the binning will not miss any possible amino acid interpretations.

Proof: For two peaks, p_i and p_j with mass of $m(p_i)$ and $m(p_j)$ respectively, and some amino acid with mass $m(AA_k)$, suppose $||m(p_i)-m(p_j)| - m(AA_k)| \leq m_t$, so there is an amino acid interpretations; also suppose after binning, their respective bin has the peak p_i^* and p_j^*. Then $||m(p_i^*)-m(p_j^*)| - |m(p_i)-m(p_j)|| \leq m_{bin}$. It follows that $||m(p_i^*)-m(p_j^*)| - m(AA_k)| \leq m_{bin}+m_t$. Given tolerance $m_t^*=m_{bin}+m_t$ after binning, it is obvious that $||m(p_i^*)-m(p_j^*)| - m(AA_k)| \leq m_t^*$. Therefore the same amino acid interpretation is not missed. Proved.

Therefore, it is clear that given the proper value of tolerance, the binning can preserve the accuracies. The binning method makes the removal of noise easier, and also makes sequencing faster and potentially more accurate, especially for noisy spectrum.

- **Preprocess to remove noisy peaks and introduce pseudo peaks**

Noisy peaks exist in every spectrum, but how to distinguish them from "true" peaks is not an easy problem. The first step is to analyze the spectrum data and find the patterns of noisy peaks. To this end, we have analyzed most abundant ion type: {b-ion, ∅, 1}, {b-ion, ∅, 2}, {b-ion, -H_2O, 1}, {b-ion, -NH_3, 1}, {y-ion, ∅, 1}, {y-ion, ∅, 2}, {y-ion, -H_2O, 1}, {y-ion, -NH_3, 1}, and assume those peaks not of these ion types noise. The analysis is

done on GPM dataset and ISB dataset. The theoretical spectrum that we have considered for peptide *P* can be obtained by generating all possible ion types from every PRM of *P*. Each of the possible ion types of a PRM is represented as a peak in theoretical spectrum. The experimental spectrum and theoretical spectrum for the corresponding peptide is compared, and peaks in experimental spectrum that can be matched with certain ion types are counted. The "content" of peaks for specific ion type is defined as the number of peaks of that ion type, over total number of peaks in experimental spectrum. The number of peaks and the contents of peaks of different ion types are analyzed, with results (average) in Table 7.

Table 7. The average contents of different types of peaks in GPM and ISB spectra. The symmetric peaks are just counted once for total content measures.

Ion type	**No. of peaks (Avg)**		**Content**	
	GPM	**ISB**	**ISB**	**GPM**
b-ion, ∅, 1	23.71	111.83	0.04	0.06
b-ion, ∅, 2	3.88	35.49	0.01	0.01
b-ion, $-H_2O$, 1	4.52	18.29	0.01	0.01
b-ion, $-NH_3$, 1	3.41	18.11	0.01	0.01
y-ion, ∅, 1	23.84	69.55	0.05	0.05
y-ion, $-H_2O$, 1	23.45	113.57	0.04	0.05
y-ion, $-H_2O$, 1	3.12	36.48	0.01	0.01
y-ion, $-NH_3$, 1	3.13	20.13	0.01	0.01
Noise	433.85	3017.9	0.83	0.80
Total	522.91	3441.35	1.00	1.00

From Table 7, we can see that noisy peaks form a significant portion of the peaks in the experimental spectrum. For GPM datasets, 80% of the peaks are noisy peaks, and the most abundant ion types - the b- and y- ion types, only compose 6% and 5% of the peaks. For ISB datasets, 83% of the peaks are noisy peaks, and the most abundant ion types - the b- and y- ion types, only compose 4% and 5% of the peaks. ISB spectra have more noisy peaks, and peptide sequencing for these spectra are more difficult.

Further analysis of the noisy peaks indicates that there are more noisy peaks in the middle part of the spectrum, than those at the two ends of the spectrum. Also, most of the noisy peaks have some features in common, such as low intensity and less other ion (b-, y-, loss of water or ammonia, for example) support.

For famous algorithms such as Lutefisk [28], there are no preprocessing done to remove noise. PEAKS [35] and PepNovo [19] are two famous algorithms that have implemented preprocessing. In PEAKS, the noise level of the spectrum is estimated, and the intensities of all the peaks in the spectrum are reduced by this noise level. Then all the peaks with zero or negative intensities are removed. PepNovo have preprocessed peaks to remove or downgrade peaks that have low intensity, and do not appear to be b- or y-ions. Recently, the AUDENS algorithm has been proposed [36]. The algorithm has a flexible preprocessing module which screens through the peaks in the spectrum, and distinguishes between signal and noise peaks.

Traditional preprocess for peptide sequencing by mass spectrometry only consider how to remove noisy peaks. However, since some fragment ions are not represented by any of the peaks, appropriate introduction of "pseudo peaks" into spectrum may connect the missing links, and increase the sequencing accuracies. The idea of pseudo peaks is first described in PEAKS [35]. It assumes that peaks are at every place in the spectrum, and those which are not present in the actual spectrum are peaks with 0 intensities. It is proven that appropriate introduction of "pseudo peaks" can partially solve the problem of missing edges in the spectrum graph approach [35]

In our preprocessing computational model, apart from noisy peaks removal, we will also introduce "pseudo peaks" into the spectrum. Notice that though the process is similar to previous work, the computation model is different.

- **The anti-symmetric problem**

We have mentioned that there are two approaches to the anti-symmetric problem. In the following part, we show that both of the approaches are based on unverified assumptions that cannot be verified in real spectrum.

To analyze the significance of the anti-symmetric problem in peptide sequencing, we generated the theoretical spectrum of known peptide sequences. We analyzed most abundant ion type: {b-ion, ∅, 1}, {b-ion, ∅, 2}, {b-ion, $-H_2O$, 1}, {b-ion, $-NH_3$, 1}, {y-ion, ∅, 1}, {y-ion, ∅, 2}, {y-ion, $-H_2O$, 1}, {y-ion, $-NH_3$, 1}, and assume there is no noise. The analysis is done on theoretical spectra for GPM dataset and ISB dataset. Two peaks are said to be overlap if their mass difference is within threshold (default of 0.25 Da). Note that each of such overlapping peaks is equivalent to a symmetric peak.

Results are shown in Table 8. The "average numbers" are the average number of symmetric peaks for theoretical spectrum of one peptide sequence, and the "average ratios" are computed as "average numbers", over average number of peaks in theoretical spectrum.

It is obvious that instances of overlaps (within threshold, 0.25 Da) are quite common. For the overlaps of b- and y-ions in GPM datasets, there is one overlap instance in about 5 peptide sequences, or in about 67 amino acids. The overall overlap instances are even

more common, one instance in about 0.36 sequences, or about 5 amino acids. The ISB datasets has a little less overlaps, but overall, there is still more than one instance in 0.35 sequences, or about one isntance in 4 amino acids.

Note that we have not considered peaks with high-charges ($z \geq 3$). But previous research [3] found significant amount of high-charge ($z \geq 3$) peaks in high-charge spectra. It is natural that the number of overlapping instances will increase when we consider high-charge peaks, and more ion types. Therefore, "strict" anti-symmetric rule is not realistic.

Table 8: The average numbers and ratios of overlapping instances for different kinds of overlaps.

Overlapping Types	**GPM datasets**		**ISB datasets**	
	Average number	**Average Ratio**	**Average number**	**Average Ratio**
b-ion, ∅, 1←→ y-ion, ∅, 1	0.213	0.015	0.154	0.011
b-ion, ∅, 1←→ y-ion, ∅,	0.203	0.015	0.173	0.012
b-ion, ∅, 1←→ y-ion, -H_2O, 1	0.307	0.023	0.307	0.023
b-ion, ∅, 1←→ y-ion, -NH_3, 1	0.199	0.014	0.129	0.008
y-ion, ∅, 1←→ b-ion, ∅, 2	0.094	0.006	0.110	0.008
y-ion, ∅, 1←→b-ion, -H_2O, 1	0.095	0.006	0.220	0.014
y-ion, ∅, 1←→ b-ion, -NH_3, 1	0.090	0.006	0.199	0.012
b-ion, ∅, 2←→ y-ion, ∅,	0.336	0.024	0.331	0.024
b-ion, ∅, 2←→ y-ion, -H_2O, 1	0.152	0.000	0.128	0.000
b-ion, ∅, 2←→ y-ion, -NH_3, 1	0.255	0.017	0.340	0.021
y-ion, ∅, 2←→ b-ion, -H_2O, 1	0.143	0.010	0.124	0.008
y-ion, ∅, 2←→ b-ion, -NH_3, 1	0.000	0.000	0.000	0.000
b-ion, -H_2O, 1←→ y-ion, -H_2O, 1	0.213	0.015	0.154	0.011
b-ion, -H_2O, 1←→ y-ion, -NH_3, 1	0.125	0.009	0.269	0.018
y-ion, -H_2O, 1←→ b-ion, -NH_3, 1	0.099	0.007	0.075	0.005
b-ion, -NH_3, 1←→ y-ion, -NH_3, 1	0.213	0.015	0.154	0.011
All	2.735	0.192	2.864	0.196

Experiments were also performed with random introduction of noise into theoretical spectrum. Results indicate that there is a significant increase in the number of overlap

instances, which are not realistic. Therefore, assuming no anti-symmetric problem is also not realistic, especially for noisy spectra.

In Lutefisk [28], the anti-symmetric problem is assumed to be non-existent, and a peak can be annotated as different ion types. In the Sherenga algorithm [18], only one ion type is possible for each peak, but the exact algorithm that solve the anti-symmetric algorithm is not described. The dynamic programming algorithm for solving anti-symmetric problem is described in [37, 38], and suboptimal algorithm that gives the suboptimal results for the anti-symmetric problem is shown in [39].

Since our experiments have shown that neither of the two approaches to the anti-symmetric problem is realistic, these simple models may be the obstacles for further improvement of these algorithms. Therefore, we propose a more realistic computational model to address the anti-symmetric problem.

3.7.2 New computational models and algorithm

We proposed a new algorithm that is based on two new computational models: 1) preprocess that can remove noisy peaks while introduce "pseudo peaks" into the spectrum; and 2) new anti-symmetric model that is more flexible and realistic to the anti-symmetric problem

Preprocess to remove noisy peaks and introduce pseudo peaks

In the binning process, since the masses of amino acids are at least of 1.0 Da difference (except for (I, L) and (Q, K), which can not be distinguished by any *De Novo* peptide

sequencing algorithms without using isotop information); the value of mass tolerance m_t* is set to be 0.5 Da, and the mass range of bin m_{bin} is set to be 0.25 Da (according to Lemma 1). With the process of binning, many noisy peaks are also removed from spectrum. Therefore, later processes can be even more accurate (lemma 1 shows that there is no loss of accuracy) as well as more efficient because less peaks are considered.

After binning, the "pseudo peaks" are introduced into every empty bins, and each of them are of 1/10 intensity of the lowest intensity in original spectrum.

After binning the peaks and introduction of "pseudo peaks", the support scores are computed for every bin. Here, we transform each of the bins (peaks) into vertices in the extended spectrum graph $G_1(S^{\alpha}_{\beta})$, and then score each of the vertices. Define $N_{support}(v_i)$ as the number of v_j ($v_j \neq v_i$), where $PRM(v_j) = PRM(v_i)$. Define the intensity function as $f_{intensity}(v_i) = \max(0.01, \log_{10}(intensity(v_i))$, so that $f_{intensity}(v_i)$ can not be less than 0. Let L be the total number of incoming and outgoing edges for v_i, and a_j be the amino acid for the edge (v_i,v_j) (or (v_j, v_i)). Then $\sum|(PRM(v_j)-PRM(v_i)|-m(a_j)|/L$ is the average mass error for v_i. To avoid "divide-by-zero" error in calculating the weight function, we define error function as $f_{error}(v_i) = \max(0.05, \sum|(PRM(v_j)-PRM(v_i)|-m(a_j)|/L)$. The definition ensure that $f_{error}(v_i)$ is larger than 0.05, a reasonably small error value. Then the score of vertex v_i in $G_1(S^{\alpha}_{\beta})$ is defined as

$$w(v_i) = \frac{N_{\text{support}}(v_i) + f_{\text{intensity}}(v_i)}{f_{\text{error}}(v_i)} \tag{9}$$

Note that this is different from (4) used in GBST algorithm. For each bin, the support score is computed and ranked.

Some of the actual peaks that are highly likely to be noise are deleted, and some of the pseudo peaks highly likely to represent ion types are kept. By this means, we can not only pruned out noise in the spectrum, but also introduce meaningful peaks into the spectrum. So we may create a better spectrum graph to process. Based on the analysis of the scores of peaks in the spectrum (details not shown here), the lowest 20% bins in scores ranking, or those bins with scores less than 1% of the highest ones are filtered out.

The Anti-symmetric Problem

Since a significant ratio of peaks in spectrum can be (correctly) annotated as different ion types, the anti-symmetric rule should not be strictly followed. Otherwise, there is loss of information. However, since there are still quite some noisy peaks after preprocessing, algorithms that do not consider anti-symmetric problem may also be misled by noisy peaks, and thus are not preferred. Thus, it would be better if a more flexible and less strict anti-symmetric rule is applied on the spectrum for anti-symmetric problem.

We have proposed the *restricted anti-symmetric model*. In this model, restricted number (r) of peaks can have different ion types. It is easy to observe that the current two approaches for anti-symmetric problem can be described by this model. The approach that do not consider the anti-symmetric problem is the one with r=number of peaks, and the approach that applied the anti-symmetric rule is the one with r=0.

Our restricted anti-symmetric model is based on the extended spectrum graph $G_i(S^{\alpha}_{\beta})$ [3] model using multi-charge strong tags in the spectrum. The principle of the *restricted anti-symmetric model* is that if a tag T_i in $G_i(S^j_k)$ is of high score, and on this tag, the number

(r) of overlapping instances (an instance is represented as two vertices of different ion type for the same peak) is within certain tolerance (half of the length of tag), then T_i is a good tag in $G_i(S^{\alpha}_{\beta})$, and it is selected for subsequent process.

It is easy to see that preprocessing and the restricted anti-symmetric models can be applied on any *De Novo* peptide sequencing algorithms to improve the accuracies (details in experiments). Below we describe our novel algorithm based on these two models.

Novel Peptide Sequencing Algorithm

The novel algorithm is based on our previously introduced GST-SPC algorithm [4] that has good performance. We emphasis again that in the first phase, the GST-SPC algorithm computes a set of tags - the set of all multi-charge strong tags (corresponding to tags of maximal length in extended spectrum graph) - and this leads to an improvement in the sensitivity that can be achieved. In the second phase, the GST-SPC algorithm try to link these tags, and computes a peptide sequence that is optimal with respect to shared peaks count (SPC) from all sequences that are derived from tags. The GST-SPC performs comparable to or better than other *De Novo* sequencing algorithms (Lutefisk and PepNovo), especially for multi-charge mass spectra.

In the novel algorithm, all of the peaks of the spectrum are binned, with each bin of the mass range m_{bin} (0.25 Da). The "pseudo peaks" are introduced into every empty bins. Bins (vertices in extended spectrum graph) that have very low scores or low support rank are filtered out.

In GST-SPC algorithm, we note that all of the strong tags can have their SPC computed before forming the paths in the spectrum. So in the novel algorithm, after strong tags are generated in the extended spectrum graph $G_1(S^{\alpha}_{\beta})$, we have filtered out the tags that violate the "*restricted anti-symmetric rule*". For the restricted anti-symmetric model on tags, we restricted r to be at maximum half the length of that tag. We have then computed the SPC for those "good" tags. Then a variant of width first search algorithm is applied on $G_1(S^{\alpha}_{\beta})$ to find paths from v_0 to v_M, so that these paths have high SPC, and they are consistent with *restricted anti-symmetric model*. Since the number of tags is small, such algorithm is efficient. A flowchart of the whole algorithm is illustrated in Figure 15.

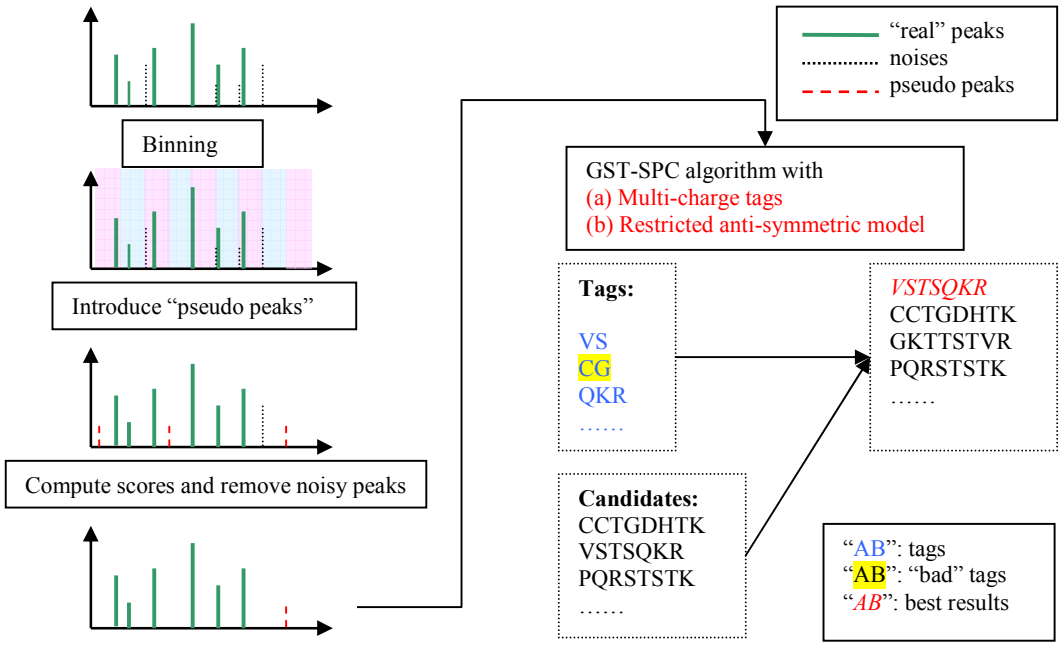


Figure 15. Flowchart of the whole algorithm. The preprocess model is illustrated at left, and the restricted anti-symmetric model is applied on the GST-SPC algorithm as shown at right. "bad" tags are tags that violate the restricted anti-symmetric model.

3.7.3 Experiments

Experiment Settings

All of the experiments in this project are performed on a PC with 3.0 GHz CPU and 1.0 GB memory, running Linux system. Our algorithm is implemented in Perl. We have also selected Lutefisk [28] and PepNovo [19], two algorithm with freely available implementations, for analysis and comparison. The best results given by different algorithms are used for analysis.

We have used spectra datasets described in Table 1. For measurement of the sequencing performance, we have adopted the measurements (5)-(8).

Results

We have first analyzed the performance of preprocess method, and compared the results with results from Lutefisk and PepNovo. We have also compared these results with theoretical *upper bounds*. The GPM and ISB spectra data are categorized by charges (given by spectrum data). The results are shown in Table 9. Note that GST-SPC without preprocess are shown previously, but for easy reading, I also put these results here.

Table 9. The performance of preprocess. The accuracies in cells are represented in a (specificity/sensitivity) format. "-" means that the value is not available by the algorithm, and "*" shows the average values based on charge 1 and charge 2 spectra.

Dataset	**No. of spectrum**	**Upper Bound**	**Lutefisk**	**PepNovo**	**GST-SPC (without preprocess)**	**GST-SPC (with preprocess)**
GPM						
Charge 1	756	1.00/0.44	0.261/0.258	0.322/0.186	0.369/0.378	0.395/0.381
Charge 2	874	1.00/0.52	0.243/0.241	0.316/0.215	0.321/0.365	0.334/0.385
Charge 3	454	1.00/0.38	0.111/0.113	-	0.291/0.291	0.312/0.327
Charge 4	207	1.00/0.36	0.065/0.063	-	0.219/0.226	0.230/0.229
Charge 5	37	1.00/0.29	0/0	-	0.192/0.191	0.195/0.190
Total	2328	**1.00/0.41**	**0.203/0.202**	**0.319/0.202***	**0.312/0.336**	**0.345/0.360**
ISB						
Charge 1	16	1.00/0.55	0.127/0.130	0.630/0.769	0.370/0.464	0.390/0.473
Charge 2	489	1.00/0.54	0.033/0.034	0.481/0.445	0.360/0.347	0.411/0.398
Charge 3	490	1.00/0.46	0.002/0.002	-	0.360/0.453	0.408/0.496
Total	995	**1.00/0.50**	**0.019/0.020**	**0.486/0.455**	**0.360/0.401**	**0.409/0.447**

From the results, we have observed that preprocess to remove the noise can effectively increase the sequencing accuracies. Compared with the results from original GST_SPC without preprocess, both of the specificity and sensitivity accuracies increase by about 8% for GPM datasets, and about 5% for ISB datasets after preprocess. This difference is probably due to the fact that ISB spectrum has more noise in it than GPM spectrum, so after preprocessing to filter out noise, ISB spectra still have more noise. Such accuracies are much superior to results from Lutefisk algorithm, especially on spectrum with high charges ($z \geq 3$). The new algorithm outperforms the PepNovo algorithm on GPM datasets; and for ISB dataset, the accuracies are closer. Compared with theoretical upper bounds, we can see that there is still much room for improvements.

We have also applied Lutefisk and PepNovo algorithms on preprocessed spectrum datasets. Since each of the preprocessed results is still a set of peaks, the application of these algorithms is easy. Results (Table 12) show that by removing noisy peaks, preprocess can also increase the sequencing accuracies for these algorithms.

We have then performed analysis of new anti-symmetric model (restricted anti-symmetric). All of the results based on GST-SPC algorithm are preprocessed. The results based on restricted anti-symmetric model are compared with the results based on strict anti-symmetric rule (strict anti-symmetric) and results from GST-SPC which do not consider anti-symmetric issue (no anti-symmetric). The results are shown in Table 10.

Table 10 shows that the restricted anti-symmetric model has superior accuracies. Compared with the results from algorithms which do not consider anti-symmetric issue

(no anti-symmetric), the application of restricted anti-symmetric model can improve the accuracies by about 5%, and this is probably due to the fact that restricted anti-symmetric model can remove some "bad" tags. About 2% to 5% improvements is observed when compared with the results from strict anti-symmetric model, this is consistent with the results of significance of the anti-symmetric problem in Table 8. The results also show a great improvement in tag specificity and tag sensitivity by using the restricted anti-symmetric rule, especially on ISB datasets. This may also be caused by the restricted anti-symmetric model that removes the "bad" tags. Further more, we have observed that sensitivity and specificity values approximate the normal distribution.

Table 10. The results based on the restricted anti-symmetric model, compared with other models. The accuracies in cells are represented in a (specificity/sensitivity[tag-specificity/tag-sensitivity]) format.

Dataset	**No. of spectrum**	**GST-SPC (no anti-symmetric)**	**GST-SPC (strict anti-symmetric)**	**GST-SPC (restricted anti-symmetric)**
GPM				
Charge 1	756	0.395/0.381 [0.131/0.130]	0.394/0.399 [0.144/0.142]	0.398/0.342 [0.144/0.145]
Charge 2	874	0.334/0.385 [0.142/0.160]	0.348/0.386 [0.130/0.158]	0.345/0.408 [0.151/0.159]
Charge 3	454	0.312/0.327 [0.077/0.091]	0.320/0.342 [0.078/0.090]	0.332/0.351 [0.079/0.096]
Charge 4	207	0.230/0.229 [0.043/0.042]	0.238/0.238 [0.043/0.041]	0.241/0.239 [0.046/0.045]
Charge 5	37	0.195/0.190 [0.020/0.027]	0.197/0.195 [0.026/0.025]	0.208/0.201 [0.028/0.029]
Total	2328	**0.345/0.360 [0.116/0.146]**	**0.344/0.364 [0.123/0.155]**	**0.347/0.375 [0.129/0.158]**
ISB				
Charge 1	16	0.390/0.473 [0.120/0.132]	0.386/0.486 [0.121/0.132]	0.393/0.491 [0.161/0.160]
Charge 2	489	0.411/0.398 [0.096/0.072]	0.414/0.397 [0.090/0.076]	0.434/0.421 [0.119/0.121]
Charge 3	490	0.408/0.496 [0.101/0.145]	0.426/0.528 [0.115/0.156]	0.419/0.531 [0.117/0.164]
Total	995	**0.409/0.447 [0.109/0.120]**	**0.419/0.464 [0.118/0.112]**	**0.427/0.475 [0.119/0.141]**

Compare the results in Table 10 with the results from Table 9, we have also observed that by the use of restricted anti-symmetric rule, the peptide sequencing results are more accurate. The GST-SPC with restricted anti-symmetric rule has results closer to accuracies of PepNovo, and significantly better than results of Lutefisk. We also note that these accuracy results are still about 20% (charge 1 and charge 2 spectrum) to 50% (charge 5 spectrum) less than the theoretical upper bounds of the accuracies given in [3].

We have also computed the number of results that are of 100% match with the correct peptide sequences (sensitiviety=1 and specificity=1). The results show that all of these algorithms that we have compared output more than 5% of 100% match results. For our novel algorithm which introduces "pseudo peaks", the problem that many of the missing fragmentations do not have enough peaks support still exists. We think that better scoring function can help to improve the ratio of 100% match results.

In Table 11, we have listed a few "good" interpretations of the novel algorithm, on which Lutefisk does not provide good results. It is interesting to note that more and longer peptide fragments are correctly sequenced by the novel algorithm - the power of preprocessing and the restricted anti-symmetric rule.

In these interpretations, we observe that the novel algorithm which incorporates preprocess and restricted anti-symmetric model can predict more and longer fragments of the correct peptides than Lutefisk, PepNovo and original GST-SPC. Specifically, for the peptide sequence "PAAPAAPAPAEKTPVKK", the two tags "*APAAPAPA*" and "*KK*" are both interpreted correctly only by this novel algorithm.

Table 11. Sequencing results of Lutefisk, PepNovo, GST-SPC and our novel algorithm. The accurate subsequences are labeled in italics. "M/Z"means mass to charge ratio, "Z"means charge, and "-" means there is no result.

M/Z	Z	Real	Lutefisk	PepNovo	GST-SPC	Novel Algorithm
1219.8	2	VAQLEQVYIR	[170.11]E*LEKVYLR*	GL*QLEQVYLR*	AVE*IEQVYIR*	VAAGKE*IEQVYIR*
1397.9	2	ELEEIVQPIISK	[242.14]*EEL*AVG[LP]*LSK*	*EELVKPLLSK*	*EIEEI*A[101.02]QH*ISK*	*EIEEI*GIIG*PISK*
1644.9	2	PAAPAAPAPAEKTPVKK	[AP]*AAPA*[HS]AP[198.14]*PA*AA[CS]	*AAPA*DFEAMTNLPK	*APAAPAPA*[56.06]APAMTKVPK	*APAAPAPA*F[51.14]APADHAAAP[8.00]*KK*
1838.8	3	SSYSLSGWYENIYIR	[172.09]L[303.17][243.13][NP][MT]*LYLR*	-	*SS*IYI[27.30]IIEPCE*IYIR*	
2000.2	4	PAAPAAPAPAEKTPVKKKAR	[323.09]R*PA*[AP]EKTN[LP]K[199.14]R	-	*APAAPA*MWNYNHKPYIR	*APAAPA*AAN[18.00]TNRGPCIIIWH[35.50]NR
1936.1	4	SIRVTQKSYKVSTSGPR	[199.14][PW][259.10]L[250.14]*KVSTSGPR*	-	VVIS*VTQK*[63.847]W*KVSTSGPR*	VVCP*VTQQ*[95.80]PG*KVSTSGPR*
2101.1	4	KIETRDGKLVSESSDVLPK	[243.09]*LV*R[TY]YT*SESS*AE[PV]R	-	IKQHTHECY*SESSDVIPK*	IKQHTHECY*SESSDVIPK*
3292.8	5	LLILEAGHRMSAGQALDHPWVITMAAGSSMK	[226.09]EL[NP][241.18][333.15][303.17][GP]ND[NM][228.08]	-	*IIE*IISH[1323.50]PPTGMTIT*SMK*	*IIE*IISSSH[1511.83]DDCHGCW[23.00]*SMK*
3752.0	5	LPPGEQCEGEEDTEYMTPSSRPLRPLDTSQSSR	-	-	*IP*VPAQV[1944.68]GRSPVQIC*SR*	*IP*VVGQV*E*[2025.98]GRSPVIKC*SR*
2359.0	5	CDKDLDTLSGYAMCLPNLTR	-	-	AFCDYA[417.18]RNQKIRCP*TR*	AFC*DID*[423.17]RNQKIRCP*TR*

We also applied preprocessing and restricted anti-symmetric model on other algorithms. We have selected PepNovo algorithm in this experiment. PepNovo takes input as the preprocessed spectra by our preprocess model, and output the tags. We have then rescored and rank these tags according to the restricted anti-symmetric model. We refer this method based on preprocess and restricted anti-symmetric check as PepNovo*.

Table 12. The performance of preprocess and anti-symmetric model on PepNovo. The accuracies in cells are represented in a (specificity/sensitivity) format.

Dataset	No. of spectrum	PepNovo	PepNovo with preprocess	PepNovo*
GPM				
Charge 1	756	0.322 / 0.186	0.320 / 0.190	0.330 / 0.201
Charge 2	874	0.316 / 0.215	0.319 / 0.221	0.333 / 0.221
Total	2328	**0.319 / 0.202**	**0.321 / 0.212**	**0.331 / 0.220**
ISB				
Charge 1	16	0.630 / 0.769	0.635 / 0.791	0.645 / 0.791
Charge 2	489	0.481 / 0.445	0.480 / 0.445	0.488 / 0.445
Total	995	**0.486 / 0.455**	**0.485 / 0.417**	**0.489 / 0.425**

The results show that upon the incorporation of preprocessing, the accuracies of PepNovo can be improved, but not substantially. Using preprocess and restricted anti-symmetric model together, the accuracies can be further improved. Therefore, we think that the

preprocessing and restricted anti-symmetric model can be applied on other algorithms to improve the accuracies of these algorithms.

Efficiency

With regards to the computational time and space, the novel algorithm can sequence each GPM spectrum (few peaks) in about 8 seconds, and each ISB spectrum (many peaks) in 20 seconds. This is slightly faster than the original GST-SPC algorithm, but slower than Lutefisk algorithm (within 10 seconds for these spectra) and PepNovo algorithm (about 10 to 15 seconds for these spectra Despite a reduction in the number of peaks by the preprocessing, overall computational time has increased due to more candidates being tested with the adoption of the restricted anti-symmetric rule. Because of the preprocessing, the space needed is less than the original GST-SPC algorithm. In general, the novel algorithm requires 20 MB memory to process one GPM spectrum, and about 50 MB memory to process one ISB spectrum, most of the which are used to store the extended spectrum graph.

3.8 Discussions

Multi-charge spectra have not been adequately addressed by many *De Novo* sequencing algorithms. In this series of projects, we first gave a characterization of multi-charge spectra and used it to analyze multi-charge spectra from GPM. Our results clearly show why existing algorithms do not perform well on multi-charged spectra.

We then present a simple *De Novo* sequencing algorithm (*GBST* algorithm) which makes use of extended spectrum graph and strong tags to predict peptides for spectra. GBST

algorithm not only works well for multi-charge spectra, but also performs well on single-charge spectra.

We have also proposed a novel algorithm, GST-SPC for *De Novo* sequencing of multi-charge MS/MS spectra. Our algorithm is based on the idea of using multi-charge strong tags to reduce the size of the candidate space to be searched. For a fixed set of strong tags, the GST-SPC algorithm optimizes the shared peaks count among all possible augmentations of the tags to form peptide sequences. The experimental results on ISB and GPM datasets show that GST-SPC is better than the GBST algorithm and Lutefisk. Against PepNovo; it performs better on GPM datasets and is worse on the ISB datasets. We have also derived the theoretical upper bound results for our algorithms.

However, it is interesting to note that none of these algorithms is close to the theoretical upper bound of the sensitivity (based on Δ^R restriction). This indicates that there is a possibility that there can be an algorithm based on MST that outperforms all of these algorithms.

We have also developed a database search algorithm for peptide sequencing using tandem mass spectrometry. The key steps of the algorithm are the selection of the tags from the spectrum of the peptide, and the approximate match of the PSP against the peptides in the database. Our algorithm does not need to compare the experimental spectrum to the theoretical spectrum of the peptides in the database; and in most of the cases, it does not even need to check all of the peptides in the database. Experiments show that our algorithm is comparable to or more accurate than other database search algorithms, including those based on tags. Since our algorithm can output results that

contain uninterrupted mass values, it has the potential to cope with the post-translational modifications.

I have also addressed two important issues in peptide identification, which are encountered in both *De Novo* and database search approaches. The first one is the preprocessing computational model that removes noisy peaks from spectrum while simultaneously introducing "pseudo peaks" into the spectrum. We have shown by the analysis of peaks that there are many noisy peaks in the spectrum, and that our preprocessing can make peptide sequencing faster, easier and more accurate. The second issue is about the anti-symmetric problem. We have shown that both using strict anti-symmetric rule and not considering anti-symmetric problem are not realistic, and we have proposed a restricted anti-symmetric model. Both models can help improve accuracies of *De Novo* algorithms, and the novel algorithm that incorporates these models is shown to have high performance on the datasets examined.

However, there is still a gap between the accuracies of novel algorithm and the theoretical upper bounds [3], and the algorithm can still be improved. This can be done by using a better scoring function (rather than SPC), a better preprocessing method, and more adaptable anti-symmetric model.

Chapter 4

Peptide Identification Algorithms Based on Tags, SOM and MPRQ

We emphasized that in the peptide identification problem, database search algorithms usually return the peptide sequences that match the parent mass of the spectrum. However, the accuracy depends on the quality of the database, and the process is slow (usually a few minutes). The *De Novo* algorithm can find tags with high accuracy [2, 3], and the process is fast (always within 1 minute) but tags are usually not complete sequences for the spectra. Hence, how to achieve a *balance between identification efficiency and accuracy* for peptide identification by tandem mass spectrum is an important consideration, and is the focus of the following series of projects.

The above mentioned peptide identification algorithms are still in the traditional framework, in which experimental spectrum (or tags from experimental spectrum) is compared against peptide sequences (in database or virtual database). I have proposed novel peptide identification algorithms that are not within this framework. In these algorithms, the experimental spectrum is converted to vector in high dimensional feature space, and then converted to points on 2D plane. The peptide sequences are also converted to vectors in high dimensional feature space and then to points on 2D space. By this way, the similarity of spectrum is converted to similarity of vectors and then to the neighborhood of points on 2D plane. Thus, the peptide identification problem has been transform to the vector comparison problem. For more accurate identification, we have also compared the candidate peptides with tags and experimental spectrum.

4.1 SOM and Multiple Point Range Query

SOM is an unsupervised machine learning technique that can transform high-dimensional vectors to 2D points on a plane. In the training process, a SOM (map) is built and the neural network organizes itself using a competitive process. The SOM usually consists of a two-dimensional regular grid of nodes. The node whose weights are closest to an input vector *V*, termed the best-matching or winner node, is updated to be more similar to *V* while the winner's neighbors are also updated (to a smaller extent) to be more similar to *V*. As a result, when a SOM is trained over a few thousand epochs, it gradually evolves into clusters whose data (in our case, peptides) are characterized by their similarity. Increasingly, SOM is used as an efficient and powerful tool for analyzing and extracting a wide range of biological information as well as for gene prediction [40-42]. The SOM is useful for peptide identification because it serves two purposes: dimensionality reduction and clustering. SOM can reduce high-dimensional data into a grid of nodes (i.e. usually a 2D map) yet preserve the "similarity" of the original data by projecting them onto clusters of points with close metric (Euclidean) distance. In short, spectrum similarity could be transformed to vector similarity (SOM data) and then to 2D points metric distance. Subsequently, MPRQ works on the 2D points to efficiently identify candidates that are similar to query spectra. Though there are other machine learning methods that serve similar purposes, SOM is chosen for peptide identification because SOM is proven to be effective for similarity search [43], and the number of candidate peptides can be easily controlled by adjusting search distance *d* (introduced in MPRQ).

The MPRQ technique is used for multi-point query on a 2D plane. The general idea behind MPRQ is to perform only one pass of the R-tree while simultaneously process

multiple query points (transformed from experimental spectrum). The R-tree is widely used as a data structure for indexing 2D points. Each node of an R-tree is represented by a minimum bounding rectangle (MBR) that bounds the location of its children (of smaller MBRs) until the leaf level where the actual 2D points are stored. At each MBR node R in the R-tree, the MPRQ algorithm processes all the children of R against all the query points. MPRQ takes $O(\log_B n + k/B)$ time using bulkloaded R-trees (such as STR [44]) which has a bounded height of $O(\log n)$, where m is the number of query points, n is the total number of points in the plane, B is the disk block size, and k is the number of results found. The key observation is that when search proceeds down the R-tree, the number of query points to be processed at each node also decreases rapidly (since the MBR is much smaller).

For peptide identification, after the theoretical spectra for the peptide sequences in the database are mapped as 2D points on a SOM map, we can transform the query (experimental) spectra into query points in 2D plane and proceed to query. It is possible to use many experimental spectra as the query, which translates to multiple points in 2D plane as the input for MPRQ algorithm. Apart from a set of query points, the MPRQ algorithm also accepts as input a parameter d that controls the radius of the search distance. The larger the value of d, the more candidate peptides will be returned. MPRQ can efficiently process the multiple input points *simultaneously* with respect to d and the MBRs during query, effectively perform configurable multi-spectra similarity search on database of known peptides.

4.2 Brief Review and My Work

This series of projects focus on how to achieve a *balance between identification completeness, efficiency and reasonable accuracy* for peptide identification by tandem mass spectrum. In this series of projects, we have used tags, SOM and MPRQ techniques for accurate peptide identification.

We have already reviewed InsPecT and some other database search algorithms based on tags. Recently, a coarse filtering method commonly associated with database search techniques was also introduced for peptide identification [45]. The spectra are converted to vectors; and then by using a metric distance-based indexing algorithm, initial candidates are produced for fine filtering later. A modified shared peaks count (SPC) scoring function was used to compute similarity among spectra. The coarse filtering can reduce the number of candidates to about 0.5% of the database. For fine filtering, a Bayesian scoring scheme is then applied on candidate spectra to more accurately identify peptide sequences. These two algorithms are similar in that they first choose a set of candidates, and then use fine scoring function to score and rank these candidates.

While algorithms based on tags can achieve reasonable accuracy and efficiency, they cannot guarantee the completeness of the results. This is because the completeness of the results (either *De Novo* or database search) is dependent on the quality of the tags which in turn is highly dependent on the quality of the spectra. On the other hand, filtering algorithms can achieve completeness and efficiency, but with less than satisfactory accuracy. This is because such spectrum comparison algorithms cannot adjust well to low-quality spectra, especially those with PTMs.

Therefore, we proposed novel peptide identification approachs that are based on multi-charge strong tags, SOM and MPRQ techniques. In these algorithms, the experimental spectrum is converted to vectors in high dimensional feature space, and then converted to points on 2D plane. The peptide sequences (transformed to theoretical spectra) in database are also converted to high dimensional feature space and then to points on 2D space. In this way, the similarity of spectrum is converted to similarity of vectors and then to the neighborhood of points on 2D plane.

In the first project based on using SOM and MPRQ [7], we analyzed the feasibility of transforming mass spectrum to vector and then to point on 2D plane. We proposed a simple algorithm PepSOM, and analyzed its performance. In this project, we emphasize on the *balance* between identification completeness and efficiency with reasonable accuracy for peptide identification by tandem mass spectrum. Our algorithm works by converting spectra to vectors in high-dimensional space, and subsequently use self-organizing map (SOM) and multi-point range query (MPRQ) algorithm as a coarse filter to reduce the number of candidates. This way, the efficient and completeness of the results of database search are achieved. The candidates are then compared with experimental spectrum by SPC to ensure accuracy.

In the second project on using tags and SOM [8], we emphasized on striking a *balance* between identification completeness, accuracy and efficiency in peptide identification by tandem mass spectrum. We again converted spectrum to vectors in high-dimensional space, and used self-organizing map (SOM) and multi-point range query (MPRQ) algorithm to reduce the number of candidate peptides. Then the candidate peptides are

scored by comparing with tags generated by *De Novo* algorithm. Experiments show that our algorithm is both fast and accurate in peptide identification. And our algorithm is also accurate for peptide identification with Post Translational Modifications (PTMs).

In the third project [9], we have emphasized on the identification of peptides with Post-Translational Modifications (PTMs). For identification of peptides with PTMs, traditional database search algorithms are constrained by known peptides, while traditional *De Novo* algorithms are limited by known amino acids, and both of them are also limited by known modifications.

In this project, we have proposed a novel algorithm (TagSOM) for peptide identification with PTMs. The algorithm first selected several important features of the spectrum that are less affected by PTMs (PTM-free features), such as highly reliable tags generated by *De Novo* algorithm. Based on these features, the algorithm transformed every putative peptide in database to high-dimensional vector, and then uses SOM to map these vectors to 2D plane. The algorithm then transformed every experimental spectrum to high-dimensional vectors according to these features, map them to the same 2D plane as query points, and perform MPRQ to retrieve a set of candidate peptides for each experimental spectrum. By comparing and validating these candidate peptides with tags and experimental spectrum, we expect that PTMs can be reliably identified. We are currently working on this project.

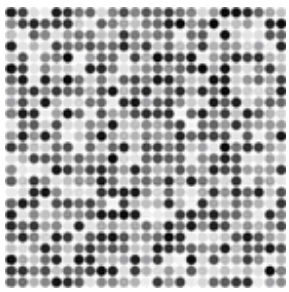
4.3 PepSOM Algorithm

The binning of the peaks, as well as the SOM and MPRQ techniques, have been described previously in “New Computational Models for Preprocess and Anti-symmetric

Problem" section.

4.3.1 The PepSOM algorithm

We propose a novel peptide identification algorithm in which candidate peptide sequences are first selected from database by SOM [43] and the MPRQ [46, 47] techniques, and then fine-filtered by comparing their theoretical spectrum with experimental spectrum by shared peaks count (SPC). More specifically, the theoretical spectra are binned to reduce the number of peaks in consideration. Then they are converted to high-dimensional vectors and trained with SOM algorithm to obtain a SOM (map). Each theoretical spectrum is then matched with the SOM map to obtain its best-matching node (expressed in (x,y)-coordinates) which forms the basis input map for the MPRQ algorithm. The experimental spectra are prepared similarly (binned, vectorized, matched; albeit without training) and the resulting coordinates form the input points for the MPRQ query. Figure 16 shows PepSOM as a coarse filtering step.



```
Coarse Filtering (DB, ES, d)
// input: peptide database DB, expt spectra ES, similarity d
// output: candidates peptides C
begin
  TS ← generate theoretical spectra for putative peptides in
DB;
  V1 ← Binning(TS);
  som_map ← TrainSOM(V1);        // SOM training
  2d_map ← MapSOM(som_map, V1);// map of (x,y)-coords
  V2 ← Binning(ES);
  Q ← MapSOM(som_map, V2);     // obtain multi points query set
  C ← MPRQ(2d_map, Q, d);      // get candidates set C from MPRQ
query
  return C;
end;
```

Figure 16. (left) In this example of a SOM, each spectrum is represented by a black dot. Neighboring dots have mutually similar shades of gray. Note that one node may represent overlapping spectra. (right) Our algorithm uses SOM and MPRQ for coarse filtering.

When these candidate peptides are retrieved, they are compared to experimental spectrum by SPC. The flow of PepSOM is illustrated in Figure 17 (a).

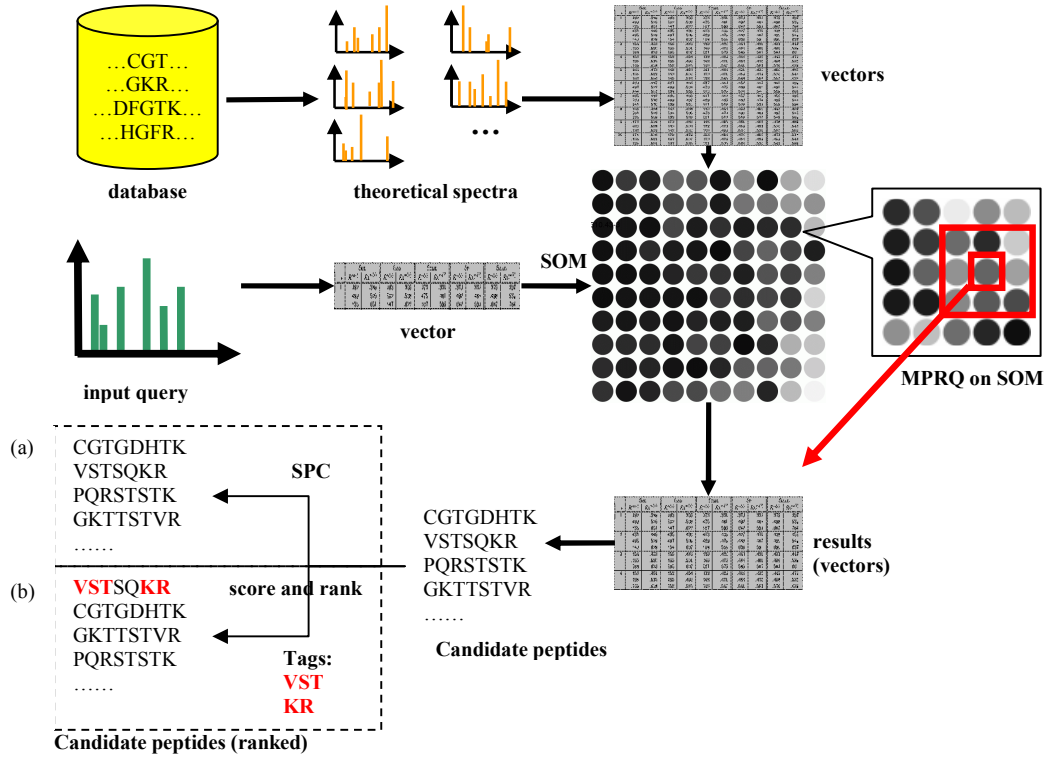


Figure 17. Diagram for the peptide identification with PepSOM. (a) SPC is used to score and rank candidate peptides. (b) Candidate peptides are scored and ranked by comparing with tags and experimental spectrum.

Although SOM has been used before for gene prediction [40], this is the first attempt of its kind to combine SOM with spatial database search for peptide identification. Many efficient algorithms exist for spatial database search in orthogonal 2D grids or hierarchical data structures. SOM is useful because we believe that by using SOM, the 2D distance between points on the map reflects the similarity of peptides. Combining SOM with MPRQ technique, peptides can be identified accurately and fast.

4.3.2 Experiments

Experiment Settings and Datasets

Experiments were performed on a PC with 3.0 GHz CPU and 1.0 GB memory, running Linux system. PepSOM was implemented in C++ and Perl. SOM_PAK [48] was the

SOM implementation that we have used. We have have selected two database search algorithms, Sequest [16, 17] and InsPecT [14]; as well as two *De Novo* algorithms with freely available implementations, Lutefisk [23] and PepNovo [19], for comparison and analysis. The best results (results with first rank) given by these algorithms were used for analysis.

Spectrum datasets were obtained from Open Proteomics Database [49], PeptideAtlas database [50] and Institute for Systems Biology (ISB) [27]. We will refer to these datasets as OPD, PeptideAtlas and ISB datasets for the rest of this project. The three datasets chosen are of vastly different sizes. We treated Sequest results (identified peptides) with cross-correlation score (Xcorr) above 2.5 as ground truth.

For OPD, the spectrum dataset used was opd00001_ECOLI, Escherichia coli spectra 021112.EcoliSol 37.1(000). The spectra were obtained from E. coli HMS 174 (DE3) cell, which is grown in LB medium until ~0.6 abs (OD 600). The spectra were generated by the ThermoFinnigan ESI-Ion Trap "Dexa XP Plus" and the sequences for these spectra were validated by Sequest algorithm [16, 17]. There are 3,903 spectra in total – of which 1573, 1165 and 1165 have parent charge $\alpha = 1$, 2 and 3, respectively. We have chosen all of the 202 spectra that were identified with Xcorr ≥ 2.5.

Spectra from PeptideAtlas database [50] were also selected. The spectrum dataset A8_IP were obtained from Human Erythroleukemia K562 cell line. Electrospray ionization source of an LCQ Classic ion trap mass spectrometer (ThermoElectron, San Jose, CA) was used, and DTA files were generated from the MS/MS spectra using TurboSequest. The dataset consists of a total of 1,564 spectra, in which there are 782 and 782 spectra for

parent charge α = 2 and 3, respectively. We have chosen all of the 44 spectra that were identified with Xcorr ≥ 2.5.

The ISB dataset was generated using an ESI source from a mixture of 18 proteins, obtained from ion trap mass spectrometry, and consists of spectra of up to charge 3. The ISB dataset was of low quality, having between 200-700 peaks each and an average of 400 peaks. The entire dataset consists of a total of 37,044 spectra. We have chosen all of the 995 spectra that were identified with Xcorr ≥ 2.5.

The databases that we used were peptides generated from the respective protein sequence datasets. Specifically, *E. coli* K12 protein sequences from OPD datasets, IPI HUMAN protein sequences from PeptideAtlas dataset and human plus control protein mixture from ISB dataset. As the number of protein sequences were very large for PeptideAtlas (60,090) and ISB (88,374) datasets, we used only the protein sequences corresponding to spectra identified with Xcorr ≥ 2.5 (our ground truth set). However, the number of extracted sequences is still very large because of many fragmentations.

Table 13. Parameters for the generation of databases and theoretical spectra.

Parameters	**Values**		
	OPD	**PeptideAtlas**	**ISB**
No. of protein sequences	4,279	31	3,553
Total database size	494,049	9,421	1,248,212
Test dataset size	202	44	995
Fragments mass tolerance	0.5 Da		
Parent mass tolerance	1.0 Da		
Modifications	–		
Charge	+2, +3		
Ion type	a, b, y, $-H_2O$, $-NH_3$		
Missed cleavages	0		
Protease	Trypsin		
Mass range	0-5000 Da		

The parameters for the generation of databases, test datasets and theoretical spectra are shown in Table 13. We used a search distance radius $d = 0.25$ as the MPRQ parameter.

The accuracy measures that we have used are (5)-(8), as we have described previously.

Experimental Results

We first analyzed the quality of peptide sequences identified by PepSOM (SOM and MPRQ) as candidates. We used a search distance radius $d = 0.25$ as the MPRQ parameter. Notice that similar spectra that correspond to the same 2D point can be losslessly retrieved by our algorithm since our algorithm has built an index for these overlapping spectra. In Table 14, the candidate peptides are scored and ranked by SPC only. The best-ranked result (highest SPC) among all candidates is labeled as first-rank peptide. It represents the peptide with theoretical spectrum that has the highest SPC against the experimental spectra. Best-match peptide refers to the peptide among all candidates that matches with its "real" peptide with the highest specificity (sensitivity).

Table 14. Statistical results on the quality of candidate identification by SOM and MPRQ. For "No. of Complete Correct" and "Complete Correct Accuracy", first-rank peptide was used for analysis. For specificity and sensitivity, the results for "first-rank peptide / best-match peptide" are shown.

Datasets	Database Size	Query Size	No. of Complete Correct	Complete Correct Accuracy	Sensitivity	specificity	Time (ms)
OPD	494,049	202	44	0.218	0.426 / 0.589	0.554 / 0.777	10.6
PeptideAtlas	9,421	44	10	0.227	0.440 / 0.632	0.330 / 0.368	10.5
ISB	1,248,212	995	116	0.117	0.672 / 0.723	0.521 / 0.879	10.8

From Table 14, it is clear that both the sensitivity and specificity of our algorithm using SOM and MPRQ is high. The sensitivity and specificity of best-match peptides are much higher that those for first-rank peptides, indicating that (i) SPC alone is not a good scoring function; and (ii) a properly designed scoring function can improve identification

accuracies significantly. Based on the results of best-match peptides, both sensitivity and specificity are higher than 0.55 for the OPD dataset; and specificity is higher than 0.70 for the ISB dataset. There are also a significant number (10% to 25%) of completely correct peptide identifications among first-rank peptides. These figures are comparable to PepNovo and InsPecT, and better than Lutefisk (details not shown). The average search time for each spectrum is less than 11 ms. This is comparable to InsPecT (with average 10 ms search time per spectrum with default settings, but based on smaller database), which is one of the fastest database search algorithms. Also, a large input (many queries) *does not* increase the overall query time by a lot. Such efficiency is due to the intelligent pruning rules embedded within the MPRQ algorithm.

Next, we compared PepSOM with other well-known peptide identification algorithms, namely Sequest [16, 17], Lutefisk [23], PepNovo [19] and InsPecT [14], among others. Recall that on these datasets, we treated Sequest results with cross-correlation score (Xcorr) above 2.5 as ground truth.

Table 15. Comparison of different algorithms on the accuracy of peptide identification. In each column, the "specificity / sensitivity" values are listed.

Datasets	Database Size	Test Size	InsPecT	Lutefisk	PepNovo	PepSOM
OPD	494,049	202	0.592 / 0.556	0.129 / 0.008	0.252 / 0.200	0.560 / 0.428
PeptideAtlas	9,421	44	0.811 / 0.402	0.162 / 0.063	0.291 / 0.135	0.334 / 0.445
ISB	1,248,212	995	0.602 / 0.633	0.032 / 0.032	0.563 / 0.593	0.529 / 0.680

We can observe from Table 15 that both specificity and sensitivity of PepSOM are better than Lutefisk and PepNovo (both *De Novo* algorithms), and comparable to InsPecT. Although InsPecT has higher specificity, our algorithm outperforms InsPecT in sensitivity. For the OPD dataset, both algorithms have specificity and sensitivity of about 0.55. For the PeptideAtlas dataset, the specificity of our algorithm is much worse than

that of InsPecT, but sensitivity is about 10% better. For the ISB dataset, PepSOM has lower specificity than InsPecT, but sensitivity is higher.

From these experiments, we note that the results for PepSOM are superior primarily because of the use of conventional SPC function. To conclude, we can say that PepSOM's performance is comparable to InsPecT in both accuracy and efficiency.

Efficiency

One of the most important features of our algorithm is speed. For batch processing of multiple spectra queries, we can see from Table 14 and Table 16 that our algotithm can complete peptide identification for large spectrum datasets (> 500 spectra) in less than 30 secs (e.g. for 500 spectra, 500 × 10.8 ms = 5.4 secs). In comparison, InsPecT takes about 10 ms on average to process one peptide. Comparing the three different datasets, we also observe that the increase in database size only affects the search time of our algorithm slightly, as each query takes about 10 to 11 ms on all three datasets.

Table 16. PepSOM-generated candidates' size, average query size and coarse filtering rate for each dataset.

Database	Database Size	Test Size	Candidates Size	Average Query Size	Coarse Filtering Rate
OPD	494,049	**202**	68,610	339.7	0.069%
PeptideAtlas	9,421	**44**	654	14.9	0.158%
ISB	1,248,212	**995**	101,443	102.0	0.008%

Traditional database search algorithms such as Sequest are much slower than PepSOM. Although *De Novo* algorithms are usually faster than PepSOM, they currently cannot generate results with comparable accuracy. In Table 16, candidates' size represents the combined total results from the coarse filtering of the database using the experimental spectra (test size) as the input query points for the MPRQ algorithm. Average query size

represents the average number of peptide sequence candidates for each spectrum (query point). Coarse filtering rate is computed by dividing average query size with database size. We only need to compare each spectrum against the candidates identified by MPRQ for it. Therefore, the coarse filtering rate is very low. Compared to the tandem cosine coarse filter used in [45] which filters to around ~0.5% of the database, it is obvious our method has a better filtering efficiency. This explains why PepSOM could achieve fast search time. From Figure 18 we find that the larger search distance radius d that we use, the larger the average query size (due to the increase of number of candidates); and the selection of $d = 0.25$ is a compromise between efficiency and accuracy. Accuracy generally improves slightly with larger values of d but the improvement is not significant.

For the calculation of processing time, note that SOM needs to preprocess the peptide sequences in the database prior to searching, just as InsPecT needs to transform the database to a trie data structure. Currently, the preprocessing time for PepSOM is a few hours for all the databases, the bulk of which is time taken to generate the coordinates of the best-matching node for all the peptides in the theoretical spectrum (the MapSOM step). The actual SOM training (the TrainSOM step) for our largest database, ISB, takes only about 15 mins while PeptideAtlas took less than 1 min to train.

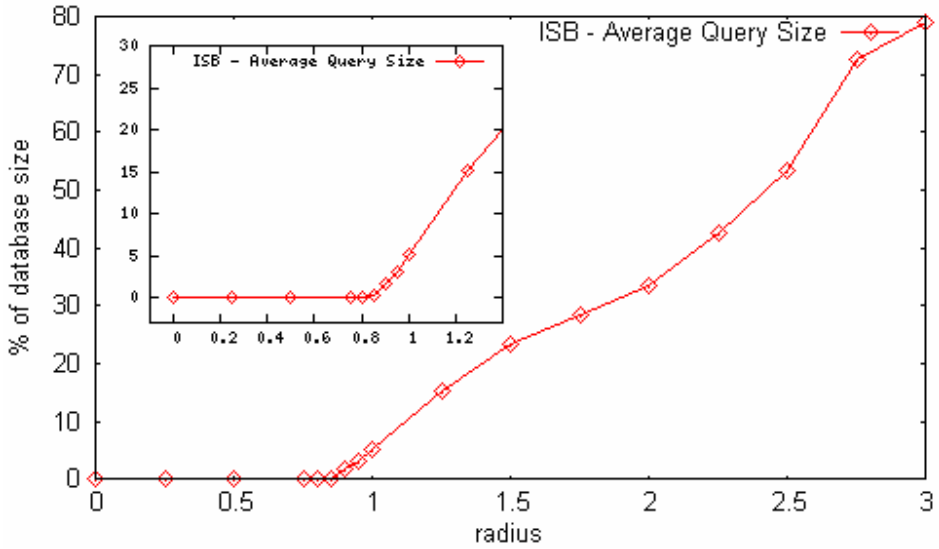


Figure 18: Average Query Size (search distance radius *d* vs % of database size) for the ISB dataset.

As for main memory requirements, we observe that InsPecT, for the sake of efficiency, requires a large amount of memory to store the trie data structure. The huge size of the sequence database also poses a challenge to us. However, in our algorithm, we can fragment the database, and subsequently transform each fragment using SOM on different workstations in parallel. This is much more efficient, especially when performed on a grid of workstations. As the input for MPRQ is a 2D map derived from SOM-trained spectra, it can handle a large amount of points with ease typical of any general database system.

4.4 Algorithm Based on Strong Tags and SOM

Previously, we have proposed PepSOM algorithm that is based on SOM and MPRQ, but not using any information of tags. In this algorithm based on tags, SOM and MPRQ, we have focused on how to achieve a *balance among identification completeness, efficiency and accuracy* for peptide identification by tandem mass spectrum with PTMs. This is an

especially important criterion for developing a successful tool to aid experts in analyzing results in the "wet laboratory".

In this project, we propose a novel peptide identification algorithm that is a combination of database search and *De Novo* approaches. It has the following steps: (i) both peptides in database and experimental spectra are first converted to high-dimensional vectors; (ii) the vectors are mapped to 2D plane with self-organizing map (SOM) [43]; (iii) the candidate peptide sequences are then selected from database with multi-point range query (MPRQ) [46, 47]; and (iv) these candidates are scored and ranked (fine filtered) by comparing them with the experimental spectrum as well as multi-charge strong tags generated by GST-SPC *De Novo* algorithm [4]. Steps (i)-(iii) can be regarded as the coarse filtering step, in which spectra similarity is transformed to vector similarity and then to 2D points metric distance similarity. By doing so, the completeness and efficiency are achieved. With the addition of (iv), the accuracy is also achieved. Our algorithm can also achieve high accuracy in identification of peptides with PTMs, as proven in our experiments.

4.4.1 Computational model and algorithm

In this project, we used multi-charge strong tags generated by the first phase of GST-SPC [4] since previous results show that the tags generated by GST-SPC are accurate ("GST-SPC Algorithm" section). In the following part of this project, we will refer to multi-charge strong tags as simply tags.

Before using SOM, binning is performed to convert peptides (transformed to theoretical spectra) in database to high-dimensional vectors in vector space. We have used the same binning scheme as described previously in the "New Computational Models for Preprocess and Anti-symmetric Problem" section.

For peptide identification, once the theoretical spectra for the peptide sequences in the database are mapped as 2D points on a SOM, we transform the query (experimental) spectra into query points in 2D plane and proceed to query. It is possible to use many experimental spectra as the query, which translates to multiple points in 2D plane as the input for MPRQ algorithm. Note again that, apart from a set of query points, the MPRQ algorithm also accepts a parameter *d* that controls the radius of the search distance as input. The larger the value of *d*, the more candidate peptides will be returned. MPRQ can efficiently process the multiple input points *simultaneously* with respect to *d* and the MBRs during query, effectively performing configurable multi-spectra similarity search on databases of known peptides.

Scoring and ranking

First we introduce *SPC score* and S_{tag} *score*: (a) The SPC score are computed as the number of shared peaks between experimental spectrum and theoretical spectrum of the identification results (within tolerance), over the number of peaks in theoretical spectrum. Note that SPC score differs slightly from SPC described previouslysince it is normalized. (b) The S_{tag} score, which measures the similarity of candidates to tags, is computed as the ratio of candidate peptide that can match one or more tags (at the correct position in the candidate, within the range of [0,100] Da), over the length of the candidate. For example,

given the candidate “VAQLEQVYIR” and two tags “VAK” and “IVYLR” appearing at the front and rear of the putative peptide, if we allow up to one mismatch, then the similarity is computed as (3+5)/10 = 0.8. To score and rank candidate peptides, we define and use a scoring function S_λ which is a weighted sum of the SPC score and the S_{tag} score against a set of tags. The values of the weights are derived empirically. Specifically, we found that $w_1 = 1$ and $w_2 = 10$ give discriminative results.

$$S_\lambda = w_1 \cdot SPC + w_2 \cdot S_{tag} \tag{10}$$

For PTM identification, it is observed that because of peptide fragmentation such as loss of water and ammonia, PTMs such as phosphorylation, as well as the experimental errors introduced by the mass spectrometer ion detector, mass shifts in spectra are very common. Specifically, each PTM corresponds to a set of shifted peaks in experimental spectrum. And highly possible PTMs should have strong support represented by such a set of mass shifts. In this project, we use a *modified SPC scoring function* (SPC*) that can better handle sets of mass shifts in spectra for identification of peptides with PTMs.

At each cleavage site, we assume any of $i*m_{bin}$ Da for all $0 \le i \le 100/m_{bin}$ (100 Da was determined empirically; details not shown) as a putative mass shift. We define $SPC_{i,j}$ as the SPC between experimental spectrum and theoretical spectrum of identified peptide P, where we assume a mass shift of $i*m_{bin}$ Da at cleavage site j of P. It is easy to see that $SPC_{0,j}$ is the SPC score of experimental spectrum with theoretical spectrum without mass shift at cleavage site j. If the largest $SPC_{i,j}$ for cleavage site j is obtained with $i > 0$, then this cleavage site j is a putative PTM site with mass shift of $i*m_{bin}$ Da, and the *PTM score*

$$S_{PTM}(j) = (SPC_{i,j} - SPC_{0,j}) \tag{11}$$

If S_{PTM} (j) is greater than a threshold T_{PTM} (determined empirically), then we say that this putative PTM site is significant, and we identify this as a PTM in peptide. We further define $SPC_{\{i1\ldots iq\},\{j1\ldots jq\}}$ as the SPC score between experimental spectrum and theoretical spectrum of identified peptide *P*, where mass shift of $\{i_1 * m_{bin} \ldots i_q * m_{bin}\}$ Da match with cleavage site $\{j_1 \ldots j_q\}$ of *P*, in which each S_{PTM} (j) is greater than T_{PTM}. And corresponding SPC* is defined as

$$\text{SPC*} = \sum_{j=1}^{K} (SPC_{(i_1\ldots i_q),(j_1\ldots j_q)} - SPC_{(i_1\ldots i_q),(0\ldots 0)}) \left| (S_{PTM}(j) > T_{PTM}) \right. \quad (12)$$

In which *K* is the length of the peptide. The modified S_λ score is then defined as

$$S_\lambda * = w_1 \cdot \text{SPC*} + w_2 \cdot S_{tag} \quad (13)$$

Which can be used for identification of peptides with PTMs. Apparently, PTMs are found at positions where tags do not match with candidate peptides, so for SPC* we do not consider those cleavage sites *j* that are covered by tags. Note that this is very different from [45], in which a fuzzy cosine distance is used on all of the peaks in the spectrum. What's more, in our SPC* function, a series of mass shifts caused by a single PTM is analyzed as a whole event, which is more realistic.

Our Algorithm

We propose a novel peptide identification algorithm:

1. Peptides from database arefirst transformed to vectors by binning
2. Candidate peptides are selected by SOM [43] and MPRQ [46, 47] given the experimental spectra;
3. Candidate peptides are scored and ranked by comparing them with the experimental spectrum and multi-charge strong tags generated by GST-SPC algorithm.

The theoretical spectra are binned to reduce noise and the number of peaks in consideration, and converting then to high-dimensional vectors at the same time. These vectors are input to the SOM algorithm to produce a SOM map. This way, each spectrum is mapped to 2D point (expressed in (x,y)-coordinates) on SOM map, which forms the input for the MPRQ algorithm. The experimental spectra are prepared similarly (binned and matched, but no training) and the resulting coordinates form the input points for the MPRQ query. Note that similar spectra may overlap on same 2D point, and our algorithm builds an index of all similar spectra on the same 2D point when retrieving candidates. Figure 16 shows coarse filtering step of our algorithm. After SOM and MPRQ, scoring functions are used to score and rank candidate peptides. The whole algorithm is similar to that in Figure 17 (b). The difference is that after SOM and MPRQ, for identifications of peptides, S_λ scoring function is used, for identification of peptides with PTMs, S_λ* scoring function is used.

4.4.2 Experiments

Experiment Settings and Datasets

The experiment settings are the same as in PepSOM.

The spectrum datasets and corresponding databases used are the same as those in PepSOM. For the ISB datasets, note that these ISB datasets were annotated by a few algorithms [14, 51] to be free of PTMs (refer to http://www.systemsbiology.org/extra/protein_mixture.html).

Also note that in these datasets, there may be different spectra corresponding to same peptides. But this will not artificially affect accuracies of different algorithms, since algorithms for peptide identification by mass spectrometry are essentially designed to identify spectrum-peptide correspondence.

The identification of PTMs is presently a very important issue in peptide identification. To analyze PTMs, we first performed experiments on experimental spectra *in silico* with artificially added PTMs (we call these simulated PTMs). We have selected spectra from ISB datasets as described above, and note that these spectra do not have any PTM annotations. For every peptide, the PTM that We have artificially added is phosphorylation for every amino acid involved. In the corresponding experimental spectrum, we shifted every peak that corresponds to the respective peptide fragment according to the restricted ion types Δ^R. Note that since our algorithm is not designed specifically for phosphorylation, it is can also be easily applied to detect other types of PTMs. Summary of modifications:

Modification	**Amino acid involved**	**Context**	**Mass difference (Da)**
Phosphorylation	T,S,Y	PTM	+79.97

We then performed experiments on the detection of PTMs on real spectra, using ISB spectra [27] that contain PTMs but are distinct from the modified ISB dataset we described above which does not. It was found that there are PTMs in these ISB datasets [51], and their identifications (called UCSD annotation) are available at (http://www.systemsbiology. org/extra/UCSD_supplemental_identifications.txt). There are 551 spectra with at least one PTM within these 2,799 ISB spectra. In our experiments, we evaluate if our algorithm can identify these annotated PTMs correctly.

To compare the different algorithms, the following accuracy measures were used:

$$\text{Recall} = \frac{\#\, correct}{|\,\rho\,|} \tag{14}$$

$$\text{Precision} = \frac{\#\, correct}{|P|} \tag{15}$$

where # *correct* is the "*number of correctly identified amino acids*". Two amino acids in the correct peptide ρ and the respective identification result P only contributes one count to #correct if they match (except (I, L), as well as (K, Q)) and their positions do not have a difference of more than 100 Da (determined empirically) and . *Recall* indicates the quality of the sequence results with respect to the correct peptide sequence - a high recall being that the algorithm recovers a large portion of the correct peptide. For a fair comparison with algorithms like PepNovo that only outputs the highest scoring tags (subsequences), we also use a *Precision* measure, which measures how many of the results are correct. Note that these recall and precision measures are different from (5), (6), since there is a position constraint on amino acids in (14), (15), rather than only using LCS in (5), (6).

Experimental Results

Firstly, we analyzed the quality of the tags that We have generated. These include the ratio of completely correct tags in the results, as well as recall and precision of tags. Results are shown in Table 17. Note that the results on OPD and PeptideAtlas datasets are not available in our previous section.

In Table 17, "No. of tags per spectrum" refers to the average number of tags generated per spectrum. "No. of complete correct per spectrum" measures the average number of tags identified that are completely correct (i.e, identified with 100% precision).

"Complete correct accuracy" is the ratio of completely correct tags to number of tags on average. We observe that more than 1/3 of the amino acids in real peptide sequences can be correctly identified by tags. Also, when the tags are generated, more than 70% of the tags are completely correct, showing that the tags generated are reliable. Since each tag is at least one amino acid in length, it can also be observed that a significant amount of tags are overlapping. The recall and precision results are obtained from tags by GST-SPC algorithm. Unfortunately, low recall for all datasets means that the sequencing results purely based on tags cannot cover the full length of the sequences. Therefore, in the following experiments, only the tags with the best scores (defined previously) are used for peptide identification.

Table 17. Statistical results on the quality of the generated tags.

Datasets	Query Size	Average Peptide length	No. of tags per Spectrum	No. of Complete Correct per Spectrum	Complete Correct Accuracy	Recall	Precision
OPD	202	10.14	7.42	6.01	0.81	0.43	0.43
PeptideAtlas	44	10.02	9.76	6.83	0.70	0.40	0.36
ISB	995	19.37	6.19	4.61	0.74	0.36	0.32

The quality of candidate peptides identified by SOM and MPRQ is already analyzed in PepSOM. Note that though in this work, the precision and recall are used with amino acid position constraint (instead of specificity and sensitivity used in PepSOM); for these candidates, the accuracies of results are only a little lower than those in analyses of PepSOM.

Another important question is: among the candidate sequences, how many of them are identical to the real peptide sequences. We have given the "complete correct accuracy" in Table 14. When we consider all of the candidates, the fraction in which the real peptide is

in the candidate sequences is much higher; for OPD dataset is 69.5%, PeptideAtlas 63.1% and ISB 65.3%. And if we allow up to two amino acids difference from real peptide sequences, the ratios increase to 80.1%, 85.3% and 78.6% respectively for OPD, PeptideAtlas, ISB datasets. Therefore, given a good scoring function, the peptide identification accuracy can be significantly increased. As the size of the candidate sequences generated by our algorithm is rather small (see Table 16), we believe these high ratios indicate good performance of the SOM and MPRQ for coarse filtering.

Subsequently, we compared our algorithm to other well-known peptide identification algorithms. For our algorithm, S_λ is used, and the results are based on peptides with the best score. The algorithms to be compared are Lutefisk [28], PepNovo [19] and InsPecT [21]. The best results (results with first rank) given by these algorithms were used for analysis. Note that since precision and recall is used, instead of specificity and sensitivity, the results is a little different from those in analysis of PepSOM.

Table 18. Comparison of different algorithms on the accuracies of peptide identification. In each column, the "precision / recall" values are listed.

Datasets	Database Size	Query Size	InsPecT	Lutefisk	PepNovo	Our algorithm
OPD	494,049	202	0.580 / 0.542	0.101 / 0.006	0.232 / 0.186	0.582 / 0.603
PeptideAtlas	9,421	44	0.801 / 0.389	0.149 / 0.057	0.275 / 0.128	0.521 / 0.457
ISB	1,248,212	995	0.584 / 0.621	0.011 / 0.022	0.548 / 0.561	0.594 / 0.695

We can observe from Table 18 that both precision and recall of our algorithm are better than Lutefisk and PepNovo (both *De Novo* algorithms). This is reasonable since *De Novo* algorithms do not utilize any information from databases. But even compare their results with the quality of tags generated by our algorithm (Table 17), we notice that the quality of tags generated by our algorithm is better than peptide identification results by Lutfisk, and comparable with those by PepNovo. Although InsPecT has higher precision, our

results outperform InsPecT in recall. Specifically, for the OPD dataset, both the algorithms have precision of about 0.58, but our algorithm has higher recall. For the PeptideAtlas dataset, the precision of our algorithm is much worse than that of InsPecT, but the recall is 17% better. For the ISB dataset, both InsPecT and our algorithm have similar precision, but recall of our algorithm is higher. These mean that our results can identify more portion of the real peptide.

Comparing Table 15 with the last column of Table 18, we have also observed that by scoring peptide candidates using S_λ, both precision and recall consistently increase (last column of Table 18), compared with only using SPC score (Table 15). This proves the superiority of S_λ scoring function.

PTM identification is of great importance to current mass spectrum analysis. Here, we used S_λ* to identify peptides with PTMs. Peptide identification accuracy is measured as the percentage of candidate peptides (search results) that contain the exact original (unmodified) peptide. PTM identification accuracy is measured as the percentage of search results in which the best-score PTM (definition in Section 2.4) identification is *correct*, where PTM identification is defined as correct if the original peptide is identified correctly *and* the putative PTM site difference from the real PTM site is not more than 100 Da. We reiterate that only when the original peptide is correctly identified will we consider the PTM identification. For example, a peptide (with PTM) "AS+80RK" is identified correctly, if "ASRK" is identified correctly by database search, *and* we have also identified the PTM site after "S".

We have analyzed the accuracies of PTM identification from spectra with simulated PTMs. The results of 995 ISB spectra with simulated PTMs are shown in Table 19.

Table 19. Accuracies (%) of PTM identification from simulated spectra by tags of different lengths. The columns with Top i = 1, 2, 3, 4 represent the (peptide / PTM) identification accuracies in Top i. "No limit" means that the best-score tags are used without any length limit. "Filtration ratio" is computed as the number of candidates after tag filtration over the number of candidates after MPRQ. "Time" is the total time to identify the peptides and PTMs for 995 spectra. Results without using tags are also illustrated.

Database Size	Query Size	Tag length	Top 1	Top 2	Top 3	Top 4	All	Filtration Ratio	Time (s)
1,248,212	995	3	46.7 / 30.2	50.1 / 36.3	62.6 / 40.5	69.2 / 46.5	**71.3 / 60.1**	0.0148	5.6
		4	34.6 / 56.9	40.5 / 25.6	44.4 / 32.6	51.0 / 39.0	**63.3 / 50.0**	0.0021	7.5
		No limit	46.8 / 32.9	52.0 / 36.1	58.3 / 43.3	64.4 / 50.1	**72.8 / 59.1**	0.0491	6.6
		No tag	31.7 / 26.4	35.5 / 26.6	41.1 / 35.2	46.9 / 39.5	**56.7 / 40.8**	–	10.7

From the results above, it can be observed that sequence tags of length 3 and 4 are able to further filter out candidates from the results of SOM and MPRQ. With reduced candidates, the accuracy for PTM identification increased. Compared with results without tags, the percentages of search results that contain the exact correct peptide are significantly higher. For example, for filtration with tags of length 3, about 46.7% to 71.3% of original peptides are identified correctly. Increase filtration tags length to 4 decreases peptide identification accuracies, but using filtration best-score tags without any length limit *do not* show such decrease. PTM identification accuracies show similar patterns. These indicate that although longer tags may have lower recall, the best-score tags are of high recall, regardless of their length. The filtration ratio is small, for instance the filtration ratio for tags with length 3 is 0.0148; for length 4 is 0.0021. This indicates that filtration by tags can further reduce the number of candidate peptides for further careful examination.

Experiments on the identification of PTMs on real ISB spectra with "UCSD annotation" were also performed. The results of the "UCSD annotation" were treated as ground truth. Since experiments on simulated PTMs (Table 19) show that best-score tags with no length limits have the best accuracies, we have used these tags here. Results show that the filtration ratio of our algorithm is 0.062. The peptide identification accuracies are 42.0, 45.7, 48.2, 50.6 and 55.5 for Top 1, 2, 3, 4 and All, respectively; and the PTM identification accuracies are 31.6, 33.1, 34.8, 40.2 and 41.8 for Top 1, 2, 3, 4 and All, respectively. These values are slightly smaller than those on simulated spectra, and we think this is due to the diversity of the PTM types in real spectrum.

Efficiency

One of the most important features of our algorithm is that it is very fast. Since the efficiency of this algorithm is essentially dependent on SOM and MPRQ query, the time and space efficiency of this algorithm is similar to that of PepSOM. So I do not describe its efficiency in more details in this section.

4.5 TagSOM Algorithm

Here I have focused on identification of peptides with PTMs. For PTMs in spectrum, many of traditional algorithms identify them by using a limited set of modifications [47, 85, 119, 156]. However, this approach is slow and erroneous on spectra with unknown modifications. Recently, there are quite some novel algorithms proposed [33, 52]. Specifically, [33]proposed a dynamic programming algorithm for blind search of PTMs. However, the large search space makes this algorithm inefficient. In [52], the tags are used to search for candidate peptides in database search by deterministic finite automaton,

and then a point process model is used for blind PTMs identification. This algorithm is efficient but its effectiveness for real PTMs identification is not clear.

In this project, we have proposed the TagSOM algorithm for peptide identification with Post-Translational Modifications (PTM). TagSOM is an algorithm that is the combination of database search strategy and *De Novo* strategy. TagSOM combines the highly reliable tags generated by *De Novo* algorithm, and reliable candidate peptides from database. The combination of tags and candidate peptides provides a basis for careful and extensive identification of PTMs. By comparing candidate peptides with tags and experimental spectrum, the putative PTMs can be identified.

TagSOM first selected several important features of the spectrum that are affected by PTMs very little (PTM-free features), such as highly reliable tags generated by *De Novo* algorithm. Based on these features, TagSOM transformed every putative peptide in database to high-dimensional vector, and then uses SOM to map these vectors to 2D plane. TagSOM then transformed every experimental spectrum to high-dimensional vectors using the same set of features, map them to the same 2D plane as query points, and perform MPRQ to retrieve candidate peptides for the experimental spectrum. By comparing and validating these candidate peptides with tags and original experimental spectrum, peptides with PTMs can be reliably identified.

Tags identification

The identification of tags in spectrum are examined in InsPecT [14] and GST-SPC [53] in "GST-SPC Algorithm" section. Generally, tags are putative subsequences of the original peptide sequence, which is strongly supported by a set of peaks in experimental spectrum.

Feature Selection

Feature selection from spectrum data are typically used for algorithms based on machine learning [54, 55]. In [55], a decision tree approach is proposed that identify peptide sequences based on peaks intensity. Recently, Arnold et. al. [54] has proposed a machine learning algorithm that uses more than 200 features to predict the peptide fragmentation patterns. Within these important features for spectrum, some of them are PTM-free features. PTM-free features refer to those features that are not affected (or affected much) by PTMs. For example, most of the highly reliable tags are PTM-free feature. PTM-free features are useful especially when we have to compare spectrum with peptides in database. This is because when transforming peptides in database to theoretical spectra, PTMs are not considered.

The theoretical spectrum of peptide sequence can be generated providing the restricted ion types Δ^R, or by peptide fragmentation prediction algorithm [54]. The theoretical spectrum can then be transformed to vectors according to the selected features. Since experimental spectra are transformed to vectors using the same set of PTM-free features, the peptide sequences can be transformed to vectors of the same format as experimental spectrum. Note that the features for fragmentation patterns (one source of PTM-free features) can be used to reliable predict the theoretical spectrum from peptide, and these features can also be used for analysis of experimental spectrum.

Once the theoretical spectra for the putative peptide sequences in the database are mapped to 2D plane by SOM, we also transform the query (experimental) spectra into query point(s) in 2D space and proceed to query. At this point, it is possible to use many

experimental spectra as the query, which translates to multiple points in 2D space as the input for MPRQ algorithm. Experiments showed that a large input (many points) *does not* increase the overall query time by a lot. This is due to the intelligent pruning rules embedded within the MPRQ algorithm. Apart from a set of query points, the MPRQ algorithm also accepts as input a parameter *d* that controls the radius of the search distance. The larger the value of *d*, the more results will be returned. MPRQ can efficiently process the input points with respect to *d* and the MBRs during the query. The correlation of nodes on SOM and peptide sequences can be retrieved simply by an indexing process.

Peptide and PTM identification

Peptides are identified by comparing candidate peptides with tags and experimental spectrum. Candidate peptides are retrieved from database, and they are compared with tags retrieved from experimental spectrum. The SPC and statistical analysis are also performed for comparison. By comparison, the candidate peptides are scored and ranked, with peptide of highest score be most putative peptide for experimental spectrum.

Those PTMs are at positions where tags do not match with candidate peptides, and has a set of shifted masses (note that b and y ions have different shift direction) compared with experimental spectrum. Highly possible PTMs should have strong support for such a set of mass shift.

4.5.1 Computational model and algorithm

Tags Identification

This step generates highly reliable tags by GST-SPC algorithm. Evaluation of the tags is based on scoring functions (defined previously) proposed in [53].

Selection of features and transform spectrum to vectors

PTM-free features are selected so that these features are affected by PTMs only a little; and they have high discriminative power when comparing theoretical spectrum with experimental spectrum.

We have first examined features mentioned in [54, 55], and select those features that are PTM-free. To facilitate the selection, Individual amino acids are encoded using binary data representation [56]. All of the candidate features are categorized as below.

(i) Spectrum based features: parent mass, parent charge, average intensity, intensity variations, different ions support, neutral loss support, isotope peaks, gas phase basicity, helicity, hydrophobicity

(ii) Environment related features: enzyme used for cleavage (trypsin)

(iii) Tag based features: tag, tag position, tag length, left flanking mass, right flanking mass, tag score.

To select PTM-free features, we have selected experimental spectrum from selected ISB datasets [27] with Xcorr ≥ 2.5 for analysis, and have chosen those ISB datasets that were annotated by a few algorithms [14, 51] to be free of PTMs (refer to http://www.systemsbiology.org/extra/protein_mixture.html). For each of the spectrum in the datasets, we have also generated a set of modified spectrum by artificially adding the PTMs. This way, we have generated a set of spectrum pairs; in which each pair contain a

spectrum without PTMs and the corresponding spectrum with additional PTMs (modified spectrum). The PTM-free features are essentially identified by comparing this pair of spectra. The specification of selected ISB datasets and the PTMs are listed in Table 20. Note that though we have selected only a few PTM types for artificial addition, they are enough to distinguish PTM-free features from other features, since PTM-free features should be discriminative enough.

Table 20. Specification of selected ISB datasets and the PTMs for analysis of PTM-free features.

Parameters	**Values**		
Dataset name	**ISB**		
Experimental Spectrum size	995		
Xcorr	3.0		
PTM type	phosphorylation	hydroxylation	oxidation
Amino acid involved	T,S,Y	P	M
Amino acid involved	+79.97	+15.99	+15.99

For every putative PTM-free feature F_l, we have checked if it is a significant feature for peptide and PTM identification. Suppose there are *N* spectra, and for each of them, we are given the corresponding peptide sequence of length *K*. For each pair of original spectrum and modified spectrum, we observe the likelihood of observing an original (or modified) fragment by the matched feature F_l, as well as the likelihood of observing a modified (or original) fragment by the mismatched feature F_l. We adopted a log odd ratio approach to combine these likelihoods. This way, we will be able to discriminate original and modified fragments (represented as PRMs), while not biased to short peptides. We define the significance score for the i-th fragment PRM_i for feature F_l, $ss_{i,l}$, as

$$ss_{i,k} = \log_{10}\left(\frac{p(original\ PRM_i \mid match\ F_l)}{p(\text{modified}\ PRM_i \mid match\ F_l)} * \frac{p(\text{modified}\ PRM_i \mid mismatch\ F_l)}{p(original\ PRM_i \mid mismatch\ F_l)}\right) \quad (16)$$

The significance score for the whole pair of spectrum for feature F_l, $SS_{j,l}$, is defined as

$$SS_{j,l} = \sum_{i=1}^{K} ss_{j_{i,l}} / K \quad (17)$$

And the significance score for this feature F_l is defined as

$$SS_l = \sum_{j=1}^{N} SS_{j,l} / N \quad (18)$$

Positive SS_l indicates that the feature is more likely to be a discriminative PTM-free feature than otherwise.

Based on this scoring function, we are currently examining all of the putative PTM-free features, and select those with positive SS.

The peptides in database are transformed to theoretical spectrum by these PTM-free features, and both theoretical spectrum and experimental spectrum are transformed to high-dimensional vectors based on these PTM-free features.

SOM and MPRQ to get candidate peptides for experimental spectrum

We have used the SOM and MPRQ techniques detailed in "PepSOM Algorithm" section.

Evaluate candidate peptides

We used scoring functions S_λ (10) and S_λ* (13), which are weighted sum of the SPC score and the S_{tag} score against a set of tags. These functions are detailed in "Algorithm Based on Strong Tags and SOM" section.

4.5.2 Experiments and current results

Experiment Settings and Datasets

We have used the same datasets as previously described. In addition, we have used spectrum datasets with real PTMs:

- ISB dataset: a public collection of MS/MS spectra [27] (http://www.systemsbiology.org/extra/protein_mixture.html). This data set was chosen as it has been queried extensively, but many spectra remain unannotated. Different from previous simulation experiments, we have analyzed all of the ISB spectra, and try to find out those modifications exist in experimental spectrum.

It is already discovered that there are some PTMs in ISB datasets [51], and these annotations (UCSD annotation) are available at http://www.systemsbiology.org/extra/UCSD_supplemental_identifications.txt. There are 551 spectra with at least one PTM in these 2,799 ISB spectra. We refer to this dataset as $PTM_{Real\text{-}ISB}$.

- Lens dataset: spectra acquired from human lens proteins [57]. A major component of the lens proteome is crystallins, which have very little turnover, and acquire modifications with age. When a person ages, the crystallins become insoluble, and the tissue increasingly opaque, often leading to cataracts. PTMs are known to play a major role in the process [57].

It is also known that there are many PTMs in Lens datasets [51], and some of these identifications by OpenSea Algorithm [58] are available at (http://medir.ohsu.edu/~geneview/publication/supplement_opensea/Opensea_Web_Supplement.html). Another high-confidence PTM annotation dataset on these Lens datasets are available at (http://bioinfo2.ucsd.edu). We refer to this dataset as $PTM_{Real\text{-}Lens}$.

In this project, we analyze and compare our PTM identification results with these published PTM annotations.

The details of datasets and references are listed in Table 21.

Table 21. Specification of the real datasets used for PTM identification.

Dataset	No. of spectrum	Database size	References
ISB	2,799	37 proteins (25kb)	[27]
Lens	8887	20 proteins (5kb)	[57]

We have used the same measurement of recall and precision as described previously in (16), (17).

Further more, to analyze the accuracies of identification of PTMs, we have introduced the precision and recall of PTM identifications.

$$Recall_{PTM} = \#\text{ correct PTMs} / \text{Total number of known PTMs} \quad (19)$$

$$Precision_{PTM} = \#\text{ correct PTMs} / \text{Total number of predicted PTMs} \quad (20)$$

In which # correct PTMs is "the number of known PTMs identified (according to [51])". The $Recall_{PTM}$ and $Precision_{PTM}$ reflect the accuracies of different algorithms on the dataset examined. However, these measurements can only be applied on peptides sequences with known PTMs.

Current Results

We are currently retrieving features and performing experiments on analyzing TagSOM algorithm. We will analyze the peptide and PTM identification accuracies by TagSOM, and compare these results with the results of other algorithms, such as InsPecT. We will also analyze the process time of TagSOM algorithm. Initial results indicate that the TagSOM algorithm is efficient; and based on our currently available features, the algorithm is accurate for identification of peptides and some known PTMs.

4.6 Discussions

Peptide identification by tandem mass spectrometry is a very important problem in proteomics. In these works, I have focused on the balance of identification completeness, efficiency and accuracy for peptide identification by tandem mass spectrum.

I have proposed a new computational model that transforms spectrum similarity to vector similarity, and subsequently to the neighborhood similarity of points on a 2D plane. Based on this, we proposed the PepSOM algorithm which first selects from database of all putative peptide sequences, and then transform them into vectors by binning. These vectors are then used for training by SOM and for querying by MPRQ, which together form a coarse filter for our approach. The resulting candidates are fine-filtered by comparing their theoretical spectrum against experimental spectrum using SPC.

Our experiments show that the accuracy of PepSOM is high. Many of PepSOM peptide identification results are identical with those identified by Sequest with high Xcorr score. These are better than or comparable to the results of the most accurate database search algorithms currently available (e.g. InsPecT). The algorithm is also efficient, especially for batch processing. However, like other database search approaches, the accuracy of our algorithm is dependent on the completeness of spectra database to some extent.

We have also proposed an algorithm that first selects all the putative peptide sequences from a database and transforms them into vectors via binning. These vectors are converted to SOM after which MPRQ is used to produce candidate peptides efficiently (same as PepSOM). Finally we fine-filter these candidate peptides by using a scoring function (S_λ for peptide identification, and S_λ* for PTM identification), to compare each

of them with experimental spectrum and highly reliable multi-charge strong tags generated by GST-SPC *De Novo* algorithm.

Our computational model combined database search to obtain candidate peptides with highly reliable multi-charge strong tags, effectively achieving a balance of identification completeness, accuracy and efficiency for the peptide identification problem by tandem mass spectrometry. Experiments indicated that our algorithm can achieve high accuracies, yet still maintaining fast, efficient processing, especially for batch processes. Another important feature of our algorithm is that our algorithm can handle the identification of peptides with PTMs with high accuracy.

In TagSOM project, we have proposed a novel algorithm, TagSOM, specifically for peptide identification with PTMs. The algorithm transformed peptides in the database, as well as experimental spectrum to high-dimensional vectors according to PTM-free features, and then use SOM and MPRQ to retrieve candidate peptides for experimental spectrum. These candidates are then compared with tags generated by GST-SPC *De Novo* algorithm, as well as with experimental spectrum by scoring function (S_λ for peptide identification, and S_λ* for PTM identification). Peptides and PTMs can thus be highly realizably identified. Experiments are now under way.

The TagSOM algorithm can be extended to become a more general algorithm by including spectrometry machine or environment dependent features such as the type of enzymes used, spectrometer measurement error tolerance and the analyzer type (ion trap, time-of-flight, etc.).

Chapter 5

Conclusions

This chapter presents a summary of my investigation in the algorithms for peptide and PTM identification problems. I have given the discussions for these works previously in their respective sections. Here, I will give an overall conclusion, and also discuss possible future research directions.

5.1 Summary

In this dissertation, I have concentrated on the problems of peptide and PTM identificantion. This includes some heuristic algorithms for identification of peptide sequences from mass spectrometry, with focus on multi-charge spectrum.

I have first introduced and analyzed the extended spectrum graph computational model. Based on this model, I have defined the "best strong tags" which are highly accurate, and later proposed the GBST algorithm based on best strong tags. Subsequently, I have extended the best strong tags to "multi-charge strong tags", and proposed the GMST and GST-SPC algorithms. The GST-SPC algorithm is also based on computing the SPC of the candidate sequences and experimental spectrum. A fast database search algorithm, PSP, is also proposed based on tags.

Then I have described algorithms that transformed spectrum to high dimensional vectors. Using the SOM and MPRQ technique, these algorithms then transformed the peptide sequence similarity to 2D point similarity on SOM map, and performed multiple simultaneous queries for candidate peptides efficiently. The first algorithm, PepSOM,

empirically proved the effectiveness of using SOM and MPRQ for efficient peptide identification. The second algorithm further improved PepSOM by scoring and ranking the candidate peptides by comparing them with tags generated by GST-SPC algorithm. This algorithm is also capable of PTM identification. The third algorithm, TagSOM, went a step further by using the information contained in these candidate peptides and tags specifically for the purpose of PTM identification.

5.2 Main Conclusion

Peptide and PTM identification are very important in bioinformatics research. My research in the area of peptide and PTM identification has contributed to the bioinformatics research.

My research in peptide identification has produced a number of fast and accurate database search and *De Novo* algorithms. I believe that these research works on peptide identification problems can help researchers in the qualitative and quantitative analysis of mass spectrometry data, and also help them to identify novel peptide sequences as well as novel post translational modifications (PTMs).

5.3 Future Research

For the work on peptides identification using mass spectrometry, I have completed projects on peptide identification by *De Novo* and database search algorithms. I have also analyzed the idea of combining the *De Novo* and database search strategies (SOM and MPRQ) to achieve a balance between identification efficiency and accuracy for peptide identification by tandem mass spectrum. The detection of PTMs is also investigated.

However, based on the analysis on characteristics of multi-charge spectra, we realize that there is still a big gap between the accuracies of the results of our current database search and *De Novo* algorithms and the upper bounds of accuracies for peptide identification. So we think that further investigation of peptide identification algorithms to improve the accuracies is possible and necessary. I have already investigated two issues that traditional peptide identification algorithms overlook, namely preprocess to remove noise and computational model for anti-symmetric problems. And we have shown that by using new computational models for these two issues, the accuracies of peptide identification algorithms can be improved. I think further scrutinization of these and other issues can further improve the accuracies of peptide identification.

For our algorithms based on tags, SOM and MPRQ, preliminary results have already shown that converting spectra to vectors according to their features can improve the accuracies of peptide identification. I think that the feature selection process for these algorithms can be further improved by using higher quality, more biologically meaningful and discriminative features for peptide identification. By performing peptide identification based on these features, I think that peptide and PTM identification can be more accurate. Moreover, there is reason to believe that peptide identifications can be independent of machines setting, since these settings can also be encoded as the features.

References

1. Gusfield, D.: Algorithm on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, New York, NY, USA (1997).

2. Chong, K.F., Ning, K., Leong, H.W., Pevzner, P.: Modeling and Characterization of Multi-Charge Mass Spectra for Peptide Sequencing. Journal of Bioinformatics and Computational Biology **4** (2006) 1329-1352.

3. Chong, K.F., Ning, K., Leong, H.W.: Characterisation of multi-charge mass spectra for peptide sequencing. The Fourth Asia-Pacific Bioinformatics Conference (APBC2006) (2006) 109--118.

4. Ning, K., Chong, K.F., Leong, H.W.: De Novo Peptide Sequencing for Mass Spectra Based on Multi-Charge Strong Tags. The Fifth Asia-Pacific Bioinformatics Conference (APBC2007) (2007) 287-296.

5. Ning, K., Chong, K.F., Leong, H.W.: Algorithm for peptide sequencing by tandem mass spectrometry based on better preprocess and anti-symmetric model. In preparation (2007)

6. Ning, K., Chong, K.F., Leong, H.W.: A Database Search Algorithm for Identification of Peptides with Multiple Charges using Tandem Mass Spectrometry. BioDM 2006, Vol. LNBI 3916 (2006) 2-13.

7. Ning, k., Ng, H.K., Leong, H.W.: PepSOM: An algorithm for peptide identification by tandem mass spectrometry based on SOM. The 17th International Conference on Genome Informatics (2006)

8. Ning, K., Ng, H.K., Leong, H.W.: An algorithm for peptide identification by tandem mass spectrometry with multi-charge strong tags and SOM. in preparation (2007)

9. Ning, K., Ng, H.K., Leong, H.W.: TagSOM: A Novel Algorithm for Peptide Identification with Post-Translational Modifications. in preparation (2007)

10. Ning, K.: Novel Algorithms for Sequencing of Peptides using Tandem Mass Spectrometry. Department of Computer Science **PhD thesis proposal** (2005)

11. Mann, M., Wilm, M.: Error-tolerant identification of peptides in sequence databases by peptide sequence tags. Analytical Chemistry **66** (1994) 4390-4399.

12. Tabb, D., Saraf, A., Yates, J.r.: GutenTag: high-throughput sequence tagging via an empirically derived fragmentation model. Analytical Chemistry **75** (2003) 6415-6421.

13. Han, Y., Ma, B., Zhang, K.: SPIDER: Software for Protein Identification from Sequence Tags with De Novo Sequencing Error. IEEE Computational Systems Bioinformatics Conference (CSB 2004) (2004)

14. Tanner, S., Shu, H., Frank, A., Mumby, M., Pevzner, P., Bafna, V.: InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. Analytical Chemistry **77** (2005) 4626--4639.

15. Frank, A., Tanner, S., Pevzner, P.: Peptide Sequence Tags for Fast Database Search in Mass Spectrometry. Ninth International Conference on Computational Molecular Biology (RECOMB 2005) (2005)

16. Yates, J.r., Eng, J.K., McCormack, A.L., Schieltz, D.: Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. Analytical Chemistry **67** (1995) 1426-1436.

17. Eng, J.K., McCormack, A.L., John R. Yates, I.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. Journal of the American Society for Mass Spectrometry **5** (1994) 976-989.

18. Dancik, V., Addona, T., Clauser, K., Vath, J., Pevzner, P.: De novo protein sequencing via tandem mass-spectrometry. Journal of Computational Biology **6** (1999) 327-341.

19. Frank, A., Pevzner, P.: PepNovo: De Novo Peptide Sequencing via Probabilistic Network Modeling. Analytical Chemistry **77** (2005) 964 -973.

20. Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., Lajoie, G.: PEAKS: Powerful Software for Peptide De Novo Sequencing by MS/MS. Rapid Communications in Mass Spectrometry **17** (2003) 2337-2342.

21. Frank, A., Tanner, S., Pevzner, P.: Peptide Sequence Tags for Fast Database Search in Mass Spectrometry. International Conference on Research in Computational Molecular Biology (RECOMB) (2005)

22. Han, Y., Ma, B., Zhang, K.: SPIDER: Software for Protein Identification from Sequence Tags with De Novo Sequencing Error. 2004 IEEE Computational Systems Bioinformatics Conference (CSB'04) (2004)

23. Taylor, J.A., Johnson, R.S.: Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. Analytical Chemistry **73** (2001) 2594-2604.

24. Ning, K., Leong, H.W.: Towards a Better Solution to the Shortest Common Supersequence Problem: A Post Processing Approach. SCBB06 of IMSCCS06 (2006)

25. Ning, K., Leong, H.W.: Towards a Better Solution to the Shortest Common Supersequence Problem: The Deposition and Reduction Algorithm. BMC Bioinformatics **7** (2006) S12.

26. Craig, R., Cortens, J.P., Beavis, R.C.: Open source system for analyzing, validating, and storing protein identification data. J Proteome Res. **3** (2004) 1234-1242.

27. Keller, A., Purvine, S., Nesvizhskii, A.I., Stolyar, S., Goodlett, D.R., Kolker, E.: Experimental protein mixture for validating tandem mass spectral analysis. OMICS **6** (2002) 207-212.

28. Taylor, J.A., Johnson, R.S.: Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. Rapid Communications in Mass Spectrometry **11** (1997) 1067-1075.

29. Wu, S., Manber, U.: AGREP - A Fast Approximate Pattern-matching Tool. Proceedings of the Winter 1992 USENIX Conference (1992) 153-162.

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

31. Craig, R., Beavis, R.C.: TANDEM: matching proteins with mass spectra. Bioinformatics **20** (2004) 1466-1467.

32. Pevzner, P.A., Dancik, V., Tang, C.L.: Mutation-tolerant protein identification by mass-spectrometry. Fourth International Conference on Computational Molecular Biology (RECOMB 2000) (2000) 231–236.

33. Tsur, D., Tanner, S., Zandi, E., Bafna, V., Pevzner, P.A.: Identification of Post-translational Modifications via Blind Search of Mass-Spectra. IEEE Computational Systems Bioinformatics Conference (CSB 2005) (2005)

34. Keller, A., Eng, J., Zhang, N., Li, X.-j., Aebersold, R.: A uniform proteomics MS/MS analysis platform utilizing open XML file formats. Molecular Systems Biology (2005) doi:10.1038/msb4100024.

35. Ma, B., Zhang, K., Liang, C.: An Effective Algorithm for the Peptide De Novo Sequencing from MS/MS Spectrum. Journal of Computer and System Sciences **70** (2005) 418-430.

36. Grossmann, J., Roos, F.F., Cieliebak, M., Lipták, Z., Mathis, L.K., Müller, M., Gruissem, W., Baginsky, S.: AUDENS: A Tool for Automated Peptide de Novo Sequencing. J. Proteome Res. **4** (2005) 1768 -1774.

37. Chen, T., Kao, M.-Y., Tepel, M., Rush, J., Church, G.M.: A Dynamic Programming Approach to De Novo Peptide Sequencing via Tandem Mass Spectrometry. Journal of Computational Biology **8** (2001) 325-337.

38. Lu, B., Chen, T.: Algorithms for de novo peptide sequencing via tandem mass spectrometry. Drug Discovery Today: BioSilico **2** (2004) 85-90.

39. Lu, B., Chen, T.: A suboptimal algorithm for de novo peptide sequencing via tandem mass spectrometry. J Comput Biol. **10** (2003) 1-12.

40. Abe, T., Sugawara, H., Kanaya, S., Kinouchi, M., Ikemura, T.: Self-Organizing Map (SOM) unveils and visualizes hidden sequence characteristics of a wide range of eukaryote genomes. Gene **365** (2006) 27-34.

41. Bertone, P., Gerstein, M.: Integrative data mining: the new direction in bioinformatics. IEEE Eng Med Biol Mag **20** (2001) 33-40.

42. Mahony, S., McInerney, J.O., Smith, T.J., Golden, A.: Gene prediction using the Self-Organizing Map: automatic generation of multiple gene models. BMC Bioinformatics (2004) 5-23.

43. Kohonen, T.: Self-Organizing Maps. Springer (2001).

44. Leutenegger, S.T., Lopez, M.A., Edgington, J.M.: STR: A Simple and Efficient Algorithm for R-Tree Packing. Proceedings of the 1997 International Conference on Data Engineering (ICDE) (1997)

45. Ramakrishnan, S.R., Mao, R., Nakorchevskiy, A.A., Prince, J.T., Willard, W.S., Xu, W., Marcotte, E.M., Miranker, D.P.: A fast coarse filtering method for peptide identification by mass spectrometry. Bioinformatics **22** (2006) 1524-1531.

46. Ng, H.K., Leong, H.W.: Path-Based Range Query Processing Using Sorted Path and Rectangle Intersection Approach. DASFAA 2004 (2004) 184-189.

47. Ng, H.K., Leong, H.W., Ho, N.L.: Efficient Algorithm for Path-Based Range Query in Spatial Databases. IDEAS 2004 (2004) 334-343.

48. Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J.: SOM_PAK: The Self-Organizing Map Program Package. Technical Report A31 (1996) FIN-02150 Espoo.

49. Prince, J.T., Carlson, M.W., Wang, R., Lu, P., Marcotte, E.M.: The need for a public proteomics repository. Nat Biotechnol. **22** (2004) 471-472.

50. Desiere, F., Deutsch, E.W., King, N.L., Nesvizhskii, A.I., Mallick, P., Eng, J., Chen, S., Eddes, J., Loevenich, S.N., Aebersold, R.: The PeptideAtlas Project. Nucleic Acids Research **34** (2006) D655-D658.

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

52. Liu, C., Yan, B., Song, Y., Xu, Y., Cai, L.: Peptide sequence tag-based blind identification of post-translational modifications with point process model. Bioinformatics **22** (2006) e307-e313.

53. Ning, K., Chong, K.F., Leong, H.W.: De novo Peptide Sequencing for Multi-charge Mass Spectra based on Strong Tags. RECOMB 2006 (2006) Poster.

54. Arnold, R.J., Jayasankar, N., Aggarwal, D., Tang, H., Radivojac, P.: A Machine Learning Approach to Predicting Peptide Fragmentation Spectra. Pacific Symposium on Biocomputing **11** (2006) 219-230.

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

56. Qian, N., Sejnowski, T.J.: Predicting the secondary structure of globular proteins using neural network models. J Mol Biol. **202** (1988) 865-884.

57. Searle, B.C., Dasari, S., Wilmarth, P.A., Turner, M., Reddy, A.P., David, L.L., Nagalla, S.R.: Identification of protein modifications using MS/MS de novo sequencing and the Opensea alignment algorithm. J. Proteome Res. **4** (2005) 546-554.

58. Searle, B.C., Dasari, S., Turner, M., Reddy, A.P., Choi, D., Wilmarth, P.A., McCormack, A.L., David, L.L., Nagalla, S.R.: High-Throughput Identification of Proteins and Unanticipated Sequence Modifications Using a Mass-Based Alignment Algorithm for MS/MS de Novo Sequencing Results. Anal. Chem. **76** (2004) 2220 -2230.

59. Jiang, T., Li, M.: On the approximation of shortest common supersequences and longest common subsequences. SIAM Journal of Computing **24** (1995) 1122-1139.

60. Chvatal, V., Sankoff, D.: Longest common subsequences of two random sequences. Journal of Applied Probability (1975) 306-315.

61. Ning, K., Choi, K.P.: Analysis of the expected length of Longest Common Subsequence. in preparation (2007)

62. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. The MIT Press (2001).

63. Hunt, J.W., McIlroy, M.D.: An algorithm for differential file comparison. Bell Telephone Laboratories CSTR #41 (1976)

64. Paterson, M., Dancik, V.: Longest common subsequences. Mathematical Foundations of Computer Science, 19th International Symposium (MFCS), volume 841 of LNCS (1994) 127-142.

65. Masek, W., Paterson, M.: A faster algorithm computing string edit distances. Journal of Computer and System Sciences **20** (1980) 18-31.

66. Hakata, K., Imai, H.: The Longest Common Subsequence Problem for Small Alphabet Size between many Strings. Proc. of 3rd International Symposium on Algorithms and Computation (ISAAC), Volume 6 of LNCS. Springer Verlag (1992) 469-478.

67. Hsu, W., Du, M.: New Algorithms for the LCS Problem. Journal of Computer and System Sciences **19** (1984) 133-152.

68. Bonizzoni, P., Vedova, G.D., Mauri, G.: Experimenting an approximation algorithm for the LCS. Discrete Applied Mathematics **110** (2001) 13 - 24.

69. Ning, K., Zhang, L.: A Post Processing Approach for the Longest Common Subsequence Problem. in preparation (2007)

70. Storer, J.A.: Data compression: methods and theory. Computer Science Press (1988).

71. Foulser, D.E., Li, M., Yang, Q.: Theory and algorithms for plan merging. Artificial Intelligence **57** (1992) 143 - 181.

72. Sellis, T.K.: Multiple-query optimization. ACM Transactions on Database Systems (TODS) **13** (1988) 23 - 52.

73. Sankoff, D., Kruskal, J.: Time Warps, String Edits and Macromolecules: the Theory and Practice of Sequence Comparisons. Addison Wesley (1983).

74. Kasif, S., Weng, Z., Derti, A., Beigel, R., DeLisi, C.: A computational framework for optimal masking in the synthesis of oligonucleotide microarrays. Nucleic Acids Research **30** (2002) e106.

75. Ning, K., Choi, K.P., Leong, H.W., Zhang, L.: A Post Processing Method for Optimizing Synthesis Strategy for Oligonucleotide Microarrays. Nucleic Acids Research **33** (2005) e144.

76. Barone, P., Bonizzoni, P., Vedova, G.D., Mauri, G.: An approximation algorithm for the shortest common supersequence problem: an experimental analysis. Symposium on Applied Computing, Proceedings of the 2001 ACM symposium on Applied computing (2001) 56 - 60.

77. Timkovsky, V.G.: On the approximation of shortest common non-subsequences and supersequences. Technical report (1993)

78. Ning, K., Leong, H.W.: The distribution and deposition algorithm for multiple oligo nucleotide arrays. The 17th International Conference on Genome Informatics (2006)

79. Ning, K., Leong, H.W.: The Distribution and Deposition Algorithm for Multiple Sequences Set. In preparation (2007)

80. Ning, K., Ng, H.K., Leong, H.W.: Finding Patterns in Biological Sequences by Longest Common Subsequences and Shortest Common Supersequences. Sixth IEEE International Symposium on BioInformatics and BioEngineering (BIBE 2006) (2006)

Appendix A: Multiple Sequences Analysis

Multiple sequences analysis is important in many applications, especially in bioinformatics. In multiple sequences comparison, the computation of the Longest Common Supersequence (LCS) and the Shortest Common Subsequence (SCS) are well-known NP-hard problems [59], and these are my focus in multiple sequence analysis. I have also investigated the SCS problem on multiple sets of sequences. And I have also applied the algorithms for SCS problem on the problem of synthesis strategy design for oligos arrays, and on the problem of pattern discovery in biological sequences.

An overview of my work in multiple sequences analysis is illustrated in Figure 19.

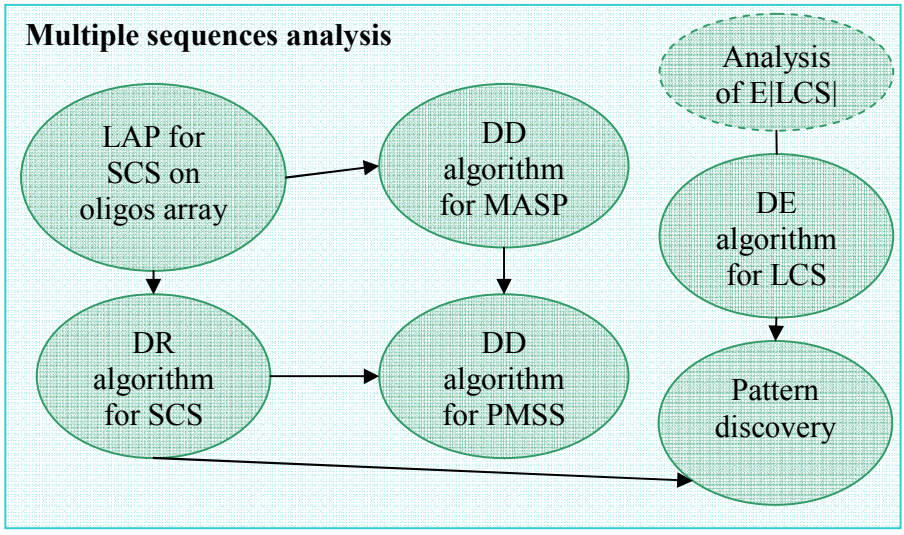


Figure 19. The outline of my research in multiple sequences analysis.

A.1 Longest Common Subsequence

The LCS of a set of sequences can be formulated as this. For two sequences S=$s_1 \ldots s_m$ and $T=t_1 \ldots t_n$, S is the subsequence of T (T is the supersequence of S) if for some $1 \le i_1 < \ldots < i_m \le n$, $s_j = t_{i_j}$. Given a finite set of sequences $S=\{S_1, S_2, \ldots, S_k\}$, a common

subsequence (CS) of S is the sequence T such that each sequence in S is a supersequence of T, and a LCS of S is the longest possible T among all of CS for this set of sequences S. In the following parts, we will define n as the length of each sequence, and k as the number of sequences in the sequences set.

In these series of projects, I have concentrated on two aspect of LCS. I have first analyzed the expected length of LCS for two random binary sequences with arbitrary length, and then extend the analysis to multiple sequences with multiple alphabets. I have also proposed a novel heuristic algorithm for LCS problem on multiple sequences.

In the theoretical aspect, Let L_n be the length of the LCS of two random binary sequences (S and T) of length n. It is proven by subadditivity property of the LCS that there exists a γ, so that expected value of L_n, $E[L_n] \sim \gamma n$ [60]. However, the exact value of γ, as well as its variances, is unknown.

In [61], we tries to empirically analyze the expected length of LCS (E(|LCS|)) for a pair or a set of sequences, with special concern on two random binary sequences. We have performed extensive simulation on E(|LCS|) of two random binary sequences, and then extended the work on sequences with more alphabets, and multiple sequences.

In the practical aspect, the problem of finding the LCS (we will refer to this simply as LCS problem in the following part) is important and has many applications in different areas of computer science. The LCS problem has applications in many areas, including data compression, pattern recognition, file comparison and biological sequence

comparisons and analysis [1, 62]; and there are some applications been commonly used based on the computation of LCS of two sequences, like UNIX diff command [63].

The LCS problem has been examined extensively by many researchers (refer to [64]). The LCS of two sequences can be computed by dynamic programming in $O(n^2)$ time and $O(n^2)$ space, and there are many researches on this problem using dynamic programming with reduced time and space [62, 65].

Unfortunately, the LCS problem on arbitrary k sequences is a well-know NP-hard problem that is even hard to approximate in the worst case [59]. Though there are efficient dynamic programming algorithms on computation of LCS for small k [66, 67], these algorithms are not suitable for dataset in which there are many long sequences [1].

Though LCS problem is NP-hard, it is so important in application that many heuristic algorithms have been proposed to solve the LCS problem [59, 64, 68]. These algorithms compute the common subsequences (not necessarily the longest) of the input sequences. However, current heuristic algorithms for the LCS problem are not suitable for both small and large LCS instances. By *large LCS instances*, we mean instances where (a) the sequences in S are *long* (n is 100 and more), (b) there are *many* sequences (k is 100 or more), and (c) large sizes of alphabets sets (q can be up to 50). And other instances are *small LCS instances*.

In [69], I have proposed heuristic algorithm for the problem of finding LCS of a set of sequences, with emphasis on large LCS instances. I have proposed a new heuristic algorithm for the LCS problem, the Deposition and Extension algorithm (DEA). This

algorithm is based on the generation of common subsequence by deposition process, and then extends this common subsequence. The algorithm is proven to generate result with length equal to or longer than those generated by Long Run algorithm. The experiments show that our algorithm performs comparable to or better than Long Run and Expansion Algorithm, especially on many long sequences. The algorithm also has superior efficiency.

A.2 Shortest Common Supersequence

The problem of finding the *Shortest Common Supersequence* (SCS) of a given set of sequences is a very important problem in computer science, especially in computational biology. The SCS of a set of sequences can be stated as follows: Given two sequences $S = s_1 s_2 ... s_m$ and $T = t_1\ t_2 ... t_n$, over an alphabet set $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_q\}$, we say that S is the *subsequence* of T (and equivalently, T is the *supersequence* of S) if for some $1 \le i_1 < ... < i_m \le n$, $s_j = t_{i_j}$. Given a finite set of sequences $S = \{S_1, S_2, \ldots, S_k\}$, a *common supersequence* of S is a sequence T such that T is a supersequence of *every* sequence S_j ($1 \le j \le k$) in S. Then, the *shortest common supersequence* (*SCS*) *of* S is a common supersequence of S that has *minimum* length. In the following part, we assume that k is the number of sequences in S, n is the length of each sequence, and $q = |\Sigma|$ is the size of the alphabet.

The SCS problem has found diverse applications in many areas, including data compression [70], scheduling [71], query optimization [72], file comparison and biological sequence comparison and analysis [62, 73].

The SCS problem has been investigated extensively by many researchers [1]. The SCS of two sequences can be computed using dynamic programming in $O(n^2)$ time and $O(n^2)$ space, and there are many researches on improving the running time and space required for the algorithms [1]. For a fixed small k, the dynamic programming algorithm can be extended to solve the SCS problem in $O(n^k)$ time and space. Unfortunately, the SCS problem on arbitrary k sequences is well-known to be NP-hard [59].

Our interest in SCS problem is intrigued by our analysis of the SCS on DNA sequences in the context of synthesis strategy design for oligos array. Several heuristic algorithms were proposed specifically for computing the SCS of DNA sequences (with alphabet size of 4) in the context of oligos array. These include Min-Height [74], Sum-Height [74] heuristics. Recently, we proposed *look-ahead* extensions of these heuristic algorithms on DNA sequences [75], as well as a post-processing reduction procedure and studied the performances of these algorithms on DNA sequences to be used for the synthesis of oligos array. For the more general SCS problem, a trivial algorithm, called Alphabet [76] gives an approximation ratio of $q = |\Sigma|$. In practice, there are many heuristic algorithms that produce results better than the Alphabet algorithm, including Majority Merge [59] (interestingly, the Majority Merge [59] and Sum-Height [74] heuristic are the same algorithm), Tournament [77], Greedy [77] and Reduce-Expand [76].

This series of projects focus on heuristic algorithms for solving SCS problem on *large SCS instances*. By *large SCS instances*, we mean SCS instances S in which

- the sequences in S are *long* (n is 100 to 1000),
- there are *many* sequences (k is 100 or more), and

- the alphabet set may be *big* (q is 20 for protein sequences, and even larger for text sequences).

Large SCS instances arise more frequently in the post-genome era in biological applications dealing with DNA and protein sequences, as well as current applications on large text sequences dataset.

In this series of projects on SCS problems, the heuristic algorithms for SCS problem are primarily motivated by our analysis on the synthesis strategy for oligos array. The broad applicability of gene expression profiling to genomic analyses has generated huge demand for mass production of microarrays and hence for improving the cost effectiveness of microarray fabrication. The first project is to analyze the algorithms for generating synthesis strategies for DNA oligos array [75], in which there are many short sequences. We have proposed a post-processing heuristic algorithm for deriving a good synthesis strategy. We assessed all the known efficient algorithms and our post-processing algorithm for reducing the number of synthesis cycles for manufacturing an oligos array of a given set of oligos. Our experimental results on both simulated and real datasets show that no single algorithm consistently gives the best synthesis strategy; and post-processing extension to existing strategy is necessary as it often reduces the number of synthesis cycles further.

Based on the heuristic algorithms for the synthesis strategy design for oligos array, we have investigated SCS problem on large SCS instances. In [24, 25], We have extended the LAP algorithm for oligos synthesis to the problem of finding the SCS of a set of sequences. We have proposed a post process heuristic algorithm for the SCS problem, the

Deposition and Reduction algorithm. The algorithm is proven to generate SCS with length equal to or shorter than $|\sum|$ times of the optimal length. The experiments show that our algorithm can perform comparable to or better than many of the best known algorithms, and outperform them a lot on large SCS instances.

A.3 Multiple Sequences Set

I have extended the problem of synthesis of oligos array to the problem of **synthesis of multiple oligos arrays**. and proposed greedy algorithms as well as the **DDA heuristic algorithm** for it [78].

Based on the analysis of the synthesis of multiple micorarray problem, I have also extended the SCS problem to **Process of Multiple Sequences Set (PMSS) problem**. I have formulated the problem mathematically, proved that it is NP-hard, and extended the DDA algorithm for this problem [79].

The Process of Multiple Sequences Set (PMSS) problem is a computational model that has many applications. Our modeling of the problem and research in this area is a pioneer work for emerging large scale combinatorial problems. The algorithms that I have proposed are potentially important for the problem.

A.4 Pattern Identification Based on LCS and SCS

Based on the analysis of the relationships between LCS, SCS and patterns, we have designed a heuristic algorithm **(PALS)** that can **find patterns in multiple sequences** from their LCS and SCS [80].

Patterns in biological sequences are important for revealing the relationships among biological sequences. Much research has been done on this problem. It is interesting that patterns, Longest Common Subsequences (LCS) and Shortest Common Supersequences (SCS) represent different aspects of a profile for a set of sequences. However, in general, for problems on a set of sequences, the relationship between their patterns and their LCS and SCS are not examined carefully. Therefore, revealing the relationship between the patterns and LCS/SCS might provide us with better algorithms for patterns discovery of biological sequences, in turn leading to better understanding of their relationship. We propose the PALS (PAtterns by Lcs and Scs) algorithms to discover patterns in a set of biological sequences by first generating the results for LCS and SCS by heuristic, and consequently deriving the patterns from these results. Experiments show that the PALS algorithms perform well (both in efficiency and accuracy) on a variety of sequences datasets.

A.5 Conclusions

This chapter of my research encompasses the analysis of algorithms on LCS and SCS problems for multiple sequences; proposing algorithms that extends the SCS problem to multiple sequences sets, as well as the application of SCS problems on synthesis strategy design for oligos arrays.

The analysis of the LCS and SCS problems has led to a deeper view of the problems, and better solutions to these problems, especially on increasing amount of large sequences dataset that contains many long sequences. Specifically, the analysis of SCS problem on

synthesis strategy design for oligos arrays can lead to more efficient synthesis applications.

Also, the mathematical formulations and extensions of these bioinformatics problems, such as the multiple sequences sets problems, are interesting combinatorial problems in computer science that have vast applications, which are worth further investigation.

For the analysis of the expected length of LCS, although we have used Monte-Carlo simulation method for empirical analysis, there is still a lack of theoretical results for this problem. I think further investigation of current upper and lower bound and optimization of their calculation may be beneficial for achieving further improvement for this problem.

As for heuristic algorithms for LCS and SCS problems, we already proved that the deposition strategy is superior to other approaches, especially on many long sequences. It is interesting to know whether there are optimization methods for deposition strategy to further improve the quality of the results without sacrificing efficiency significantly.

The Process of Multiple Sequences Sets (PMSS) problem is a novel mathematical formulation for a broad set of application, with a special case in the Multiple Array Synthesis Problem (MASP). I think deeper investigation of this problem will be beneficial for research in many applications.

For the PALS algorithm designed for pattern discovery in biological sequences, though it performs well on a large number of biological sequences, it does not output all of the possible patterns. And I think further investigation of the relationship among pattern,

LCS and SCS may lead to more accurate and complete pattern discovery results in biological sequences.