Statistical Methods for Peptide and Protein Identification using Mass Spectrometry

Qunhua Li

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

2008

Program Authorized to Offer Degree:
Department of Statistics

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Qunhua Li

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

_____
Matthew Stephens

Reading Committee:

_____
Matthew Stephens

_____
Michael MacCoss

_____
Adrian Raftery

Date: _____

University of Washington

Abstract

Statistical Methods for Protein Identification using Mass Spectrometry

Qunhua Li

Chair of the Supervisory Committee:
Professor Matthew Stephens
Dept of Statistics

Protein identification using mass spectrometry is a high-throughput way to identify proteins in biological samples. The identification procedure consists of two steps. It first identifies the peptides from mass spectra, then determines if proteins assembled from the putative peptide identifications are present in the samples. In this dissertation, we present statistical methods for these two steps.

The main goal of the first step is to select the peptide sequence that is most likely to generate the observed spectrum from candidate sequences in a protein database, according to the similarity between the observed spectrum and the theoretical spectra predicted from candidate sequences. For this part, we developed a likelihood-based scoring algorithm based on a generative model, which measures the likelihood that the observed spectrum arises from the theoretical spectrum of each candidate sequence. Our probabilistic model takes account of multiple sources of noise in the data, e.g. variable peak intensities and errors in peak locations. We also provided two measures for assessing the uncertainty of each identification. We evaluated the performance of our method on a benchmark dataset, and compared with a widely-used peptide identification algorithm.

The main goal of the second step is to assess the evidence for presence of proteins and constituent peptides identified from mass spectra. We developed a model-based clustering approach to this problem, based on a nested mixture model. In contrast to commonly-

used two-stage approaches, our model provides a one-stage solution that simultaneously identifies which proteins are present, and which peptides are correctly identified. In this way, our model incorporates the evidence feedback between proteins and their constituent peptides. Using simulated data and a yeast dataset, we compare and contrast our method with existing widely-used approaches. For peptide identification, our single-stage approach yields consistently more accurate results. For protein identification, the methods have similar accuracy in most settings, although we exhibit some scenarios in which the existing methods perform poorly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my adviser, Matthew Stephens. I am greatly indebted to him for agreeing to be my adviser and taking on a field that I had been working on but was somewhat new to him, at the time when he was about to move to another institute, for his conscientious efforts to make long-distance advising work, for listening to me and giving me freedom to explore, and for believing in me all the time. I am very grateful for his constructive feedback and insightful advice to my research and career development, enormous efforts to help me improve my writing, and constant support, encouragement and inspiration.

I am grateful to Mike MacCoss, with whom I have been collaborating on the project of protein identification, for his constant support, insights to proteomics, and many valuable suggestions on this work. I would like to thank Murray Hackett for teaching me fundamentals in proteomics, and thank Marina Meila, with whom I started working on proteomics. Some work with Marina and Murray motivated part of the research (chapter 2) in this dissertation. My thanks also go to Jimmy Eng and Lukas Kall for helpful discussions and for providing some of the data used in the dissertation.

I would like to thank Adrian Raftery for his feedback on part of this work (chapter 3), which motivated the design of simulation studies, and also for his advice on research in general. I also thank Werner Stuetzle and Thomas Richardson for being on my committee and for their support.

I would like to thank Martin McIntosh in the Fred Hutchinson Cancer Research Center for providing financial support for almost two years when I was working on this dissertation, and for providing me an exposure to proteomics research outside of my dissertation.

I also want to thank those, especially the research groups of Adrian Raftery, Mike MacCoss and Bill Noble, who gave me opportunities to talk in their group meetings and gave me feedback on this work. Their comments greatly help me improve my work and presentation.

Finally, I would like to thank my family for their support, encouragement and understanding in many years. I especially thank my daughter, Claire, who has been my greatest joy and unconditional support, for letting me grow with her.

Chapter 1

INTRODUCTION

Proteins are essential parts of organisms and participate in every process within cells. The major goal of proteomics research is to identify and characterize the proteins in cells grown under a variety of conditions (Aebersold and Mann, 2003). Over the past few years, tandem mass spectrometry (MS/MS) has become the most widely used tool for identifying proteins in complex biological samples (Steen and Mann, 2004).

This dissertation develops statistical methods for protein identification using MS/MS spectra. In this chapter, we review some basic concepts for understanding the technology of tandem mass spectrometry and different steps involved in protein identification using mass spectra.

The organization of this chapter is as follows. Section 1.1 introduces some basic concepts of protein chemistry. Section 1.2 briefly introduces the experimental procedure of mass spectrometry and the steps towards protein identification. Section 1.3 reviews the fragmentation process, which is fundamental for understanding how to interpret and analyze mass spectra. Section 1.4 reviews the methods for predicting fragmentation pattern from peptide sequences. Section 1.5 provides a general review of the peptide identification algorithms. Section 1.6 reviews the approaches to identifying proteins from peptide identifications. Section 1.7 outlines the statistical questions involved in protein identification using mass spectrometry and the contribution of the dissertation.

Table 1.1: Monoisotopic residue masses of 20 basic amino acids. Note that I and L have identical mass.

amino acid	mass	amino acid	mass	amino acid	mass	amino acid	mass
A	71.04	C	103.01	D	115.03	E	129.04
F	147.07	G	57.02	H	137.06	I	113.08
K	128.09	L	113.08	M	131.04	N	114.04
P	97.05	Q	128.06	R	156.10	S	87.03
T	101.05	V	99.07	W	186.08	Y	163.06

1.1 Basic concepts in protein chemistry

Amino acids are building blocks of proteins. There are 20 basic amino acids. They have the same basic structure and are distinguished from each other by their side chain residues. Most of the amino acids have distinguishable masses (Table 1.1), which makes peptide identification by MS/MS spectra possible.

Proteins are polymers formed from the linking, in a defined order, of amino acids. The link between two amino acids in a protein (or peptide) sequence is known as an amide bond or a peptide bond. Peptides are short polymers of amino acids. They are often used to refer to substrings of proteins. Proteins and peptides are defined by their unique sequences of amino acid residues.

1.2 Tandem mass spectrometry and protein identification

Tandem mass spectrometry has become the method of choice for high-throughput protein identification, because it enables to identify a large number of proteins with high sensitivity, fast speed and simple sample preparation.

In a typical MS/MS experiment (Figure 1.1), a mixture of proteins is first digested with an enzyme, which breaks proteins into shorter peptides. Next, the resulting peptide mixture is separated by liquid chromatography and transformed into electrically charged particles

before being analyzed by a mass spectrometer. In the mass spectrometer, the charged peptides (called precursor ions) are first separated by their mass-to-charge ratios (m/z). They are then fragmented individually, and the m/z values (x-axis) and the abundance (y-axis) of the product ions from the fragmentation are measured. These data form the MS/MS spectrum that is characteristic of each peptide. In a typical experiment of this type, thousands of MS/MS spectra are generated. An example observed spectrum of a peptide containing 7 amino acid is shown in Figure 1.3c.

After spectra have been generated, the peptide that is most likely to generate each observed MS/MS spectrum is identified using a computational method. The identified peptides are used to infer the identity of proteins in the original biological sample. These two statistical problems are the focus of this thesis. The overall experimental procedure and flow of data are shown in Figure 1.2.

1.3 Fragmentation process

This section briefly introduces the chemical process to generate MS/MS spectra in mass spectrometer, which is essential to the understanding of the following fundamental questions:

- Why can tandem mass spectrometry be used for sequencing peptides?
- What are the sources of noise on mass spectra?
- What information is available for computational strategies to identify peptide sequences from spectra?

In tandem mass spectrometry, a mass spectrum is obtained by fragmenting a peptide using a process called low-energy collision-induced dissociation (CID) (See Kinter and Sherman (2003) for a review). In the most simplified process, each copy of a peptide is fragmented on only one peptide bond, to form a pair of complimentary prefix and suffix fragments (the most common ones being called b- and y-ions). As the fragmentation process can occur on multiple copies of the peptide at different bond positions, a series of consecutive peptide

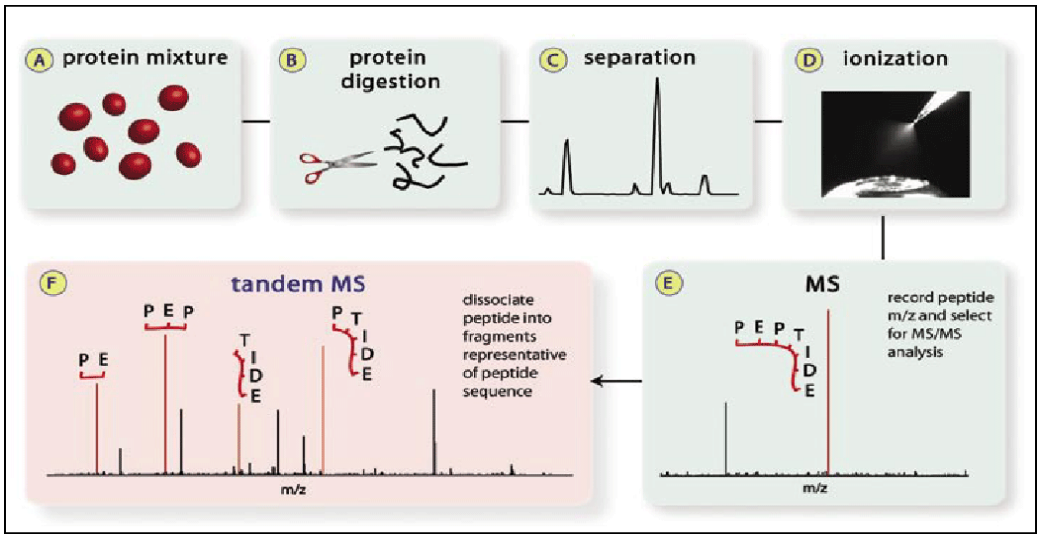


Figure 1.1: A typical tandem mass spectrometry experiment. (A) The process begins with a mixture of proteins. (B) Enzymes are used to digest the proteins into peptides. (C) Digested peptides are separated with liquid chromatography. (D) Peptides are transformed into electrically charged particles (called ionization). (E) The charged peptides are separated based on their mass/charge (m/z) within the mass spectrometer. (F) Peptides are individually selected for fragmentation to obtain sequence information. Following peptide ion dissociation the newly generated fragments are m/z analyzed to produce the MS/MS spectrum. Taken from Coon et al (Coon et al., 2005).

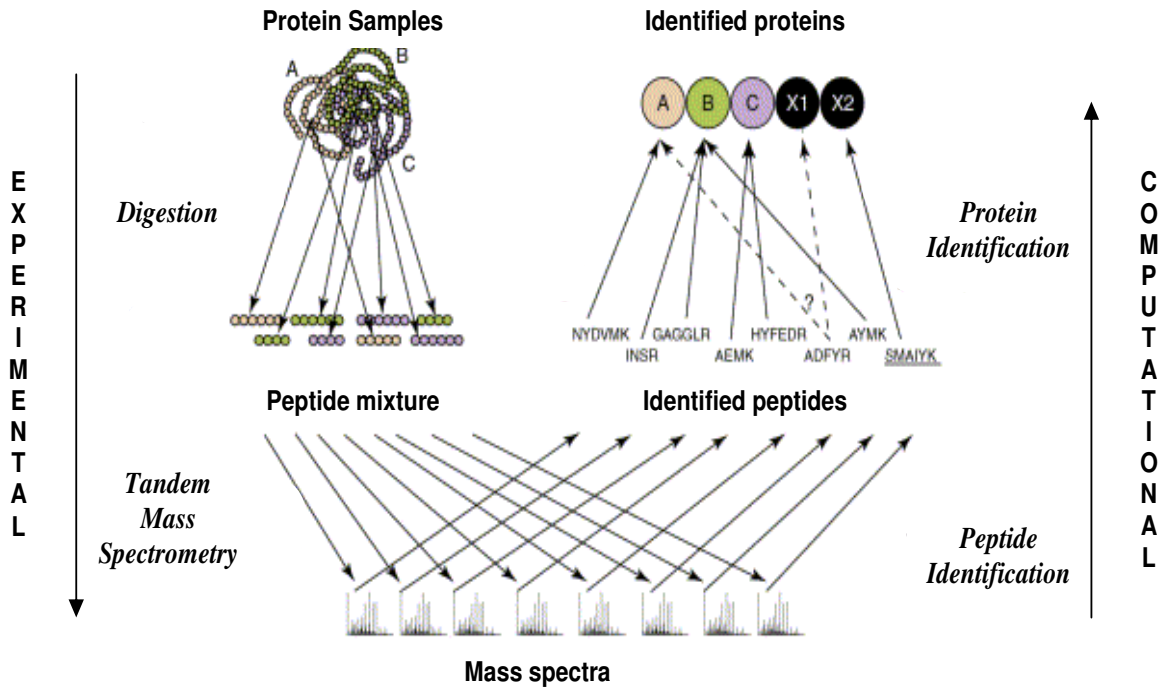


Figure 1.2: Experimental procedures and flow of data in a typical analysis of a complex protein mixture based on high-performance liquid chromatography (LC) coupled with tandem mass spectrometry (MS/MS). First, sample proteins are digested into short peptides. After separation using LC, peptides are ionized, and selected ions are fragmented to produce MS/MS spectra. Computational tools are used to assign a peptide to each acquired MS/MS spectrum. The next step is to determine which proteins are present in the original sample (A, B and C) and which are false identifications (X1 and X2) corresponding to incorrect peptide assignments (underlined peptide sequence). Modified from Nesvizhskii and Aebersold (2004).

fragments (called a "ladder"), which differ in mass by a single amino acid, can be observed. Figure 1.3a illustrates the ladder of b- and y-ions, using a peptide of seven amino acids as an example.

As most amino acids have distinguishable masses, each peptide sequence at a given charge state has its signature ladder of masses, which, in conjunction with the charge of each fragment, designates the locations of peaks on the spectrum. (Note that amino acids I and L can not be distinguished by mass spectrometry, because they have identical mass.) If a complete ladder is observed on a spectrum and the charge of each fragment is known, one can read the amino acid sequence of the peptide.

However, the fragmentation process in practice is much more complicated than what is described above. Though the products from the dissociation on the peptide bonds usually are the dominant fragments, the experimental spectra contains a large number of peaks from the product ions of minor fragmentation pathways or loss of small neutral compounds (e.g. water), and unknown noise. The fragmentation pattern of a peptide also depends on its charge state (i.e. the number of protons obtained at ionization), which is determined by its amino acid composition. In general, peptides with higher charge states have more complicated fragmentation pattern than the ones with lower charge states, because they can undergo more fragmentation pathways. In addition, the abundance of fragments in CID strongly depends on the physiochemical properties of the amino acids and sequences of the peptides. As a result, the peaks on the experimental spectra usually have highly variable intensities.

Though observed spectra have complicated fragmentation patterns, naive predictions that are based mainly on the b-y ions are commonly used for measuring the similarity to the observed spectra. To see the deviation between the naive prediction and experimental spectra, Figures 1.3b-c show the naive prediction corresponding to the ladder and an experimental spectrum for the example above.

The information of fragmentation process can be used for identifying peptide sequences from MS/MS spectra in two ways. One is to find the peaks on the ladder and read the amino acid sequence from the ladder. The other is to generate expected fragmentation pattern for a list of peptide candidates according to the fragmentation rules, and identify the peptide whose expected pattern is most similar to the observed spectrum. These two approaches form the two main paradigms of peptide identification algorithms, which is described in detail in Section 1.5.

1.4 Prediction of fragmentation patterns

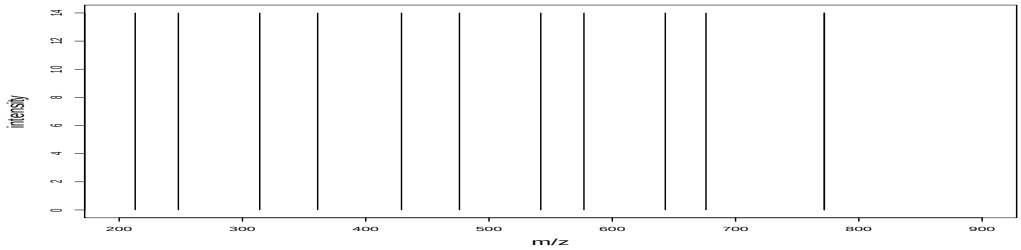
Understanding the fragmentation patterns of peptides in CID is important for identifying peptides from MS/MS spectra. To quantify the factors influencing peptide fragmentation pattern, many approaches have been attempted, including both computational approaches and models based on chemical principles. Examples of computational methods include Elias et al. (2004), who used a decision tree approach, Klammer et al. (2008), who used a Bayesian network approach, and Schutz et al. (2003), who used a regression model to identify the effects of amino acids composition. Though they provide insights into the factors influencing peptide fragmentation patterns, none of these methods can predict the fragmentation pattern for a given peptide sequence.

By far the most extensive program that can predict the intensities on the theoretical spectra is a kinetic model developed in Zhang (2004, 2005). This model is based on a mobile proton model Wysocki et al. (2000), which is a chemical model to explain intensity patterns observed in MS/MS. This prediction model builds on a comprehensive set of reaction pathways for modeling the fragmentation process in a mass spectrometer, and considers parameters including amino acid composition and experimental conditions. In addition, the parameters were trained on a large number of mass spectra. As a result, it predicts fragmentation patterns that are much more similar to the experimental spectra than the naive prediction mentioned in section 1.3. In particular, it provides a more complete set of peaks and predicts the theoretical intensities according to the amino acid composition

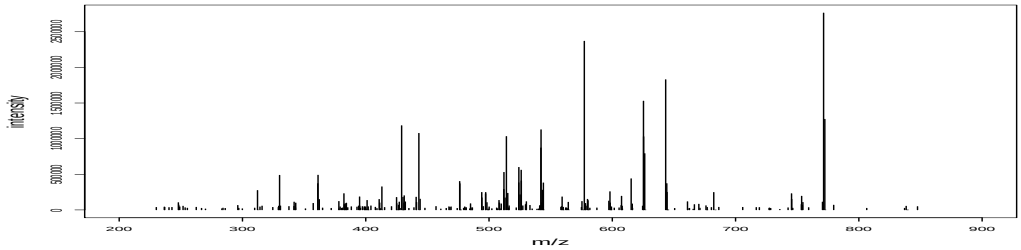
Peptide: L-V-T-D-L-T-K

mass	ion		ion	mass
114	b1	L V-T-D-L-T-K	y6	676
213	b2	L-V T-D-L-T-K	y5	577
314	b3	L-V-T D-L-T-K	y4	476
429	b4	L-V-T-D L-T-K	y3	361
542	b5	L-V-T-D-L T-K	y2	248
643	b6	L-V-T-D-L-T K	y1	147

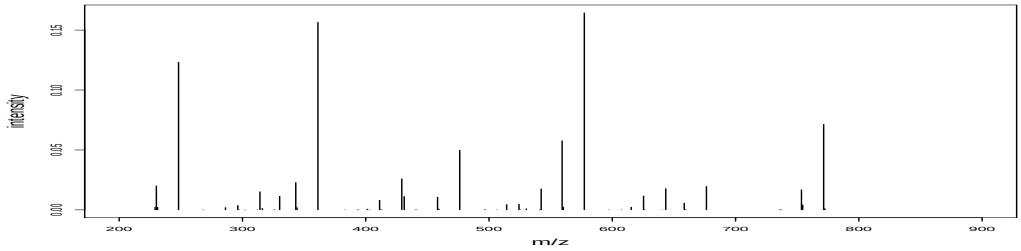
a. The ladder of b- and y-ions



b. The spectrum corresponding to the ladder



c. An observed spectrum



d. The spectrum predicted using Zhang's algorithm

Figure 1.3: An illustration of fragmentation pattern for peptide LVTDLTK. (a) The ladder of b- and y-ions, (b) the spectrum corresponding to the ladder, (c) an observed spectrum of this peptide, and (d) its theoretical spectra predicted based on Zhang's algorithm.

of peptides. Figure 1.3d shows the prediction based on Zhang's model for the example in Section 1.3.

1.5 Peptide identification

1.5.1 Paradigms of peptide identification

As mentioned earlier in Section 1.3, based on how fragmentation patterns are used for peptide identification, the peptide identification algorithms using mass spectrometry can be categorized into two main paradigms: *de novo* sequencing algorithms and database search algorithms. *De novo* sequencing infers peptide sequences directly from spectra by finding mass differences between the peaks that correspond to amino acids. Because this approach requires a spectrum of good quality and only partial sequences may be inferred if some of peaks in the series are missing, its use for high-throughput peptide identification is limited. Database search algorithm is the most commonly used strategy for high-throughput peptide identification (Sadygov et al., 2004). It assumes the protein database is a complete collection of all the possible peptide sequences. Peptides are identified by finding the sequence that matches best to the observed spectrum in the protein database of the species that the sample comes from. There are also hybrid approaches that combine the inference of partial sequences obtained from *de novo* algorithm with database search (Tabb et al., 2002; Tanner et al., 2005). Our focus here is the database search algorithms, as they are most relevant to the research presented in this thesis.

1.5.2 General framework of database search programs

Most database search programs follow the same general framework (Sadygov et al., 2004) (Figure 1.4). First, all peptide candidates that might generate the observed spectrum are extracted from a protein database. Next, the theoretical spectra of the candidate peptides are generated using the fragmentation pathways. Then the resulting theoretical spectra are scored by their similarity to the observed spectrum. The peptide that obtains the best score is selected as the potential correct sequence.

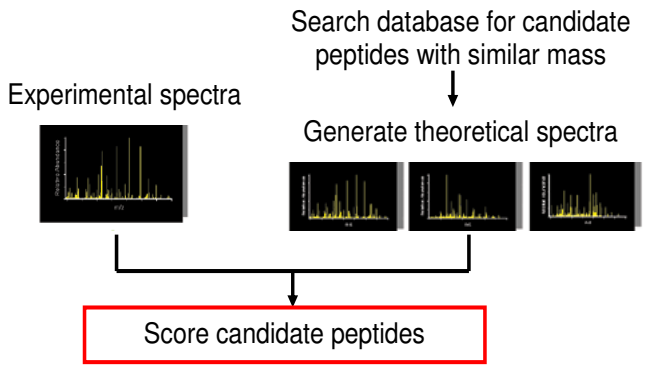


Figure 1.4: General framework for peptide identification using database search. Revised from Sadygov et al. (2004).

1.5.3 Review of database search algorithms

Many different algorithms have been developed for identifying MS/MS data using database searching (See (Sadygov et al., 2004; Hernandez et al., 2006) for recent reviews). The main difference between different algorithms is the scoring function used to quantify the degree of similarity between the compared spectra. Here, I only review the ones that are most relevant to the research presented in this thesis.

SEQUEST (Eng et al., 1994) is one of the most widely used algorithms for database searching. It scores peptide sequences by the cross-correlation between the intensities of peaks on the observed and the theoretical spectra. The cross-correlation is computed as $Xcorr = R_0 - \sum_{t=-75}^{t=75} R_t/151$, where $R_t = \sum_{i=1}^{n} x_i y_{i+t}$, x_i and y_i are the intensities of the peaks at location i (the location of each peak is rounded into the closest integer) on the observed and theoretical spectra, respectively. To increase identification speed, before the computation of the cross-correlation, SEQUEST first calculates a fast preliminary score (called Sp score), and filter out sequences with low Sp scores. It uses a theoretical spectrum containing b- and y-ions and their neutral losses with uniform intensity for the peaks of each ion type. In addition to the cross-correlation score, it also provides several other values to describe peptide-spectrum matches and features of identified sequences.

Mascot (Perkins et al., 1999), another widely used database search algorithm, calculates the probability that the observed matches for a given peptide mass are chance events and reports a p-value. Unfortunately, the algorithm is proprietary and no public details are available.

A natural approach is to use probabilistic models to model the peptide-spectrum match. Several algorithms are motivated by probabilistic ideas and use alignment-based procedures for scoring. SCOPE (Bafna and Edwards, 2001) introduces the idea of modeling the probability that a sequence (p) produces a given spectrum (S) by

$$P(S \mid p) = \sum_{F \in \mathcal{F}} P(S \mid F, p) P(F \mid p)$$

where the sum is taken over all the possible fragmentations of the peptide. Though this idea is initially motivated by a generative model, their actual algorithm does not correspond to the proposed model. For example, although the component $P(F \mid p)$ aims to describe the probability of a particular fragmentation pattern of a peptide, in their implementation, these probabilities are obtained from expert knowledge, which is far from sufficient for modeling the peptide-specific fragmentation patterns. The component of $P(S \mid F, p)$ aims to describe the probability of observing S for a given fragmentation pattern. However, their fitting procedure resembles a sequence alignment and does not specify what the generative model is. The alignment algorithm is later improved and made more explicit in InsPecT (Tanner et al., 2005) with an extension to handle peptides with post-translational modifications. Both approaches use naive fragmentation patterns and do not consider the intensities of observed spectra in alignment.

Similar to SCOPE, PepHMM (Wan et al., 2006) attempts to score peptides by $P(S \mid p)$, but with a different matching model. Instead of matching individual peaks on theoretical and observed spectra as in SCOPE, it considers the joint matching states of the complimentary pair of b, y ions (i.e. both matched, b-ion matched, y-ion matched, neither matched) along each peptide bond, using a hidden Markov model. Though the intensities of observed peaks are considered in its scoring procedure through emission probabilities, they are incor-

porated only in matched states but not unmatched states. Thus, the emission probabilities for different hidden states are not defined on the same dimension, and consequently the comparison of emission probabilities between different states is not on the same probability space. Same as SCOPE, though it claims to model the probability that S is generated from p, if given a peptide sequence, it can not provide a recipe to generate an observed spectrum.

Because peak intensities are highly varied (see Li et al. (2006) for an assessment on noise structure of peak intensities), most scoring algorithms rely primarily on peak locations, and largely ignore peak intensities or only use peak intensities as filtering criteria for denoising. There are only several algorithms using intensities for scoring peptide candidates. For example, SCOPE and InsPecT weigh the matched peaks by the frequency that a given type of theoretical peaks is observed in literature. Though not mentioned in their papers, this formulation essentially is equivalent to incorporating intensities of theoretical peaks, assuming that the frequencies reported in literature approximate the theoretical intensities. Havilio et al. (2003) makes this approximation more explicitly, and weighs matched peaks by the frequencies of peak types. PepHMM incorporates the observed intensities through the emission probabilities of matched states in their hidden Markov model. Elias et al. (2004) scores peptides using a decision tree approach, where observed intensities are used as an attribute. Recently, Tabb et al. (2007) assumes the distribution of matched peaks for a random spectral match follows a hypergeometric distribution that is defined by discritized peak intensities, and scores peptides by computing the level of deviation from random matches. However, none of these methods take account of the noise structure of peak intensities.

1.5.4 Error sources of peptide identification

Though many scoring algorithms have been developed aiming to improve the accuracy of peptide identification, a large fraction of the top ranked peptides are still wrong for the following reasons:

- The scoring schemes used in current database search programs are all based on over-

simplified representations of the fragmentation process. For example, most scoring schemes only consider the location information for the most dominant fragmentation pathways and ignore the information on the intensities.

- The charge state of a spectrum often can not be accurately determined due to technical limitations (except when charge state is 1+). When the charge states are uncertain, a common strategy is to identify the spectrum at *all* possible charge states in order to cover the correct charge state. This strategy, however, introduces a large number of incorrect identifications when identifying using incorrect charge states.

- Some spectra have low quality and are not able to be identified.

- Some spectra are generated from more than one peptide.

- Some spectra are generated from peptides that are not in protein databases, due to the incompleteness of databases or modification on peptides.

Thus, the best matches in the database can not be assumed to be correct. They need to be further assessed to determine which identifications are correct.

1.6 Protein identification

The ultimate goal of a high-throughput proteomics approach is to determine the identity of the proteins present in biological samples. However, because MS/MS spectra are produced from peptides rather than proteins, all the conclusions drawn about the protein content are based upon the identification of peptides (Nesvizhskii and Aebersold, 2004). Because the connectivity between peptides and proteins is lost when complex protein samples are digested, the first step to identify proteins is to group the identified peptides according to their corresponding protein entries in the protein database. Next, for each protein, the combined information from its identified peptides is used to assess the evidence for its presence in the sample.

1.6.1 Challenges in protein identification

The inference from peptides to proteins is not straightforward due to two major challenges.

The first challenge arises from the high error rate of peptide identifications described in Section 1.5.4. Because putative proteins are constructed from identified peptides, a large number of incorrect protein identifications can be introduced by the incorrect peptide identifications.

The other challenge is due to the presence of degenerate peptides, which refer to the peptide whose sequence is present in more than one entry in the protein sequence database. Their presence makes it difficult to determine which corresponding proteins are present in the sample.

1.6.2 Current approaches for protein identification

Current practice for protein identification almost uniformly follows a two-stage strategy. In this strategy, the strength of evidence for each peptide identification is first evaluated, then protein evidence is estimated by combining peptide evidence. Here, I briefly review several commonly used computational approaches for each of the two steps.

Approaches for evaluating peptide evidence

An early approach to separate correct from incorrect peptide identifications is to apply an *ad hoc* cutoff value of the database search scores, often in conjunction with some properties of the assigned peptide and expert inspection. However, the score distributions produced by a search tool vary across experiments, making comparisons across different experiments or groups impossible.

Several computational and statistical methods have been developed for assessing the strength of evidence for peptide identifications (See (Nesvizhskii and Aebersold, 2004; Nesvizhskii et al., 2007) for recent reviews). For example, Pep_Probe (Sadygov and Yates, 2003) uses a hypergeometric distribution to model the matches between an observed spectrum and a random peptide candidate, and assesses each identification by the significance

level of the deviation from random matches. Recently, Kall et al (Kall et al., 2007) uses a semi-supervised approach to discriminate correct and incorrect peptide identifications based on a support vector machine. PeptideProphet (Keller et al., 2002) seems to be by far the most widely used approach. It uses a mixture model, based on certain parametric assumptions, to cluster the identified peptides into correct and incorrect identifications, according to identification scores and information related to peptides. It was extended recently to relax the restriction of the parametric assumptions (Choi and Nesvizhskii, 2008a), and to allow the option of semi-supervised learning (Choi and Nesvizhskii, 2008b).

Approaches for protein identification

Protein identification is usually performed by grouping identified peptide sequences into proteins deterministically or probabilistically.

A classical deterministic rule is to accept a protein identification if two distinct peptides on the identified protein are classified as being correctly identified according to their individual peptide evidence. For example, DTASelect (Tabb et al., 2002) groups peptides into proteins and reports proteins selected by a user-specified deterministic rule. Though this method does not provide any uncertainty measure for the identified proteins, it is still commonly used.

Several methods have been developed to provide quantitative assessment for protein identification, based on the individual peptide probabilities that are calculated in the first stage, for example, ProteinProphet (Nesvizhskii et al., 2003), Prot_Probe (Sadygov et al., 2004) and EBP (Price, 2007). The most widely used method among them is ProteinProphet (Nesvizhskii et al., 2003). It computes the probability that a protein is present in the sample by computing the probabilities that one or more identified peptides are correctly identified. It adjusts the individual peptide probabilities, calculated by e.g. PeptideProphet, according to the composite peptide information on the corresponding protein. It handles degenerate peptides by sharing each such peptide among all its protein parents, and estimates the weight that each degenerate peptide contributes to each protein parent in an *ad hoc* way.

For proteins with many shared peptide identifications, such as homologs, it groups them into a single entry, and reports the probability of the group as the probability that one or more identified peptides in the group are correctly identified. It then derives a minimal protein list sufficient to account for the identified peptides.

1.7 Outline of dissertation

Mass spectrometry data are complex, noisy, high-dimensional and usually with massive sizes. The data-analytic challenge of processing these types of data is to find the right balance between uncovering scientifically meaningful structure and avoiding erroneously identifying seemingly meaningful patterns that are actually the result of experimental noise.

This dissertation develops statistical methods for the two steps towards protein identification: *identifying peptides from mass spectra using database search*, and *identifying proteins from putative peptide identifications*. For both problems, our goal is to use statistical modeling approaches to take account of the complicated noise structure in the data, and ultimately improve the accuracy of identification.

In Chapter 2, we develop a likelihood-based scoring algorithm for identifying peptides from mass spectra using database search. Chapter 3 develops a nested mixture model for identifying proteins from putative peptide identifications. In Chapter 4, we conclude and discuss future work.

Chapter 2

PEPTIDE IDENTIFICATION USING MASS SPECTROMETRY

2.1 Introduction

Peptide identification by tandem mass spectrometry is a key component to identify proteins in complex biological samples. In the experimental procedure, the proteins in the sample are first broken into short peptides, then the resulting peptide mixture is subjected to mass spectrometry, which generates spectra that are characteristic of peptides. The task of peptide identification is, for each observed spectrum, to identify the peptide that generated the spectrum.

Currently, the most widely used computational methods for high-throughput peptide identification is database search, which makes use of the protein database of the species that the sample is generated from. It assumes that (1) the protein database contains all the peptides that could possibly generate the observed spectra, and (2) each observed spectrum is generated from one peptide in the database. Based on these premises, it identifies peptides by finding the peptide sequence in the database that is most likely to generate each observed spectrum. It generally includes three steps (Figure 1.4). First, for each observed spectrum, a list of candidate peptide sequences that satisfy certain selection criteria (e.g. similar mass to the observed spectrum) is selected from the protein database. Then the theoretical spectrum for each candidate sequence is generated according to the fragmentation pathways that generates the observed spectra. Each candidate peptide then is scored by the similarity between its theoretical spectrum and the given observed spectrum, and the peptide sequence with the best score is the potential identification.

Though simple in principle, generating correct peptide identification is not straightforward in practice. For example, some mass spectra may be generated from more than one

peptide sequence, or sometimes peptides that generate the observed spectra may not be in the protein databases. In addition, because experimental spectra often have highly noisy and complicated patterns, which are difficult to predict in the theoretical spectra, there is a big deviation between the theoretical spectrum and the observed spectrum from the same sequence. As a result, identification of peptide sequences using mass spectrometry remains a challenging task.

In the past decade, a number of scoring algorithms for peptide identification have been developed (see Chapter 1 for details). For example, the first and a widely-used program, SEQUEST (Eng et al., 1994), used a cross-correlation scoring function. Mascot (Perkins et al., 1999), another widely-used program, measures the significance of a match by comparing with the random matches. Recently, many programs that use probabilistic-based scoring functions have been developed. Among them, PepHMM (Wan et al., 2006) uses a hidden Markov model to score the similarity, and InsPecT (Tanner et al., 2005) uses a dynamic programming algorithm to align theoretical spectra and observed spectra.

Despite the differences of scoring schemes, these methods all score candidate peptides using very coarse theoretical spectra, which contain locations of peaks from only few fragmentation pathways and without much differentiation of peak intensities. Furthermore, all of these scoring algorithms focus primarily on information on peak locations, and largely ignore the information on peak intensities. However, as part of the spectral signature, peak intensities and peaks from a more complete set of fragmentation pathways provide useful information for distinguishing specific matches between the given observed spectrum and its corresponding theoretical spectrum from the random matches. Thus, one would expect improved identification accuracy, with the use of (a) a fine theoretical spectrum that contains a comprehensive set of peaks and accurate prediction of intensities and (b) a scoring algorithm that effectively takes account of information on both peak locations and intensities when assessing the similarity between the theoretical and the observed spectra.

As the aforementioned assumptions in database searches do not always hold in actual experiments, the peptide with the best score is not necessarily the right sequence. Thus,

it is necessary to assess the uncertainty of each identification, in order to determine if the best-scored sequence will be accepted as the identification. Ideally, one would hope these uncertainty measures are statistically justifiable and well-calibrated, which unfortunately is not provided by many existing approaches due to their heuristic nature. Though there are post-processing methods that re-assess the uncertainty of identifications based on the scores of all the identifications using machine learning methods (e.g. PeptideProphet (Keller et al., 2002), Percolator (Kall et al., 2007)), it is still desirable to assess the uncertainty with respect to all the candidate peptides for the same observed spectrum, especially when scores are confounded with other factors, for example, peptide length, or when there are not enough spectra for the postprocessing methods to learn the discriminative features sufficiently accurate.

In this study, we propose a likelihood-based scoring algorithm for peptide identification based on a generative model. Our model views the observed spectrum as a noisy version of the theoretical spectrum, which is generated from a prediction algorithm (Zhang, 2004) that predicts both locations and intensities for peaks from a comprehensive set of fragmentation pathways. Our generative model takes account of multiple sources of noise in the data, including variable peak intensities and errors in peak locations. Our likelihood-based approach also provides two measures for assessing the uncertainty of each identification.

Our approach includes the following steps:

1. Generate theoretical spectra from candidate sequences, where we will use a prediction algorithm that predicts both locations and intensities of peaks on the theoretical spectra Zhang (2004).

2. Preprocess observed spectra to remove the dependence between neighboring peaks and denoise.

3. Preprocess theoretical spectra to remove the dependence between neighboring peaks and denoise.

4. For each observed spectrum O, score each candidate sequence by the likelihood ($P(O \mid T_i)$) that the observed spectrum is generated from its theoretical spectrum T_i.

5. For each observed spectrum, assess the uncertainty of the identification to determine if the candidate with the highest likelihood will be called as the identified peptide.

The organization of this chapter is as follows. Section 2.2 describes the generation of theoretical predictions. Section 2.3 describes the preprocessing process. Section 2.4 describes our generative model. Section 2.5 describes how to score peptide sequences using the generative model. Section 2.6 describes the estimation procedure. Section 2.7 presents two ways to measure the uncertainty of a peptide identification. In section 2.8, we use simulated data to assess the performance of the estimation procedure. In section 2.9 we illustrate our method using a publicly available benchmark dataset. In section 2.10 we conclude and suggest future enhancement.

2.2 Prediction of theoretical spectra

We use the prediction algorithm developed by Zhang (2004) to predict the theoretical spectra for peptide sequences. This algorithm offers two major advantages over simple theoretical predictions used in many peptide identification algorithms. First, it produces theoretical spectra with a more complete set of peaks than the simple prediction because it incorporates a comprehensive set of fragmentation reactions. Second, its prediction of intensities, though still rough, captures some of the dependence of the observed intensities to the amino acid composition of the peptide sequence. Figure 2.1 shows the observed spectrum from a peptide with 7 amino acids and the theoretical spectrum predicted based on my implementation of Zhang's algorithm, respectively.

Because the distributed version of Zhang's algorithm does not allow batch prediction, I re-implemented a batch version based on the model published in Zhang (2004) using Java. Since only peptide sequences with charge 1+ and charge 2+ are predicted in Zhang (2004) (though later charge 3+ is considered in an extended model in Zhang (2005)), my

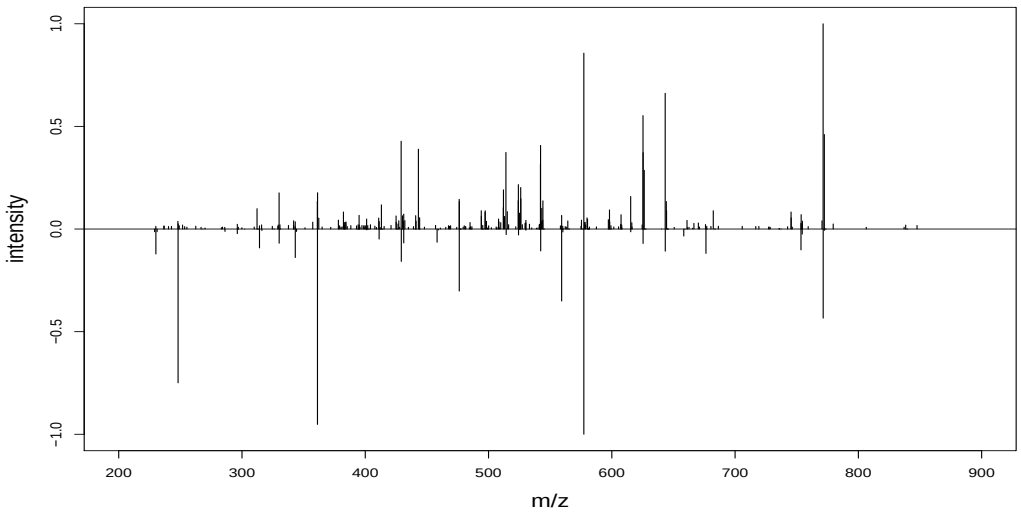


Figure 2.1: Observed and theoretical spectra for peptide sequence LVTDLTK. Top: Observed spectrum, Bottom: Theoretical spectrum generated using my implementation of Zhang's algorithm. Both spectra are rescaled by dividing by their highest peak for better visualization.

implementation can only predict theoretical spectra for peptide with these two charge states.

To check the correctness of implementation, I compared the predictions from my implementation with those from Zhang's algorithm on several peptide sequences. The results show that they have similar general patterns, though small differences exist.

We use the theoretical spectra generated from my implementation of Zhang's algorithm in our peptide identification. It is worth noting that, though this algorithm provides much closer predictions to observed spectra than the naive ones, there are still marked deviations between these predictions and their corresponding observed spectra. Hence, discriminating specific matches between an observed spectrum and its corresponding theoretical spectrum from random matches remains a major challenge in this work.

2.3 Preprocessing

Preprocessing has been found essential for peptide identification (personal communication with Michael MacCoss and Jimmy Eng). Here we develop a preprocessing procedure that facilitates our modeling of noisy peaks and variable intensities. It mainly includes three steps: (1) first cleans spectra using a procedure that is adaptive to local peak intensities, then (2) normalizes peaks on theoretical and observed spectra to make their intensities on a comparable scale, and (3) stabilizes peak intensities by transformation. Both theoretical and observed spectra are processed using this procedure before their similarities are scored. The details of this procedure are described as follows, and the detailed steps can be found in Table 2.1.

Our cleaning procedure is motivated by the observation that peaks on mass spectra form clustering patterns (Figure 2.2a). To simplify the modeling process, we distill the data down to the primary signals by keeping one member for each cluster of peaks. As informative peaks usually have relatively high intensities, we represent the peaks in each cluster by their local modes. To proceed, we first smooth the spectra by binning the peaks according to their locations, then summarize the peaks in each bin by summing their peak intensities. We then select the bins that are local modes of the summed intensities, and represent each

of the selected bins by the highest peak that is within a specified distance to the center of each bin. These peaks form the cleaned spectrum.

This approach serves several purposes. First, it reduces the dependence between peaks, which allows our generative model to assume peaks are independent to each other. Second, it denoises the spectra by removing the low-intensity peaks in local clusters, and avoids modeling the less important information that does not need modeling. In addition, unlike removing low-intensity peaks by thresholding, this denoise procedure is adaptive to local peak intensities. As peak intensities on mass spectra are highly varied, this adaptive feature ensures the peaks at different locations, regardless of their absolute heights, are represented after preprocessing. Figure 2.2 shows an example of spectra before and after preprocessing using our approach.

The cleaned spectra then undergo several further processing steps before scoring. First, as only the peaks in a certain range (200-2000 Da) can be observed on the observed spectra, the theoretical peaks that fall out of this range are trimmed. Then for each processed spectrum, peak intensities are normalized by dividing by the 90% percentile of the intensities of its peaks, to make the peaks on different spectra on a comparable scale. Normalization with respect to the significant peak on the same spectra is commonly used in other scoring algorithms; for example, SEQUEST normalizes peaks with respect to the highest peak within a certain distance Eng et al. (1994). We choose 90% percentile rather than the most significant peak because 90% percentile is less sensitive to outliers than the highest one. To stabilize the highly variable intensities, the normalized intensities are transformed by raising to 1/4 power.

Clearly, these steps involve *ad hoc* decisions, which may not generate optimal results. Therefore, we experimented several parameter choices, for example, different combinations of parameters in cleaning procedure ($b \in \{0.5, 1, 2\}$ and $c \in \{0.5, 1, 2\}$ in Table 2.1), and different transformations (e.g. square root transformation). Among all the parameters experimented, the parameters reported in Table 2.1 generate the best performance in terms of the correct identification rate on the training spectra. These decisions may depend on

Table 2.1: Preprocessing spectra.

1. Bin the peaks on a spectrum according to peak locations with prespecified binwidth b (e.g. $b = 2Da$). Let x_i^b be the locations of bin boundaries, then $x_i^b = x_1 + (i-1)b$, $i = 1, \ldots, n^b + 1$, where the number of bins $n^b = \lceil \frac{x_n - x_1}{b} \rceil$, and x_1 and x_n are the smallest and largest m/z value of peaks, respectively.

2. Sum peak intensities in each bin ($y_i^b = \sum_{\{j: x_j \in [x_i^b, x_{i+1}^b)\}} y_j, i = 1, \ldots, n^b$).

3. Find the bins that are local mode of summed peak intensities, defined as $M = \{i : y_{i-1}^b \leq y_i^b \bigcap y_i^b \geq y_{i+1}^b, i = 1, \ldots, n^b\}$, where $y_0^b = 0$ and $y_{n^b+1}^b = 0$.

4. Keep the highest peak within distance c from the center of the bins that are local modes, i.e. $K = \bigcup_{i \in M} \{j : argmax_{j \in [\frac{x_i^b + x_{i+1}^b}{2} - c, \frac{x_i^b + x_{i+1}^b}{2} + c)} y_j\}$ (e.g. $c = 2Da$).

5. Remove peaks with $\{k : x_k < 200 \bigcup x_k > 2000, k \in K\}$.

6. Normalize peak intensities by $y'_k = \frac{y_k}{y_{0.9}}$, where $y_{0.9}$ is the 90% percentile of $y_k, k \in K$.

7. Transform intensities by $y_k^* = (y'_k)^{\frac{1}{4}}$.

8. Keep $(x_k, y_k^*), k \in K$ to form the processed spectrum.

the instrument and datasets.

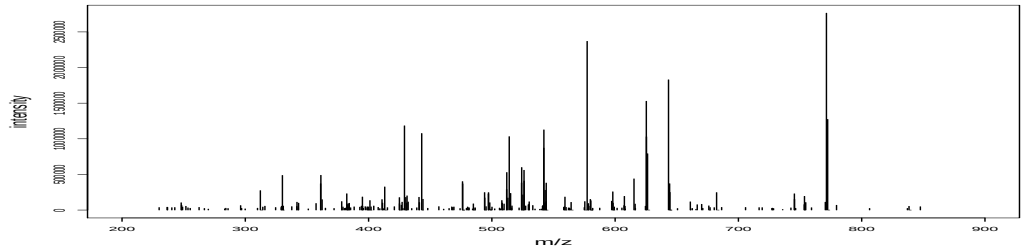
2.4 Statistical Methods

2.4.1 A generative model

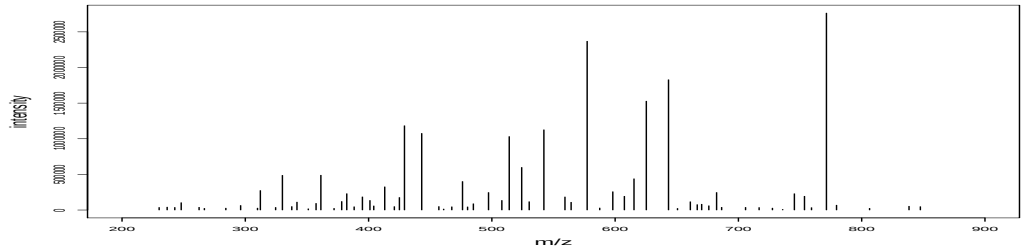
Let $\mathbf{T}$ be a theoretical spectrum with n number of spectral peaks $\mathbf{T} = (T_1, \ldots, T_n)$, where $T_i = (X_i^t, Y_i^t)$ denotes the location (X_i^t) and intensity (Y_i^t) for the ith peak; and $\mathbf{O}$ be an observed spectrum with m number of spectral peaks $\mathbf{O} = (O_1, \ldots, O_m)$, where $O_i = (X_i^o, Y_i^o)$.

If $\mathbf{T}$ and $\mathbf{O}$ are from the same peptide sequence, then we consider $\mathbf{O}$ as a distorted realization of $\mathbf{T}$ generated from the following generative model. Our generative model assumes each theoretical peak T_i have a probability p_i to emit an observed peak O_j that is normally distributed around T_i, and a probability $(1 - p_i)$ to not emit any observed peak. As the emitted observed peaks should be close to their corresponding theoretical peaks, the distribution of X_j^o should be truncated at $[x_i^t - w, x_i^t + w]$, where w is a positive constant related to the resolution of instrument. The observed peaks that are not emitted from theoretical peaks are noise randomly distributed on the observable range of m/z scale. Then the emission statuses for the peaks on the theoretical spectrum can be represented as $\mathbf{e}^t = (e_1^t, \ldots, e_n^t)$, where $e_i^t = 0$ if the ith peak does not emit any observed peak, and $e_i^t = j$, $(j = 1, \ldots, m)$, if the ith peak emits peak O_j. Similarly, the emission statuses for the peaks on the observed spectrum can be represented as $\mathbf{e}^o = (e_1^o, \ldots, e_m^o)$, where $e_j^o = 0$ if the jth peak is a noise peak, and $e_j^o = i$, $(i = 1, \ldots, n)$ if the jth peak is emitted from the ith theoretical peak. Note that $\mathbf{e}^o$ and $\mathbf{e}^t$ are equivalent, since they contain the same information. Then the number of emission peaks can be denoted as $k = |\{i : e_i^t > 0\}| = |\{j : e_j^o > 0\}|$.

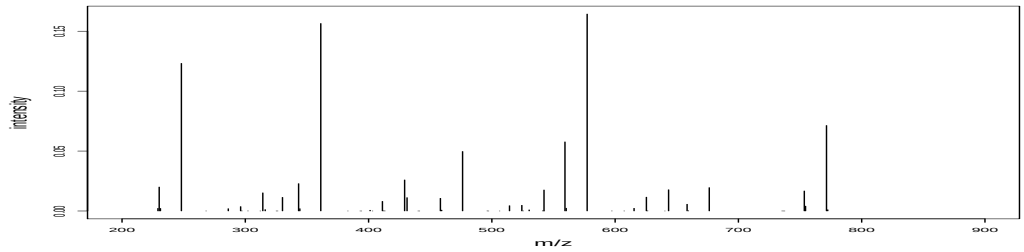
With the notation of emission statuses, the distribution of peak locations X_j^o can be



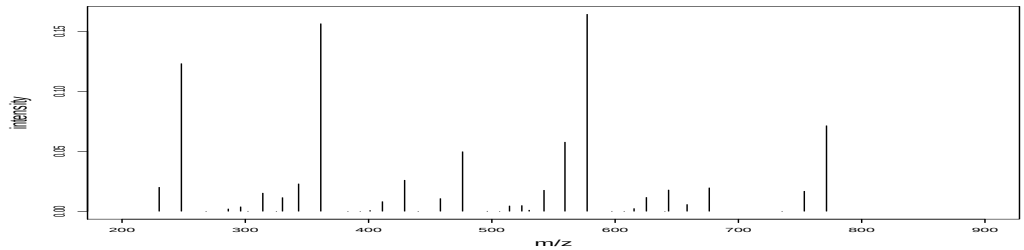
a. Raw observed spectrum



b. Cleaned observed spectrum



c. Raw theoretical spectrum



d. Cleaned theoretical spectrum

Figure 2.2: The observed and theoretical spectra of peptide sequence LVTDLTK before and after cleaning.

written as

$$X_j^o \mid e_j^o > 0 \sim N_T(x_{e_j^o}^t, \sigma^2, w), \tag{2.4.1}$$

$$X_j^o \mid e_j^o = 0 \sim Uniform(a_1, a_2), \tag{2.4.2}$$

where $N_T(\cdot)$ is a truncated normal density as

$$N_T(d_j; 0, \sigma^2, w) = \frac{N(d_j; 0, \sigma^2)}{1 - 2\Phi(-w/\sigma)} \quad \text{for } d_j \in [-w, w]$$

where $d_j = x_j^o - x_{e_j^o}^t$, (a_1, a_2) is the m/z range where peaks can be detected, Φ is the standard normal cdf and w is assumed to be given.

Given an observed spectrum, we will score a candidate peptide l by the likelihood that the observed spectrum arises from the theoretical spectrum $\mathbf{T}_l$ of this candidate peptide. Note that, indeed, the observed spectra are actually unlabeled. Though the observed peaks on $\mathbf{O}$ are labeled, the label is arbitrary and is only a device for notational convenience. In addition, $p(\mathbf{O}_\phi \mid \mathbf{T}_l)$ is identical for any permutation ϕ of $1, \ldots, m$. Thus, if we denote the unlabeled observed spectrum with $\{\mathbf{O}\}$, the score for $\mathbf{T}_l$ is $p(\{\mathbf{O}\} \mid \mathbf{T}_l) = \sum_\phi p(\mathbf{O}_\phi \mid \mathbf{T}_l) = m! p(\mathbf{O}_\phi \mid \mathbf{T}_l)$.

2.4.2 Model with only peak locations

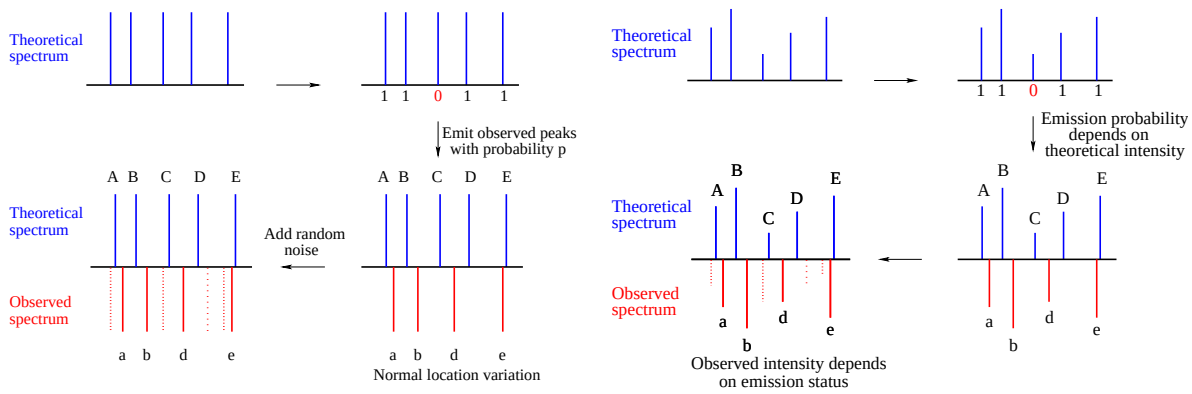
Model peak locations

We first consider a model with only peak locations but no intensities (Figure 2.3a) as our basic model. In this model, the emission probability p for all the theoretical peaks is identical.

Likelihood for the model with only locations

For a given $\mathbf{e^t}$, assuming the peaks on the same spectrum are independent, then the model with only locations for a labeled observed spectrum is

$$P(\mathbf{O} \mid \mathbf{T}, \mathbf{e^t}) = \left(\frac{1}{r}\right)^{m-k} \prod_{j \in \{j : e_j^o > 0\}} N_T(d_j; 0, \sigma^2), \tag{2.4.3}$$



a. Model peak locations

b. Model peak intensities

Figure 2.3: Our generative model.

where $d_j = x_j^o - x_{e_j^o}^t$ is the difference of locations between the jth observed peak and the corresponding theoretical peak, $r = a_2 - a_1$ is the range of locations that a noise peak may appear.

Because the emission statuses $\mathbf{e}^t$ is unknown, ideally the likelihood should be summed up across all the possible configurations. The likelihood for the unlabeled observed spectrum is as follows:

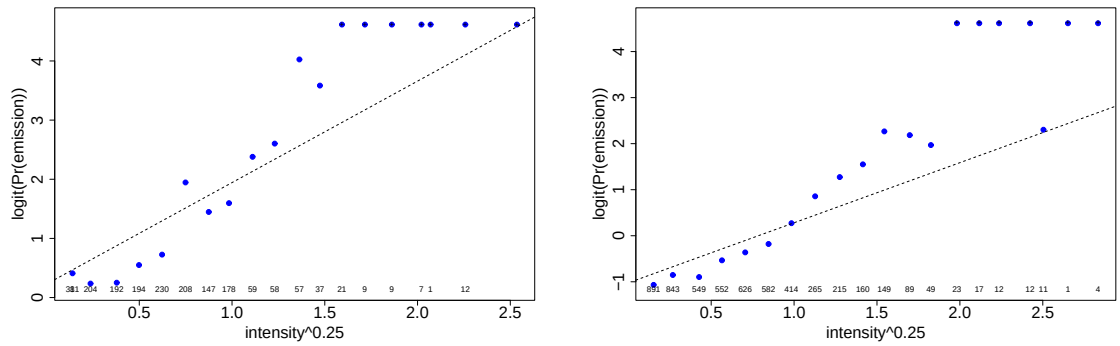
$$L(p,\sigma) = P(\{\mathbf{O}\} \mid \mathbf{T}) = m!P(\mathbf{O} \mid \mathbf{T}) = m!\sum_{\mathbf{e}^t} P(\mathbf{O} \mid \mathbf{T}, \mathbf{e}^t)P(\mathbf{e}^t) \tag{2.4.4}$$
$$= m!\sum_{\mathbf{e}^t}\left(\frac{1}{r}\right)^{m-k}\Big[\prod_{\{j:e_j^o>0\}} N(d_j;0,\sigma^2)\Big]\frac{(m-k)!}{m!}p^k(1-p)^{n-k},$$

where k is the number of emission peaks.

2.4.3 Model with both locations and intensities

Model theoretical intensities

Empirical observations (Figure 2.1) show that high-intensity theoretical peaks are more likely to appear in observed spectra. To incorporate this tendency, we model the emis-



a. Charge 1+ b. Charge 2+

Figure 2.4: Linear trends between theoretical intensities and logit of estimated emission probabilities. The theoretical intensities are normalized and transformed as described in section 2.3, and are binned (20 bins) with a fixed binwidth. The emission probabilities are estimated as the proportions of putative matches (i.e. the observed peaks and the theoretical peaks that locate less than 2Da apart) in each bin from training data. The number of observations in each bin is marked at the bottom of the plot. The dashed line is the regression line.

sion probabilities as a function of theoretical intensities using a logistic regression. For a theoretical spectrum s,

$$\log \frac{p_{s,i}}{1 - p_{s,i}} = \mu_s + \beta y^t_{s,i}, \tag{2.4.5}$$

where $p_{s,i}$ is the emission probability of the ith theoretical peak on the sth spectrum, μ_s is the spectrum-specific intercept of the logistic regression, and the slope β is common for all spectra. The intercept is chosen to be spectrum-specific to take account of the variation between spectra. The linearity between the logit of $p_{s,i}$ and $y^t_{s,i}$ is verified with empirical data (Figure 2.4).

Model observed intensities

It is known that peaks in the observed spectra that match to peaks in the theoretical spectra tend to have higher intensities than noise (Figure 2.1). However, the association

between the theoretical peak intensities and the observed peak intensities is weak. To capture this weak association, we model the observed intensities as being dependent on the emission statuses (either $e_j^o = 0$ or $e_j^o > 0$) of the correspondent observed peaks. Figure 2.5 shows the empirical distributions of the intensities of observed peaks at different emission statuses.

One approach to modeling the intensity of observed peaks in each emission state is to use a 2-component mixture of gamma distributions.

$$f_0(y_j^o) \equiv p(y_j^o \mid e_j^o = 0) = \pi_0 g_h(y_j^o) + (1 - \pi_0) g_l(y_j^o) \tag{2.4.6}$$

$$f_1(y_j^o) \equiv p(y_j^o \mid e_j^o > 0) = \pi_1 g_h(y_j^o) + (1 - \pi_1) g_l(y_j^o), \tag{2.4.7}$$

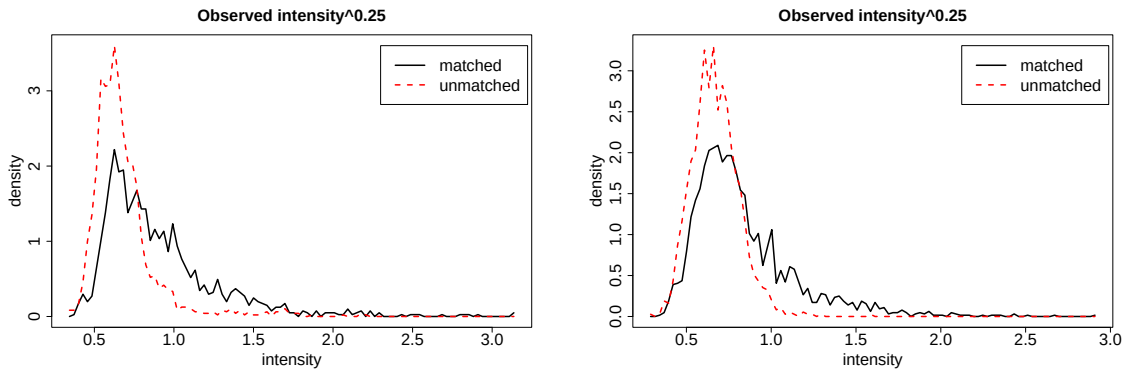
where g_h and g_l are gamma distributions to represent distributions of the high intensity component and the low intensity component, respectively; π_0 and π_1 are mixing proportions for emitted peaks and noise peaks, respectively.

Another approach is to use histogram density estimators to estimate $f_0(y_j^o)$ and $f_1(y_j^o)$. The advantage of histogram density estimators is that their estimation of the densities of the right tails, where most informative peaks lie, is more adaptive to the empirical distributions than the parametric approach.

Likelihood for the model with intensities

If we represent the parameters in f_0 and f_1 with Ψ_0 and Ψ_1, the likelihood of the generative model with intensities is as follows:

$$L(\mu, \beta, \sigma, \Psi_0, \Psi_1) = \sum_{\mathbf{e}^t} \Big[(m-k)! \big(\frac{1}{r}\big)^{m-k} \prod_{\{j: e_j^o = 0\}} f_0(y_j^o) \big[\prod_{\{j: e_j^o > 0\}} N(d_j; 0, \sigma^2) f_1(y_j^o) \big] \prod_{\{i: e_i^t > 0\}} p_i \prod_{\{i: e_i^t = 0\}} (1 - p_i) \Big] \tag{2.4.8}$$



a. Charge 1+ b. Charge 2+

Figure 2.5: Empirical distribution of intensities for observed peaks. The emission status of each observed peak is approximated by whether there exists a theoretical peak that is less than 2Da apart from the observed peak. The observed intensities are normalized and transformed as described in section 2.3, and are binned (100 bins) with even binwidth for plotting.

2.5 Complete data likelihood

Because mass spectra usually have a large number of peaks, there are many possible configurations for $\mathbf{e}^t$ and $\mathbf{e}^o$. Exact computation for the generative likelihoods above is computationally expensive. Fortunately, because the emitted observed peaks should locate close (e.g. within 2 Da) to the corresponding theoretical peaks, we only need sum over the configurations that satisfy the distance constraint.

In many cases, at most one observed peak satisfies the distance constraint of a theoretical peak, then the mapping between observed peaks and theoretical peaks is clear. However, ambiguities will arise, when multiple observed peaks are within such a distance of one theoretical peak or vise versa. Therefore, we use the complete data likelihood under the most probable configuration instead, which is analogous to the classification likelihood (McLachlan and Peel, 2000) in clustering. As it is known that the classification likelihood could perform poorly in some cases, for example, producing biased estimates, we will use simula-

tion studies to assess the performance, in terms of parameter estimation and classification errors of the emission vectors.

The complete data likelihoods for models with and without intensities are as follows. Model without intensities:

$$L_0(p,\sigma) = \max_{\mathbf{e}^t} \Big[(m-k)!\big(\frac{1}{r}\big)^{m-k}\big[\prod_{\{j:e_j^o>0\}} N(d_j;0,\sigma^2)\big]p^k(1-p)^{n-k}\Big] \tag{2.5.1}$$

Model with intensities:

$$L_1(\mu,\beta,\sigma,\Psi_0,\Psi_1) = \max_{\mathbf{e}^t}\Big[(m-k)!\big(\frac{1}{r}\big)^{m-k}\prod_{\{j:e_j^o=0\}} f_0(y_j^o)\big[\prod_{\{j:e_j^o>0\}} N(d_j;0,\sigma^2)f_1(y_j^o)\big] \tag{2.5.2}$$

$$\prod_{\{i:e_i^t>0\}} p_i \prod_{\{i:e_i^t=0\}} (1-p_i)\Big]$$

For each candidate sequence, we will search for the most probable configuration between its theoretical spectrum and the corresponding observed spectrum, and compute the complete data likelihood. The resulting likelihood is the score for the candidate peptide.

2.6 Initialization, parameter estimation and scoring

2.6.1 Initialization

As shown in Section 2.4, prior to scoring a peptide sequence, our model requires a mapping between the theoretical spectrum of the sequence and the corresponding observed spectrum. As mass spectrometers usually have a good resolution on peak locations, a theoretical peak and its corresponding observed peak should locate closely, for example, within 2Da. Thus, we initiate the alignment by mapping each theoretical peak to the observed peak(s) within such a prespecified distance. When ambiguity of mapping arises, for example, multiple observed peaks compete for one theoretical peak or vice versa, the peaks involved in the ambiguous assignments are grouped together. After matching, there are three possible cases: peaks without matches, peaks with one matched peak, and peaks with ambiguous matches. The peaks without matches at initialization will stay unmatched. The potential matched

peaks in the latter two cases are called putative matches or putative emission pairs, whose matching statuses will be determined in the scoring procedure.

2.6.2 Parameter estimation using training data

As the parameters in the likelihood describe the characteristics of peaks on the observed spectra and the theoretical spectra from the *same* sequences, we estimate them using the observed spectra and the theoretical spectra of their *correctly* identified peptides in a training dataset.

Since different pairs of spectra have notably different patterns of putative emission (e.g. proportion of matched peaks), we treat some parameters related to emission probabilities (p in (2.5.1) or μ in (2.5.2)) as spectrum-specific to account for the variation across spectra, and treat the rest parameters roughly constant across all spectra. One way is to use a Bayesian approach, where the spectrum-specific parameters can be modeled as following a certain distribution. For simplicity, we do not adopt the Bayesian approach, though it may be more appealing. Instead, we estimate the parameters that are not spectrum-specific using *pooled* spectra, and the spectrum-specific parameters for each *individual* spectrum pair.

As even for the training data, the configurations $\mathbf{e}^t$ and $\mathbf{e}^o$ are unobservable, the training procedure alternates the estimation of the parameters and the search of the most probable configurations, to maximize the likelihood function. In parameter estimation, the peaks in *all* the training spectra are first pooled according to their current configurations. Then the parameters that are constant across spectra are estimated by maximizing the likelihood of the *pooled* peaks. The spectrum-specific parameter is maximized for *individual* spectrum pairs, along with the configuration search. In the search of the most probable configuration, the peaks that satisfy the distance constraint are grouped, then configurations are updated by traversing all the groups and selecting the most probable configuration (i.e. the configuration resulting the highest likelihood) in each group in a greedy fashion. To minimize the greedy feature of this search, the update procedure is repeated multiple times with different

random traversing schedules. As a configuration is updated only when the likelihood is increasing, this procedure guarantees the likelihood will be nondecreasing in the iterations. The likelihood will converge after finite steps, as the number of peaks is finite. Table 2.2 provides a detailed description of the iterative training procedure. The objective function and estimation details are described in Appendix A.

2.6.3 Scoring test data

The spectrum-nonspecific parameters estimated from training data are used to score the candidate sequences for the observed spectra in the test data. In the scoring procedure for test data, the likelihood is maximized only with respect to configurations and spectrum-specific parameters, i.e. it only performs step 2(b) in Table 2.2.

2.7 Uncertainty of identifications

Because it is possible that none of the candidate sequences is the one that generates the observed spectrum, the top-scored candidate sequence is not guaranteed to be the correct identification, even for an ideal scoring algorithm. It is important to measure the uncertainty of identification in order to determine if the top-scored candidate sequence should be called as the identification.

We propose two uncertainty measures. One is the distinguishability of the scores, defined as follows:

$$D = \log(L_{best}) - \log(L_{2nd\ best}) \tag{2.7.1}$$

where L is the score computed from our model. It measures how much the top-scored candidate differs from the next-best candidate. As our score is likelihood-based, this quantity is the log-likelihood ratio of the top two candidate sequences.

Another one is the posterior probability that an observed spectrum is generated from a certain candidate sequence l, i.e. $P(\mathbf{T}_l \mid \mathbf{O})$, which assesses the uncertainty of the identification directly. If all the candidate sequences are assumed to have the same prior probability

Table 2.2: Procedure for estimating parameters from training data.

1. Initialization:
   (a) Group peaks that satisfy the distance constraint as putative emissions.
   (b) Pair up peaks with the closest m/z distances as the initial configuration.
2. Alternate 2(a) and 2(b):
   (a) Estimate parameters (see Appendix A for details):
      i. Pool the theoretical and observed peaks in all training spectra according to their current emission statuses.
      ii. Estimate $(\sigma, \beta, \Psi_0, \Psi_1)$ from the pool by maximizing likelihood (2.5.2).
   (b) Configuration update:
      For each theoretical spectrum, repeat below until the likelihood converges.
      i. Generate a random schedule for traversing peak groups.
      ii. For each group on the traversing schedule:
         A. Maximize the likelihood wrt p (w/o intensity) or μ (with intensity) for each configuration.
         B. Update the configuration with the one that generates the highest likelihood.

to generate the observed spectrum, the posterior probability can be approximated by normalizing the likelihood over all the candidates.

$$P(\mathbf{T}_l \mid \mathbf{O}) = \frac{L_l}{\sum_{i \in C_o} L_i} \tag{2.7.2}$$

where C_o is the collection of candidate sequences for the observed spectrum O.

The formulation of the two quantities shows that the posterior probability weighs all the candidates, rather than just the top two candidates, thus it is more desirable in principle. For example, it distinguishes the case, where the spectrum is generated from two top candidate sequences (i.e. top two have close high scores but differ from the rest), from the case, where the spectrum is not generated from any of the candidate sequences (i.e. all have close low scores); whereas, D does not distinguish these two cases.

2.8 Simulation studies

We use a scoring procedure based on the complete data likelihood at the most probable configuration (2.5.2). To check the performance of this approach, we simulate observed spectra from theoretical spectra using the generative model, then estimate parameters and emission labels using the complete data likelihood. The performance then is assessed by the accuracy of parameter estimations and classification errors of the estimated emission statuses.

To generate simulations similar to realistic datasets, we first estimate parameters from a training data (detailed description in Section 2.9), which consists of 50 randomly selected charge 1+ and charge 2+ spectra from a curated real dataset, using the estimation procedure described in Table 2.2 with our model with intensities (2.5.2). The observed intensities are estimated using the mixture of two Gamma components. The estimated parameters (Table 2.3) then are used as simulation parameters. Because charge 1+ and charge 2+ spectra have different peak characteristics, they are simulated and analyzed separately.

To simulate observed spectra, we randomly picked one theoretical spectrum in each charge state (charge 1+: n=27 peaks; charge 2+: n=41 peaks) and generate 1000 observed

Table 2.3: Parameters used for simulation studies. They are estimated from the training data. Here, p^* is the average of emission probabilities, which is calculated using the logistic regression in (2.4.5) with σ and μ; γ is the noise rate, estimated from empirical data, based on the proportion of observed noise peaks.

	σ	μ	p*	β	π_m	π_u	(α_s, β_s)	(α_n, β_n)	γ
charge 1+	0.39	-1.24	0.601	2.97	0.724	0.104	(8.45, 0.125)	(32.08, 0.020)	0.9
charge 2+	0.16	-5.06	0.223	4.74	0.885	0.035	(9.21, 0.114)	(24.89, 0.027)	0.32

spectra for each theoretical spectrum from our full model, using the following procedure:

(a) Calculate p_i for each peak using (2.4.5) with the parameters estimated from the training set.

(b) Sample e_i^t according to p_i calculated in (a)

(c) For each theoretical peak $i \in \{i : e_i^t > 0\}$, generate $X^o_{e_i^t}$ from (2.4.1) and $Y^o_{e_i^t}$ from (2.4.6).

(d) Adding random noise at locations X_j^o sampled from (2.4.2), with intensity Y_j^o sampled from (2.4.7). The noise rate is a random number between $(\frac{2}{3}\gamma, \frac{4}{3}\gamma)$, where γn is the number of noise peaks.

When estimating parameters from the simulation, we consider three approximation models: the model with only locations (L_0), the model with locations and both theoretical and observed intensities (L_1), and the model with locations and theoretical intensities (L_t), which is a reduced model of L_1 by removing the terms involving observed intensities. In the estimation, we fix β, Ψ_0, Ψ_1 at the simulation value and only estimate μ and σ.

The results (Table 2.4) show that the estimated parameters are close to the true values, which indicates the complete data likelihood is adequate for parameter estimation. Though the complete data likelihood and the generative likelihood still would not agree numerically, empirically the occurrence rate of ambiguity cases is low (< 3 per spectrum) and only involves 2-3 peaks each time. Thus it seems complete data likelihood is a reasonable approximation of the generative model in this application.

As shown in model L_t, the incorporation of theoretical intensities reduces the average

Table 2.4: Parameter estimation and classification errors (CE) from simulated data, when the observed intensities are modeled using Gamma mixtures. CE_T is CE of emission labels for peaks on the theoretical spectra after estimation, CE_O is CE of emission labels for peaks on the observed spectra after estimation, and CE_p is CE of the emission labels for theoretical peaks before estimation (i.e. initial putative emissions).

	charge 1+			charge 2+		
	L_0	L_t	L_1	L_0	L_t	L_1
$\hat{p}$	0.641 (0.094)	-	-	0.25 (0.045)	-	-
$\hat{\mu}$	-	-1.053 (0.533)	-1.072 (0.533)	-	-4.81 (0.257)	-5.01 (0.421)
$\hat{\sigma}$	0.394 (0.106)	0.379 (0.089)	0.377 (0.085)	0.177 (0.097)	0.157 (0.035)	0.155 (0.033)
CE_T	0.054 (0.046)	0.048 (0.042)	0.045 (0.042)	0.028 (0.028)	0.017 (0.016)	0.013 (0.014)
CE_O	0.059 (0.046)	0.055 (0.044)	0.053 (0.044)	0.057 (0.052)	0.037 (0.032)	0.029 (0.029)
CE_p	0.074 (0.050)	0.074 (0.050)	0.074 (0.050)	0.087 (0.031)	0.087 (0.031)	0.087 (0.031)

classification error of emission labels. The incorporation of observed intensities in model L_1 then reduces the average classification error further.

2.9 Applications on the ISB data

We illustrate our method on a mass spectra data set from Institute of System Biology Keller (2002). This dataset contains a mixture of 18 purified proteins. The dataset was analyzed using SEQUEST, with 504 peptides assigned to spectra of charge 1+, 18496 to spectra of charge 2+, 18044 to spectra of charge 3+. The dataset provided a list of 10-11 top-ranked candidates by SEQUEST for each spectrum. The top-ranked peptide assignments were then manually scrutinized to determine if they appeared correct (personal communication with Jimmy Eng). The spectra passing hand curation formed a dataset consisting of 125 1+ spectra, 1640 2+ spectra, and 1010 3+ spectra. We call this dataset the hand-curated dataset in the rest of the text.

In this study, we only consider charge 1+ and charge 2+ spectra, because my implementation of the prediction algorithm can only generate theoretical spectra for sequences with

those two charges. The identification of spectra with charge states of 3+ or higher could follow the same model and procedure. We choose $w = 2$ for this dataset.

2.9.1 Evaluation on the curated dataset

We first evaluate the performance of our method on the hand-curated dataset. For each charge state, we randomly choose 50 observed spectra as the training data, and use the rest for testing. The spectra are identified using our models with only peak locations (2.4.4) and with both peak locations and intensities (2.5.2), respectively. The parameters are estimated from training data using the observed spectra and the theoretical spectra of their curated correct identifications. The estimated parameters then are applied to test data for scoring the candidates. Because most informative peaks have high intensities, we estimate the observed intensities using the histogram density estimator, as it describes the density of right tails without being restricted by the shape constraints of the parametric approach. As a wider distance constraint includes more potential emission pairs, here we used 2Da as the distance for putative matches when initializing the fitting procedure in both training and test data.

Parameter estimation

The parameters estimated from the training data are summarized in Table 2.5. To show the effect of training procedure, the distance between locations of emission pairs are plotted at the start and the end of the training procedure (Figure 2.6).

At the start of the training procedure, the distances between putative matched pairs (Figure 2.6 a1, a2) spread between 0-2 Da and show a clear mixture pattern. For example, the distances clutter at 0, ± 1 and ± 2 for charge 1+ spectra, and also for charge 2+, though less clear. This is likely due to the matches between isotopic peaks, which differ in mass by integer values, e.g. 1, 2, etc. (Charge 2+ spectra have less clear cluttered pattern, because it contains fragments with charge 2, whose location differences will be (mass difference)/2 and may not be integers.)

Table 2.5: Parameters estimated from the training set of the curated ISB data using our model with both locations and intensities. Pool size shows the total number of peaks that are labeled as being matched at the start (s) and the end (e) of iteration.

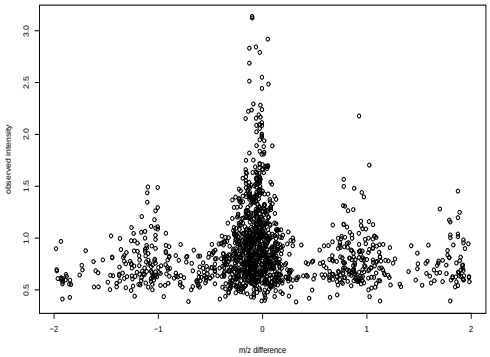
	$\log(L_1)$	pool(s - e)	σ	μ	β	tail bin	tail density (matched)	tail density (unmatched)
charge 1+	-8873.00	1437 - 1205	0.42	-1.12	2.91	[1.68, 3.14]	0.030	0.0067
charge 2+	-11301.96	2422 - 981	0.16	-4.93	4.65	[1.04, 2.91]	0.200	0.0150

After the training procedure, the distances between matched pairs are much more concentrated, though charge 1+ spectra still have notable mixture patterns. This shows that the training procedure removes most of the distant matches, which indicates the distance seems to have a dominant effect on the determination of the statuses. For matches with similar distances, those with lower intensities are more likely to be removed, as shown at the boundaries in Figure 2.6 b1, b2. The distributions of observed intensities are better separated after the training procedure (Figure 2.7).

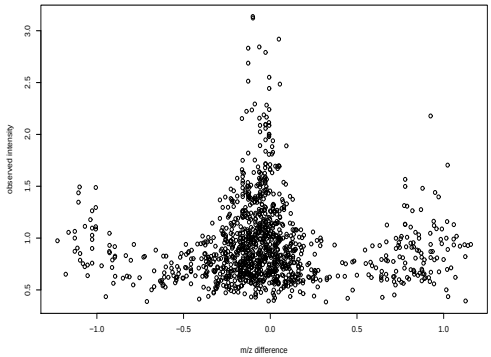
Correctness of identification

As this dataset is hand-curated, we call an identification correct, if the peptide candidate that is assigned the highest score by our method is the sequence chosen by SEQUEST and confirmed by hand curation. Mass spectrometry does not distinguish the amino acids {I, L}, because I and L have identical mass. It is also hard to distinguish {E, K, Q}, because the masses of K and Q are very close (difference < 0.1), and Q and E can convert to each other in liquid samples. Because of this, we count the sequences as being correct if they differ from the hand curation only by the exchange of the amino acids in these sets. They are referred as indistinguishable peptides in the rest of text.

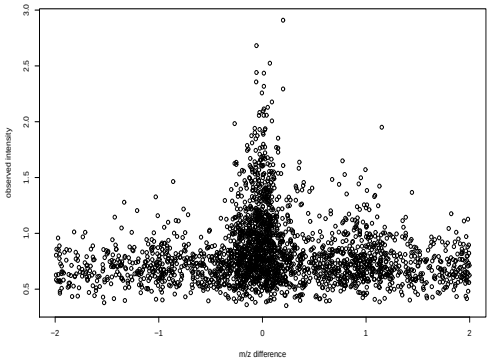
The correct identification rate is summarized in Table 2.6. The results show that incorporation of peak intensities improves the correct identification rate, though peak locations contain majority of information for scoring.



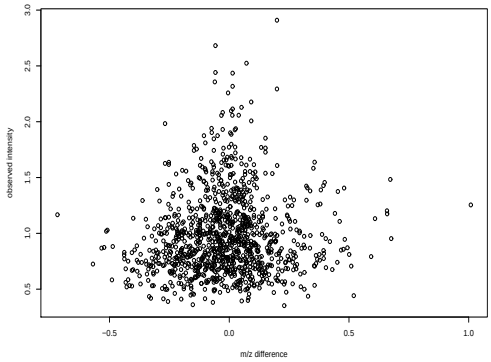
a1. Charge 1+, Before training



b1. Charge 1+, After training



a2. Charge 2+, Before training

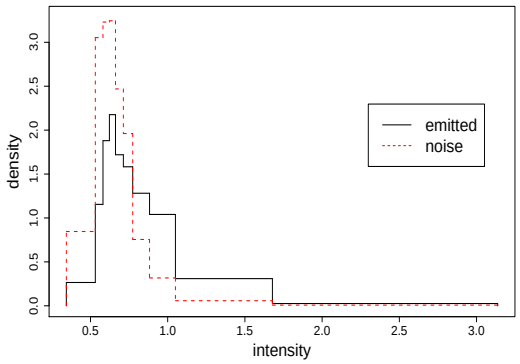


b2. Charge 2+, After training

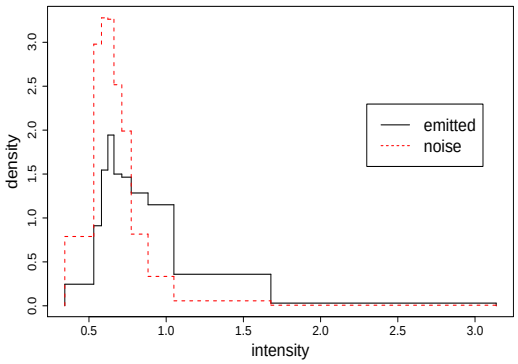
Figure 2.6: The emitted observed peaks before and after the training procedure. Plotted is the intensities of emitted observed peaks and the distances to their corresponding theoretical peaks. At the start of the training, the emission pairs are defined as peaks located less than 2Da apart.

Table 2.6: Correct identification rate on the curated ISB dataset.

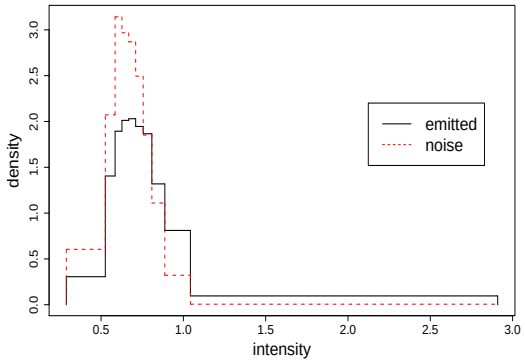
	With intensity		No intensity	
	test	train	test	train
Charge 1+	93.3	94.0	85.3	84.0
Charge 2+	96.8	100.0	90.3	88.0



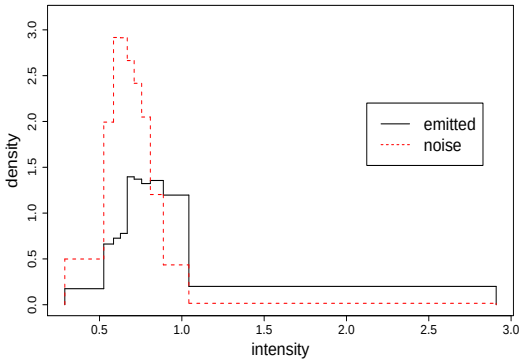
a1. charge 1+, before training



b1. charge 1+, after training

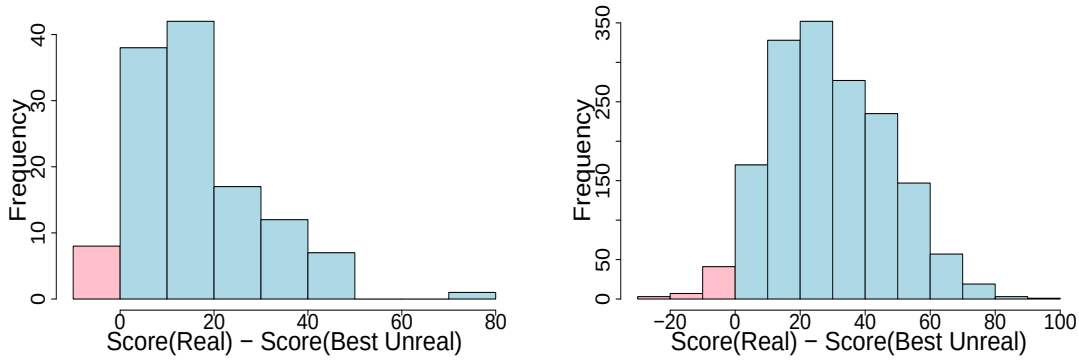


a2. charge 2+, before training



b2. charge 2+, after training

Figure 2.7: Distributions of observed intensities before and after the training procedure. Emission statuses are estimated in the training procedure, where intensities are estimated using histogram density estimators.



a. Charge 1+ ion

b. Charge 2+ ion

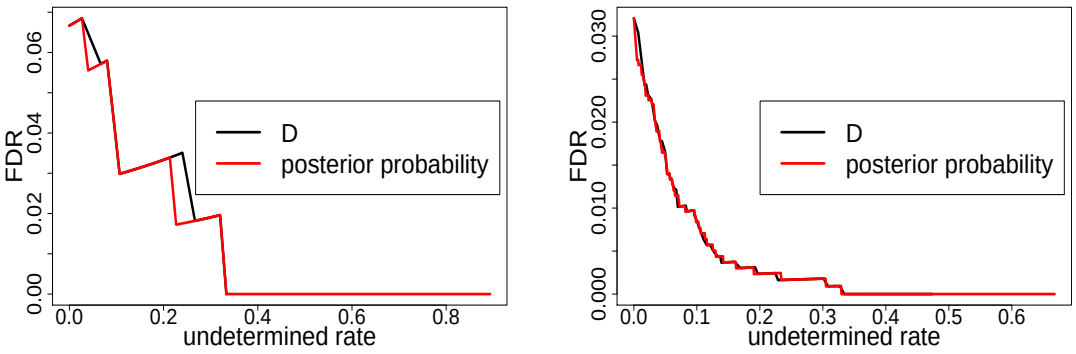
Figure 2.8: Separation of scores between real and top-ranked unreal sequences. Plotted is the histogram of $\log L_1^{\text{Real}} - \log L_1^{\text{Best Unreal}}$. Blue: correct identification; Pink: incorrect identification.

Separation of scores between real and top-ranked unreal sequences

It is of interest to know how well the scores can separate the real sequences from the incorrect ones. We plotted the difference between the scores of the real sequences and the top-ranked incorrect sequences on this dataset (Figure 2.8). As shown, our score provides a big separation between the score assigned to the real sequence and those to the incorrect ones, when identifications are correct (positive region in Figure 2.8). When identifications are incorrect (negative region in Figure 2.8), the scores assigned to the real sequences are close to the scores of the top-ranked incorrect sequences, which suggests mistakes of our method usually occur when the theoretical spectra of the real and incorrect sequences are similar.

Uncertainty of identification

We then assess the uncertainty of identification using both distinguishability and posterior probabilities (Figure 2.9) on this dataset. For each spectrum, two actions can be made : either *accept* or *refuse to accept* the top candidate as the identification depending on how



a. Charge 1+ ion

b. Charge 2+ ion

Figure 2.9: Uncertainty of identification measured by false discovery rate and the undetermined rate on test data. D is distinguishability.

distinguishing it is in terms of the selected uncertainty measure. Then there is a tradeoff between the proportion of false identifications made in the calls (i.e. FDR (Benjamini and Hochberg, 1995)) and the "undetermined rate", which is the proportion of spectra whose top candidate is refused to be accepted as an identification. We may use these two quantities at each threshold of the uncertainty measure to describe the performance of an uncertainty measures. Figure 2.9 shows that distinguishability and posterior probabilities have similar performance for ions at both charge states.

Calibration of posterior probabilities

We hope to use the posterior probabilities estimated from our model as a way to assess the uncertainty of the calls. However, to justify this strategy, the model should ideally produce approximately calibrated probabilities.

To assess the calibration of posterior probabilities from our model, we consider two quantities: the posterior probabilities for the top-ranked sequences and the posterior probabilities for all the candidate sequences. As the top-ranked sequences are potential identifications, the first quantity reflects the calibration of the identifications; whereas, the second quantity

reflects the overall calibration. For both quantities, the observations are binned by the assigned probabilities. In each bin, the assigned probabilities are then compared with the proportion of identifications that are real sequences.

As shown in Figure 2.10, our method assigns reasonably calibrated posterior probabilities in this dataset.

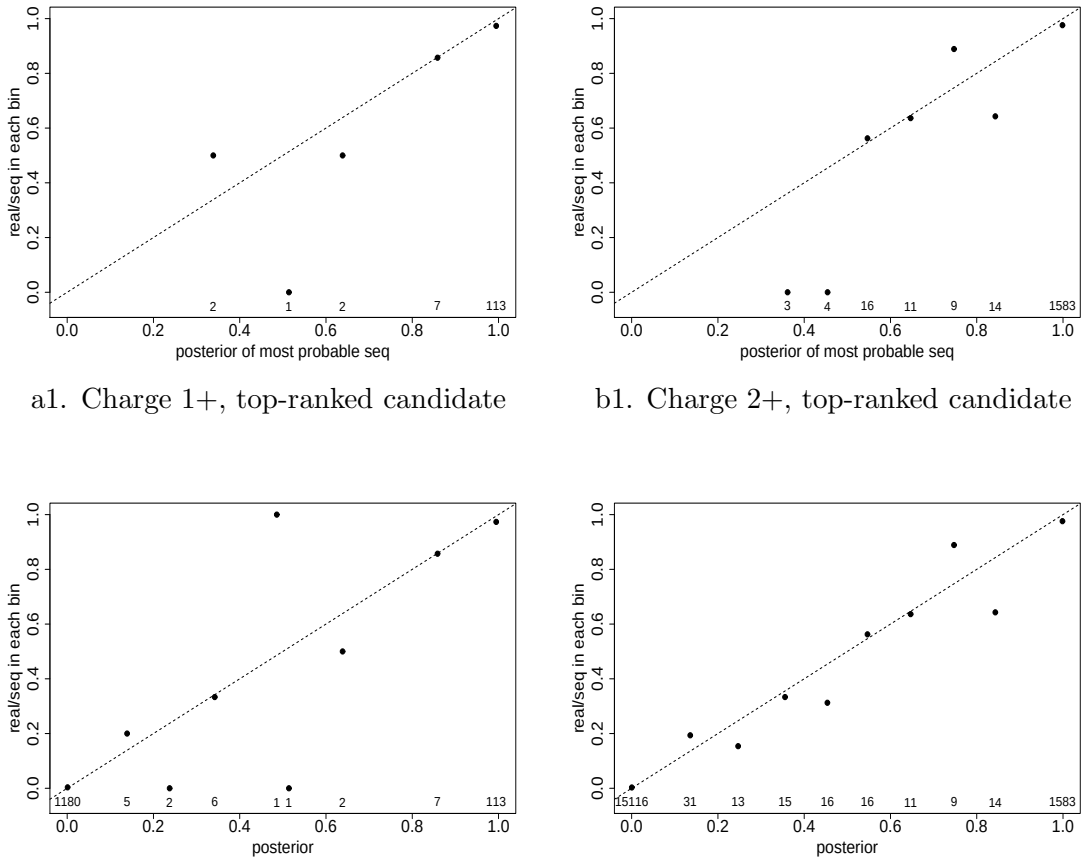
2.9.2 Comparison with SEQUEST

The analyses in Section 2.9.1 were carried out on the subset that SEQUEST made correct identifications. It is of interest to compare with SEQUEST directly.

As the ISB dataset does not provide the true sequences for the spectra outside of the curated subset, we adopted a widely used way to approximate the true sequence: an identification is declared to be correct if the identified sequence is a substring of a known protein in the mixture. Though this definition of correct identification is more relaxed than hand curation, the chances for a substring to be identified by chance is small when the database is large relative to the number of proteins in the mixture, which is the case here.

As mass spectra data typically contain a large number of spectra that are not able to be identified by any algorithms, we aim to focus our comparison on a subset that is able to be identified. To find such a subset, we selected the spectra whose real sequences are shortlisted (ranked within top 10) by SEQUEST. This subset contains 504 charge 1+ spectra and 3669 charge 2+ spectra. In this dataset, our method identified substantially more real sequences than SEQUEST for both charge 1+ (our: 89.9% vs SEQUEST: 68.1%) and charge 2+ spectra (our: 89.4% vs SEQUEST: 82.0%) (Table 2.7). In particular, our method successfully identified the majority (charge 1+: 79.7% and charge 2+: 63.5%) of the spectra that are misidentified by SEQUEST (Table 2.8).

The procedure described here can be viewed as a two-stage procedure, which first ranks spectra by SEQUEST, and then reranks the shortlisted candidates using our approach. The comparison above shows that it performs better than just using SEQUEST.



a1. Charge 1+, top-ranked candidate

b1. Charge 2+, top-ranked candidate

a2. Charge 1+, all candidates

b2. Charge 2+, all candidates

Figure 2.10: Calibration of posterior probabilities. Row 1: posterior probabilities of the highest-ranked candidates; Row 2: posterior probabilities of all the candidates. The dashed line marks perfect calibration. The number of observations at each point is marked at the bottom of the plot.

Table 2.7: Correct identification rate for the spectra whose real sequences are shortlisted (ranked within top 10) by SEQUEST in the ISB data. Exact: the top-ranked sequence is a substring of the protein mixture; Indistinguishable: the top-ranked sequence is indistinguishable to a substring of one of the proteins in the mixture by mass spectrometry.

	Our method			SEQUEST		
	exact	indisting.	total	exact	indisting.	total
charge 1+ (n=504)	434	19	453 (89.9%)	312	31	343 (68.1%)
charge 2+ (n=3669)	3219	62	3281 (89.4%)	2984	24	3008 (82.0%)

Table 2.8: Correct identification rate for the spectra whose real sequences are shortlisted (ranked within top 10) but not ranked top by SEQUEST in ISB data. Exact: the top-ranked sequence is a substring of the protein mixture; Indistinguishable: the top-ranked sequence is indistinguishable to a substring of the protein mixture by mass spectrometry.

	Our method			SEQUEST		
	exact	indisting.	total	exact	indisting.	total
charge 1+ (n=192)	138	15	153 (79.7%)	0	31	31 (16.4%)
charge 2+ (n=685)	373	62	435 (63.5%)	0	24	24 (3.5%)

2.10 Discussion

We have proposed a likelihood-based scoring algorithm for peptide identification using database search. Our algorithm is based on a generative model, which attempts to measure the likelihood that the observed spectrum arises from the theoretical spectrum of each candidate sequence. By explicitly modeling the noise structure, our probability model takes account of multiple sources of noise in the data, e.g. variable peak intensities and errors in peak locations. This attribute enables us to make use of the information on the sophisticated theoretical spectra, such as peak intensities. To my knowledge, our method is the only identification algorithm that scores the candidate sequences using sophisticated theoretical spectra.

Our results demonstrate that incorporating peak intensities improves the accuracy of peptide identification, in both the comparison with our model with only peak locations and the comparison with SEQUEST on the shortlisted candidates in ISB benchmark data. The comparison with SEQUEST shows that: our method has a higher correct identification rate than SEQUEST for the spectra whose real sequences are shortlisted by SEQUEST. In particular, our method identifies the majority of the spectra whose real sequences are shortlisted by SEQUEST but not identified correctly. We attribute the improved performance to the modeling of the noise structure on the spectra in our model. In doing so, our method is able to extract some of the more subtle signals which other methods miss, and ultimately improves the accuracy of peptide identification.

We provide two ways to assess identification uncertainty for each spectrum for our likelihood-based score. One is based on the log likelihood ratio, which describes the distinguishability of the top-scored sequence the next best sequence. The other is the posterior probability that a sequence generates the observed spectrum. These measures allow one to determine the uncertainty of identification with respect to the candidates of the given observed spectrum, instead of the identification scores of a pool of observed spectra. They are especially useful when the uncertainty assessment based on a pool of spectra is not suited, for example, small spectra sets or when scores are confounded with peptide length. These

measurements also are useful for flagging the spectra with multiple high-scored peptides, such as those peaks are generated from multiple peptides.

The assessment on the curated ISB data shows that the posterior probabilities are well calibrated and the two measures achieve similar performance. Though the two measures have similar performance on this dataset, we expect the posterior probability will perform better on data with longer candidate lists.

Due to limitations of computational efficiency in the generation of theoretical spectra and our current scoring procedure, we tested our approach on a short candidate list, instead of all the candidates obtained from a database. Because the candidates we tested are the ones shortlisted by SEQUEST, they are supposed to be the ones hardest to distinguish from the real one among all the candidates. In principle, the additional lower-ranked candidates from the protein database should not impact much on the results from our current assessment. To check this, we tested our algorithm on a technical replicate of the hand-curated dataset with 500 candidates shortlisted by SEQUEST (kindly provided by Jimmy Eng). In only 3% charge 1+ and 7% charge 2+ spectra, candidates ranked lower than 10 were selected by our method. This indicates that the correct identification rates only have a small decrease (decrease by at most 3% for charge 1+ and 7% for charge 2+).

Our peptide identification method takes a supervised approach, where the scoring parameters that are characteristic of the charge states and the instrumental condition are learned in the training procedure. Though supervised approach requires a training set, which seems to be less desirable because a training set with known truth is hard to obtain in realistic datasets, this type of scoring schemes are also used in other programs, e.g. PepHMM (Wan et al., 2006). We also would like to point out that our procedure uses a fairly small training set, comparing with other supervised scoring algorithms. For example, for the experiments in Section 2.9, our method was trained on 50 spectra for each charge state, with a ratio of testing to training of 1.5:1 for charge 1+ and 31.8:1 for charge 2+ spectra; while PepHMM achieves a similar performance with a ratio of 1:4 (i.e. 5-fold cross-validation). This seems to suggest that our method does not heavily rely on the training

data. Thus, it may be possible to learn the scoring parameters from a generic training set, for example, a calibration dataset collected at the same experimental condition, which is commonly available in a lab.

There are several possible enhancements to our current method. Currently, we model the distance between two peaks in the emission pairs as a normal distribution. However, as shown in Figure 2.6, the distance has multiple modes, for example, centered around $0, \pm 1, \pm 2$ for charge 1+ ions, which may be due to matches between isotopic ions. One possibility is to just model the middle component. Actually, this modeling strategy is supported by the results from our experiments with distance constraints of 0.5Da and 1Da on the curated dataset. After the training results with these distance constraints, only the middle component is kept (results not shown), and the correct identification rate is higher when the parameters from training are applied to the testing data (Table 2.9). Or alternatively, we can use a mixture model for the distribution of this distance. Another possible improvement is to approximate the generative likelihood by summing the complete likelihood over multiple probable configurations. The likelihood resulted from this approximation is numerically closer to the generative likelihood than the complete likelihood at the most probable configuration, and will lead to rankings that are more consistent to the generative likelihood. In addition, we may model σ as a function of locations to take account of the varied local density of peaks.

As computational efficiency is important for high-throughput peptide identification, our method needs to be sped up for practical use. There are several possible ways. For example, we may update μ once in each round of configuration update, rather than once at each site. We may use dynamic programming to choose the most probable configuration for each site, instead of an exhaustive search. For practical use, our R program will need to be converted into C or C++.

In this work, a sophisticated theoretical spectra is used to illustrate our statistical method. However, it is worth noting that our method actually is independent of the theoretical spectra that are used, and can be applied to the coarse theoretical spectra as well.

Table 2.9: Parameters estimated from curated ISB data using our model with both locations and intensities. Pool size shows the total number of peaks that are labeled as being matched in the training set at the start (s) and the end (e) of iteration. The procedure was carried out using different starting emission definitions (0.5Da and 1Da).

	$\log(L_1)$	pool(s - e)	σ	μ	β	tail bin	tail density (matched)	tail density (unmatched)
charge 1+								
Def=0.5	-8725.90	965-909	0.15	-1.78	2.69	[1.65, 3.14]	0.0415	0.0069
Def=1	-8726.16	1177-916	0.15	-1.78	2.72	[1.65, 3.14]	0.0413	0.0066
charge 2+								
Def=0.5	-11301.79	1340 - 981	0.16	-4.82	4.49	[1.22, 2.91]	0.128	0.0054
Def=1	-11165.07	1815 - 1003	0.17	-4.82	4.57	[1.14, 2.91]	0.149	0.0082

In addition, though our method is demonstrated using only charge 1+ and 2+ spectra in our experiments, it indeed can also be applied to spectra with higher charge states and peptides with post-translational modification, as long as their theoretical spectra are available. Since our method explicitly models the noise structure in the data, it is expected to have advantages over the methods that lack the sophistication in handling the complicated noise structure, when working on spectra with fine details (e.g. observed spectra). One example of this kind is spectral library search, where the observed spectra are identified by matching to the previously annotated observed spectra that are collected in the library. Our method can be applied by simply replacing the theoretical spectra by the annotated observed spectra.

Appendix A

On a training set S, the objective function of iterative training for the approximation model is the following likelihood function:

$$L_1(\mu,\beta,\sigma,\Psi_0,\Psi_1) = \prod_{s\in S} \max_{\mathbf{e}_s^t} \Big[(m_s-k_s)!\big(\frac{1}{r_s}\big)^{m_s-k_s} \prod_{\{j:e_{s_j}^o=0\}} f_0(y_{s_j}^o)\big[\prod_{\{j:e_{s_j}^o>0\}} N(d_{s_j};0,\sigma^2) f_1(y_{s_j}^o)\big] \tag{2.10.1}$$

$$\prod_{\{i:e_{s_i}^t>0\}} p_{s_i} \prod_{\{i:e_{s_i}^t=0\}} (1-p_{s_i})\Big]$$

The likelihood can be factorized into independent terms to ease optimization. Suppose ${\mathbf{e_s}}^t$ is the most probable configuration, if we omit max for notational simplicity, the log-likelihood is

$$\begin{aligned} l' &= \sum_{s\in S}\Big[(\log((m_s-k_s)!) - (m_s-k_s)\log r_s) + \sum_{\{j:e_{s_j}^o>0\}} \log N(d_{s_j};0,\sigma^2) \qquad (2.10.2)\\ &+ \big[\sum_{\{j:e_{s_j}^o=0\}} \log f_0(y_{s_j}^o) + \sum_{\{j:e_{s_j}^o>0\}} \log f_1(y_{s_j}^o)\big] + \big[\sum_{\{i:e_{s_i}^t>0\}} \log p_{s_i} + \sum_{\{i:e_{s_i}^t=0\}} \log(1-p_{s_i})\big]\Big] \\ &= \sum_{s\in S}\big[\sum_{\{i:e_{s_i}^t>0\}} \log p_{s_i} + \sum_{\{i:e_{s_i}^t=0\}} \log(1-p_{s_i})\big] + \sum_{s\in S}\sum_{\{j:e_{s_j}^o>0\}} \log N(d_{s_j};0,\sigma^2) \\ &\sum_{s\in S}\big[\sum_{\{j:e_{s_j}^o=0\}} \log f_0(y_{s_j}^o) + \sum_{\{j:e_{s_j}^o>0\}} \log f_1(y_{s_j}^o)\big] + \sum_{s\in S}(\log((m_s-k_s)!) - (m_s-k_s)\log r_s) \end{aligned}$$

The first three terms above are the objective functions for estimating β,σ and (Ψ_0,Ψ_1), respectively. This suggests the iterative procedure in Table 2.2, which alternates the update of the three "pooled" parameters and the search of the most probable configuration. As μ_s depends on individual spectra, we optimize it along with the search of the most probable configuration for individual spectra. The values of μ_s's are fixed in the estimation of β,σ and Ψ_0, Ψ_1.

Chapter 3

PROTEIN IDENTIFICATION USING PEPTIDE IDENTIFICATIONS

3.1 Introduction

The ultimate goal of most proteomic experiments is not the identification of peptides, but the identification of the proteins present in the sample (Nesvizhskii and Aebersold, 2005). This chapter will consider how to determine protein identities based on the peptides identified from mass spectrometry by peptide identification algorithms. We will focus on peptides identified by the database searching algorithms, as it is the most widely used approach for large-scale peptide identification. However, our framework is general enough to work with other types of peptide identification algorithms, such as *de novo* identification, with little adaptation.

To understand the problem, it is helpful to think of peptide identification algorithms as providing two pieces of information for each experimental spectrum: an identified peptide sequence that is most likely to generate the spectrum among all the candidate sequences in the database, and a score that summarizes the strength of the evidence for the identified sequence being correct. In general, this score could be multi-dimensional, and may incorporate several types of information, including the goodness of the match between the observed spectrum and theoretical predictions. However, for simplicity it may help to initially think of this score as a single scalar.

Since each peptide sequence, which may be identified by one or more spectra, maps to one or more proteins, one may assemble a list of putative proteins from the identified putative peptides using this mapping between peptides and proteins (Figure 3.1a). However, because peptide identification algorithms are known to produce a significant number (80-90%, personal communication with Murray Hackett and Michael MacCoss) of incorrect

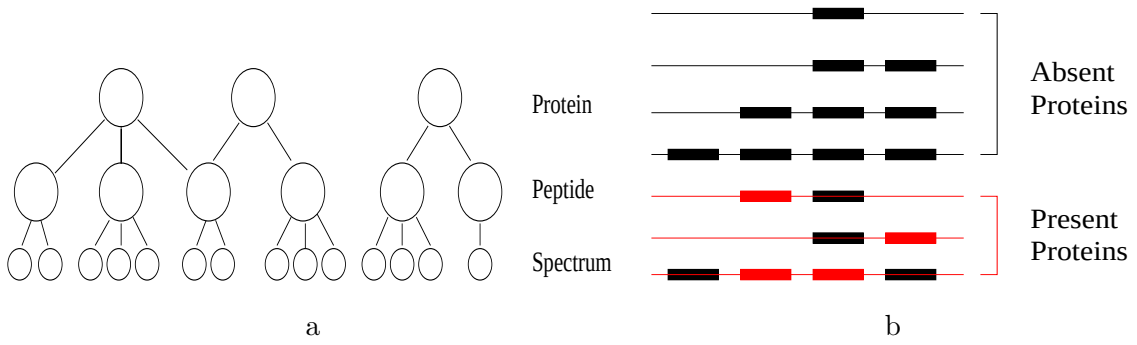


Figure 3.1: Putative peptide identification and reconstructed proteins. a: Mappings between spectra, peptides and proteins. b: Reconstructed proteins from putative peptide identifications. Black: incorrect peptide identifications; Red: correct peptide identifications

peptide assignments (if the top-scored candidate is assumed to be correct without further filtering) (Keller, 2002; Nesvizhskii and Aebersold, 2004), many proteins constructed in this way are actually not present in the sample. Thus the overall strength of evidence for protein being present needs to be assessed based on the scores of the putatively identified peptides that map to them.

Currently, the common practice of protein identifications is to divide the task into two steps: first compute a probability for each peptides being correctly identified, separately for each peptide, based on the peptide identification scores (e.g. PeptideProphet (Keller et al., 2002), Percolator (Kall et al., 2007)), then decide which proteins are present by combining these probabilities in some way. For example, protein statuses may be estimated either by simple *ad hoc* thresholding rules, e.g. selecting proteins with two or more "high-scored" peptides, or more sophisticated, though still *ad hoc* approaches (e.g. ProteinProphet (Nesvizhskii et al., 2003), Prot_Prob (Sadygov et al. (2004)) and EBP (Price (2007))). Though this two-stage approach is widely used, it ignores the feedback between protein statuses and peptide statuses resulting from the nested structure between peptides and their parent proteins. In addition, because their computations of protein probabilities are *ad hoc*, these methods do not assign well-calibrated probabilities to proteins.

In this chapter, we propose an unsupervised approach to protein identification based on a nested mixture model. This approach incorporates the feedback between peptide statuses and protein statuses by simultaneously identifying which proteins are present and which peptides are correctly identified. As a result, it provides, in principle, properly-calibrated probabilities for each protein being present in the sample, and more accurate peptide identifications.

The organization of the chapter is as follows. Section 3.2 describes our modeling approach. Section 3.3 describes the widely-used programs in peptide validation (PeptideProphet) and protein identification (ProteinProphet) that our method is compared with. In Section 3.4, we use simulated data to illustrate parameter estimation and model fitting. In Section 3.5, we compare our method with PeptideProphet and ProteinProphet on data from yeast. We present the justification of model choices and other models we have attempted in Section 3.6. In section 3.7, we conclude and suggest future enhancement.

3.2 Methods

3.2.1 A nested mixture model

Since putative proteins are assembled from putative peptide identifications, each protein is described by information on its identified peptides, namely, the number of identified peptides on the protein and the scores of these peptide identifications. In general, correct peptide identifications have higher scores than incorrect ones, and proteins that are present tend to have more high-scored peptide identifications than the ones that are not present. Our goal is to use this information to determine which assembled proteins are present in the sample and which peptides are correctly identified.

Suppose N proteins are constructed from putative peptides identified from a set of mass spectra. Let n_k denote the number of identified peptides on protein k, and $X_k = (X_{k,1}, \ldots, X_{k,n_k})$ denote the scores associated with peptides on that protein, where $X_{k,i}$ is the score of peptide identification for the ith peptide. We use T_k to indicate whether a protein k is present ($T_k = 1$) or absent ($T_k = 0$) in the sample, and $P_{k,i}$ to indicate whether

a peptide i on the protein k is correctly ($P_{k,i} = 1$) or incorrectly ($P_{k,i} = 0$) identified. For simplicity, here we ignore peptide sharing or "degeneracy" (Nesvizhskii et al., 2003), which refers to the situation that one peptide maps to *multiple* proteins, in that we treat observations on different proteins as independent even if they share a peptide in common. The practical consequence of this is the shared peptides are overcounted. We view extension of our method to deal properly with degeneracy as an important aim for future work. We refer to this treatment of degeneracy as the "nondegeneracy assumption" for the rest of text.

Because peptides are substrings of proteins, under the nondegeneracy assumption, once a peptide is correctly identified, its parent protein is present in the sample. Consequently, an absent protein contains only incorrectly identified peptides, whereas a present protein typically contains at least one correctly identified peptides and maybe some incorrectly identified peptides (Figure 3.1b).

If we assume that (1) the peptide scores are conditionally independent given the status of proteins and that (2) all the present proteins have the same proportion of correct peptide identifications, then the densities of absent proteins (g_0) and present proteins (g_1), for a given number of peptide identification n_k, are as follows:

$$g_0(\mathbf{x_k}) \equiv P(\mathbf{X_k} \mid n_k, T_k = 0) = \prod_{i=1}^{n_k} f_0(x_{k,i}) \tag{3.2.1}$$

$$g_1(\mathbf{x_k}) \equiv P(\mathbf{X_k} \mid n_k, T_k = 1) = \prod_{i=1}^{n_k} [\pi_1 f_0(x_{k,i}) + (1 - \pi_1) f_1(x_{k,i})], \tag{3.2.2}$$

where π_1 is the proportion of incorrect peptides on the proteins that are present; $f_0(x_{k,i})$ and $f_1(x_{k,i})$ represent the score distributions of the incorrect and correct peptide identification, respectively. Here, g_1 takes a format of a mixture model McLachlan and Peel (2000), which is the lower-level of our nested mixture model. The two assumptions will be discussed further in later sections.

As assembled proteins are a mixture of present and absent proteins, the joint probability

for protein k is as follows:

$$P(\mathbf{X_k}, n_k) = \pi_0^* g_0(\mathbf{x_k}) h_0(n_k) + \pi_1^* g_1(\mathbf{x_k}) h_1(n_k), \tag{3.2.3}$$

where π_0^* and π_1^* are the proportions of absent and present proteins, respectively, with $\pi_0^* + \pi_1^* = 1$; h_0 and h_1 are the distributions of the number of uniquely identified peptides on the absent and present proteins, respectively. Here, f_j, g_j and h_j $(j = 0, 1)$ are unknown parameters that need to be estimated from data. The modeling details are provided in Section 3.2.3 for f_j and in Section 3.2.5 for h_j. This is the upper-level of the nested mixture model.

Let ψ represent all the parameters involved. Under the nondegeneracy assumption, all the proteins are independent, and the mixture likelihood is

$$L(\psi) = \prod_{k=1}^{N} [\pi_0^* g_0(\mathbf{x_k}) h_0(n_k) + \pi_1^* g_1(\mathbf{x_k}) h_1(n_k)]. \tag{3.2.4}$$

For notational simplicity, we suppress ψ when no confusion arises.

3.2.2 Latent variable representation

In this problem, the inference of the statuses of peptides and proteins is of primary interest. Thus, it is convenient to introduce the latent variable formulation of the model. Let T_k and $P_{k,i}$ be the allocation variables for protein k and peptide i on protein k, respectively. As peptides are substrings of proteins, the two levels of latent variables form a nested structure. At the protein level, the model (3.2.4) can be written in terms of the latent variable $T_1, \ldots, T_N$ with probability mass function

$$Pr(T_k = j) = \pi_j^* \quad (k = 1, \ldots, N; j = 0, 1) \tag{3.2.5}$$

Conditioning on T's, protein scores $\mathbf{x_1}, \ldots, \mathbf{x_N}$ (or more explicitly $(\mathbf{x_1}, n_1), \ldots, (\mathbf{x_N}, n_N)$) are assumed to be independent observations from the densities

$$p(\mathbf{x_k}, n_k \mid T_k = j) = p(\mathbf{x_k} \mid n_k, T_k = j) p(n_k \mid T_k = j) = g_j(\mathbf{x_k}) h_j(n_k). \tag{3.2.6}$$

At the peptide level, all the peptide identifications on an absent protein are incorrect, i.e.

$$Pr(P_{k,i} = 0 \mid T_k = 0) = 1. \tag{3.2.7}$$

When proteins are present, the peptide status $P_{k,i}$ is assumed to be independent and identically distributed with probability mass function

$$Pr(P_{k,i} = 0 \mid T_k = 1) = \pi_1 \quad (k = 1, \ldots, N; i = 1, \ldots, n_k). \tag{3.2.8}$$

Conditioning on $P_{k,i}$, peptide scores $x_{k,i}$ are assumed to be independent observations from the densities

$$p(x_{k,i} \mid P_{k,i} = j) = f_j(x_{k,i}) \tag{3.2.9}$$

The classification probabilities for proteins are

$$P(T_k = j \mid \mathbf{x_k}) = \frac{\pi_j^* g_j(\mathbf{x_k}) h_j(n_k)}{\sum_{j=0,1} \pi_j^* g_j(\mathbf{x_k}) h_j(n_k)} \tag{3.2.10}$$

and the classification probabilities for peptides on the proteins that are present are

$$P(P_{k,i} = 1 \mid x_{k,i}, T_k = 1) = \frac{\pi_1 f_1(x_{k,i})}{\pi_1 f_0(x_{k,i}) + (1 - \pi_1) f_1(x_{k,i})} \tag{3.2.11}$$

Because an absent protein only contains incorrect peptide identifications, i.e. $P(P_{k,i} = 1 \mid \mathbf{x_k}, T_k = 0) = 0$, then

$$\begin{aligned} & P(P_{k,i} = 1 \mid \mathbf{x_k}) \qquad (3.2.12) \\ &= P(P_{k,i} = 1 \mid \mathbf{x_k}, T_k = 0) P(T_k = 0 \mid \mathbf{x_k}) + P(P_{k,i} = 1 \mid \mathbf{x_k}, T_k = 1) P(T_k = 1 \mid \mathbf{x_k}) \\ &= P(P_{k,i} = 1 \mid \mathbf{x_k}, T_k = 1) P(T_k = 1 \mid \mathbf{x_k}) \end{aligned}$$

This formulation shows that the peptide status is affected by the status of its parent protein.

The above is essentially a model-based clustering method, which simultaneously estimates the correctness of peptide identification and infers the protein evidence by jointly clustering the statuses of peptides and proteins in a nested structure. The idea of validating peptide identification using a mixture model seems to originate with PeptideProphet (Keller et al., 2002), where peptide statuses are estimated only with information on the peptides, without incorporating information from proteins; whereas, our approach incorporates the protein-level feedback.

3.2.3 Modeling distributions of peptide identification scores

In general, the scores of peptide identification are multi-dimensional, and may incorporate several types of information, including the goodness of the match between the observed spectrum and theoretical predictions. The selection of predictive features in peptide identification scores is critical for determining the correctness of peptide identifications (Kall et al., 2007; Keller et al., 2002; Sadygov and Yates, 2003). However, since it is not the focus here, for convenience of comparison, we summarize multi-dimensional scores using a summary score as in PeptideProphet, although our model can take any scores in the scalar format.

We model the distribution of the summary score as the following:

$$X_{k,i} \mid P_{k,i} = 0 \sim N(\mu, \sigma^2)$$
$$X_{k,i} \mid P_{k,i} = 1 \sim Gamma(\alpha, \beta, \gamma),$$

where α, β and γ are the shape parameter, the scale parameter and the shift of the Gamma distribution, and μ and σ^2 are the mean and variance of the normal distribution. The choice of the distributions is made based on the shapes of the empirical observations (Figure 3.2a), the density ratio at the tails of the distributions, and the goodness-of-fit between the distributions and the data, e.g. BIC (Schwarz, 1978). In particular, to assign peptide labels properly in the mixture model, it requires $f_0/f_1 > 1$ for the left tail of f_0 and $f_1/f_0 > 1$ for the right tail of f_1. Note that our distribution choice is different from the ones in PeptideProphet, which models f_0 as Gamma and f_1 as Normal, because the distributions chosen by PeptideProphet do not satisfy the requirement of f_0/f_1 mentioned above and can pathologically assign observations at the lower end into the higher component. The distributions selected fit our data well. However, one may need to adapt to other distributions depending on the empirical data. More discussion on model choices can be found in Section 3.6.1.

3.2.4 Incorporating additional features of peptides

In addition to identification scores, ancillary information on identified peptide sequences, such as how well the identified peptides satisfy the *a priori* experimental conditions, has been reported to be useful for validation of peptide identifications (Kall et al., 2007; Keller et al., 2002; Choi and Nesvizhskii, 2008b). Here we consider the number of tryptic termini (NTT) and the number of missing cleavage (NMC), because of their reported predictive importance (Kall et al., 2007; Choi and Nesvizhskii, 2008b). Because NTT $\in \{0, 1, 2\}$ (Figure 3.2b), we model it using a multinomial distribution. We discritize NMC, which usually ranges from 0 to 10, into states (0, 1 and ≥ 2) (Figure 3.2c), and also model it as a multinomial distribution. These treatments are similar to those in PeptideProphet.

Empirical observations in literature show that peptide identification scores and features on peptide sequences are conditionally independent given the status of peptide identification (Keller et al., 2002; Choi and Nesvizhskii, 2008b). Thus we may incorporate the ancillary information by replacing $f_j(X_{k,i})$ in ((3.2.1) and (3.2.2)) with $f_j(X^S_{k,i}) f_j^{NTT}(X^{NTT}_{k,i}) f_j^{NMC}(X^{NMC}_{k,i})$ $(j = 0, 1)$, where $x^S_{k,i}$ is the (summary of) identification score modeled as in Section 3.2.3, $x^{nmc}_{k,i}$ is the number of missed cleavage and $x^{ntt}_{k,i}$ is the number of tryptic termini.

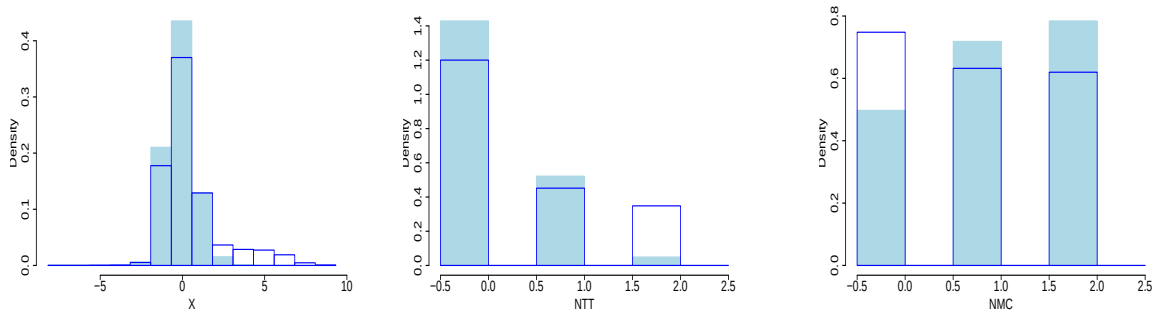
3.2.5 Incorporating protein length

It is known that long proteins tend to have more identified peptides than short proteins because of their potentials to generate more peptides in both experimental procedure and to be randomly matched in database search (Figure 3.3). We incorporate this feature by allowing the distribution of n_k to depend on the protein length.

It is known that the rate of incorrect peptide identification in a fixed protein length is roughly uniform across all the proteins in the database. For absent proteins, which only have incorrect peptide identifications, we model

$$n_k \mid l_k \sim Poisson(c_0 l_k),$$

where c_0 represents the average number of incorrect peptide identifications in a unit pro-



a. Summary score X b. Number of tryptic termini c. Number of missing cleavages

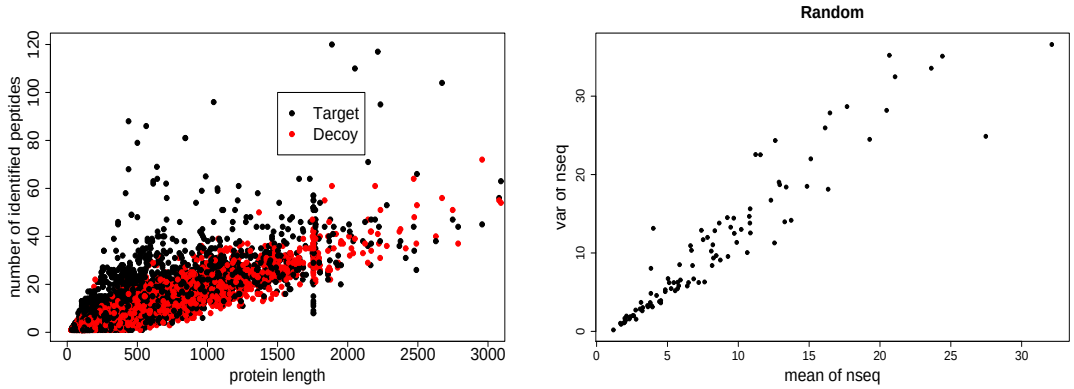
Figure 3.2: Empirical distribution of features from peptide identification in a yeast data. The data is obtained by searching against a database concatenated from a real database, which is a mixture of correct and incorrect identifications, and a decoy database, which approximates the distribution of incorrect identifications. Border histogram: features of peptides identified from the real database, which is a mixture of correct and incorrect identifications. Solid histogram: features of peptides identified from the decoy database.

tein length and is assumed to be constant for all the absent proteins. The mean-variance relationship of n_k for absent proteins in a real dataset (Figure 3.3b) confirms that Poisson model is a reasonable fit.

Because the number of correct identifications depends on many factors additional to protein lengths, the relationship between the number of identified peptides and the protein length for present proteins, which have both correct and incorrect identifications, are more heterogeneous than absent proteins (Figure 3.3). For convenience, we also model it with a Poisson distribution, $n_k \mid l_k \sim Poisson(c_1 l_k)$, where c_1 is constant for all the present proteins.

Note that because constructed proteins are assembled from one or more identified peptides (i.e. $n_k > 0$), the Poisson distributions should be truncated at 0, i.e.

$$h_j(n_k \mid l_k) = \frac{\exp(-c_j l_k)(c_j l_k)^{n_k}}{n_k!(1 - \exp(-c_j l_k))} \quad (j = 0, 1). \tag{3.2.13}$$



a b

Figure 3.3: Relationship between the number of peptide hits and protein length in a yeast data. a. The relationship between the number of peptide hits and protein length. Red dots are decoy proteins, which approximate absent proteins; black dots are target proteins, which contains both present proteins and absent proteins. b. Verification of the Poisson model for absent proteins, approximated by decoy proteins, by mean-variance relationship. Proteins are binned by length with each bin containing 1% of data. Mean and variance of number of sequences are calculated for the observations in each bin.

3.2.6 Parameter estimation and initialization

We use an expectation-maximization (EM) algorithm (Dempster et al., 1977) to estimate the parameters in our model and infer the statuses of peptides and proteins (see appendix A for details), with the statuses of proteins (T_k) and the statuses of peptides ($P_{k,i}$) as latent variables. For protein k, the augmented data takes the form of $Y_k \equiv (\mathbf{X_k}, T_k, (P_{k,1}, \ldots, P_{k,n_k}))$. The EM algorithm is standard except that the nested structure requires the peptide status conditional on the status of its protein parent.

As with any EM algorithm, our algorithm will typically find a local maximum of the likelihood function. For realistic data sets, this likelihood surface will have many local maxima. To select a reasonable starting point in the real datasets, we initialize the parameters related to incorrect peptide identification ($f_0, f_0^{NTT}, f_0^{NMC}$ and c_0) using the estimation from the search results on a decoy database[1], which approximates the distribution of the incorrect identification, when it is available. In particular, for f_0, we initialize the shift $\gamma^{(0)} = \min_{k,i}(x_{k,i}) - \epsilon$, where ϵ is a small positive number to ensure $x_{k,i} - \gamma^{(0)} > 0$, and estimate α and β using the sample mean and sample variance. We initialize the parameters related to correct peptide identification ($f_1, f_1^{NTT}, f_1^{NMC}$) with the scores from the search results on the real database[2] . We initialize f_1^{NTT} and f_1^{NMC} using the peptide identifications that are scored above 90% percentile. As $c_1 > c_0$, we choose $c_1 = bc_0$, where b is a random number in $[1.5, 3]$. The starting values of π^* and π_1 are randomly chosen from $(0, 1)$. For each inference, we choose 10 random starting points as described above and report the results from the one that converges to the highest likelihood.

---

[1]A decoy database contains dummy sequences that are created by permuting the sequences in the real database. As these sequences are different from the real sequences, all the peptide identifications against this database should be incorrect. Then the distribution of the scores from the search against decoy database presumably approximates the distribution of incorrect identifications. This strategy is well-adopted in proteomics (Elias and Gugi, 2007).

[2]The peptide identifications obtained from the search against a real database contain both correct and incorrect identifications. Sometimes we refer a real database as a target database or a forward database in contrast with a decoy database.

3.3 Methods of Comparison

Here, we compare performance of our methods with the most widely used 2-stage methods for protein inference (ProteinProphet (Nesvizhskii et al., 2003)) and peptide validation (PeptideProphet (Keller et al., 2002)). The model underlying PeptideProphet bears some similarities to our model, being based on the idea of clustering identified peptides into correct and incorrect identifications using a mixture model, but the cluster membership is estimated only based on the information from individual peptides without feedbacks from their protein parents.

The key idea of ProteinProphet is to compute the probability that a protein is present as the probability that the protein contains at least one correctly identified peptides ($P_{prod}(T_k = 1) = 1 - \prod_i P(P_{k,i} = 1)$), which is called as the *product rule* in the rest of text. To combine information across peptides on the same protein, ProteinProphet first adjusts the peptide probabilities computed by PeptideProphet, with the expected number of other correctly identified peptides on the same protein in a heuristic fashion, then use the adjusted peptide probabilities for the product rule. When proteins have shared peptides, ProteinProphet uses a heuristic way to partition the contribution of each shared peptide to its protein parents.

The product rule above implicitly assumes that the correctness of peptides is independent for peptides on the same protein. Though its empirical results are reasonable, this assumption does not hold in practice. For example, all peptides on an absent protein are incorrectly identified, i.e. they are highly dependent. We will later explore the practical effect of this assumption by simulations.

3.4 Simulation studies

We first conduct simulation studies to evaluate the performance of our methods and compare with the competing 2-stage approaches. The simulation studies offer several advantages that are not available from real datasets. First, simulation studies provide the ground truth that is unknown in real datasets. (Truth is unknown even for the datasets generated from purified protein mixtures due to unknown impurities in the samples.) Second, simulation studies

allows us to focus on the behavior under the assumption of lack of degeneracy, as our model ignores this important but difficult part in real data. Third, by controlling the simulation conditions, we can gain insights into how the approaches of comparison work in particular experimental scenarios.

At the peptide level, our model is compared with the peptide probabilities computed by PeptideProphet and the adjusted peptide probabilities. At the protein level, our model is compared with three methods: the product rule with the two peptide probabilities above as inputs (called as *naive product rule* and *adjusted product rule*, respectively), and the rule that calls a protein present if it has two or more high-scored peptides (called as *two-peptide rule*). As the product rule is the basis of ProteinProphet, the comparison with the product rule, instead of ProteinProphet, not only provides convenience of implementation, but also allows us to focus on the fundamental differences between our method and ProteinProphet, without involving the complication of degeneracy handling and technical details (e.g. unpublished heuristic adjustments and interface to protein databases) in ProteinProphet. Note that the adjusted product rule is the one that resembles a simplified version of ProteinProphet.

Three simulations were carried out to address the following three aspects:

- performance comparison when the model for estimation is the model to generate data
- scenarios where the product rule fails
- sensitivity of our method to the assumption that the proportion of incorrect peptides is constant for the proteins that are present.

As PeptideProphet uses Gamma for f_0 and Normal for f_1, we followed this practice in the simulations. To generate realistic simulations, we first estimated parameters from a yeast dataset (Kall et al., 2007) using the model in section 3.2, except for this change of f_0 and f_1. The estimated parameters (Table 3.1) will be the basis of parameters for our simulations. For simplicity, in all simulations, we only simulated one identification score for each peptide and set identical ancillary features for all the peptides (NMC=0 and

Table 3.1: Simulation parameters and parameter estimation in the simulation from our proposed model. True: simulation parameters, which are estimated from a yeast data. Protein length is simulated with exp(1/500). Gamma distribution is represented as Gamma(shape, scale, shift). π_0 is the proportion of incorrect peptides on the proteins that are absent in the simulation. Full: our full model; Reduced: our reduced model. The likelihood of reduced model is not provided, as it is not comparable due to different number of parameters.

	π_0^*	c_0	c_1	π_0	π_1	f_0	f_1	$\log(L)$
True	0.88	0.018	0.033	1	0.58	$G(86.46, 0.093, -8.18)$	$N(3.63, 2.07^2)$	-36086.68
Full	0.87	0.018	0.032	-	0.58	$G(86.24, 0.093, -8.18)$	$N(3.57, 2.05^2)$	-36082.44
Reduced	0.86	-	-	-	0.58	$G(86.33, 0.093, -8.18)$	$N(3.56, 2.10^2)$	-

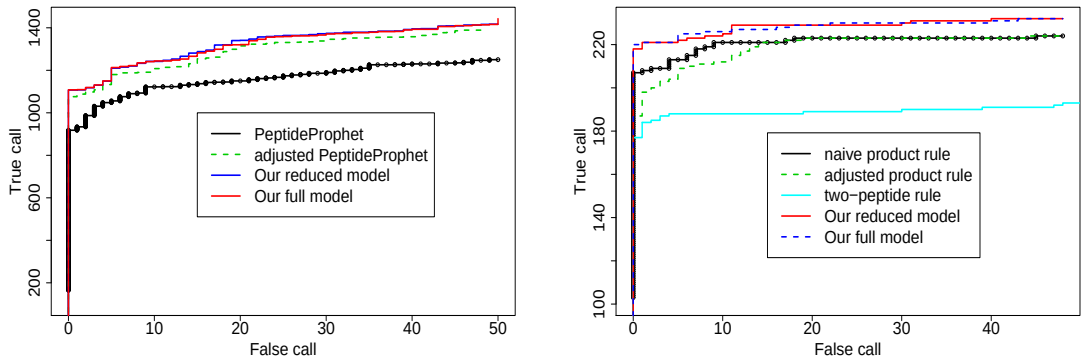
NTT=2). In all three simulations, 2000 proteins were simulated, and all the proteins that are present were ensured to have at least one correctly identified peptide. In each simulation, we estimated parameters using two models: one is (3.2.4), and the other is similar, except that we assumed $h_0 = h_1$. The former is referred as the "full model", and the latter as the "reduced model" in the rest of the text. The EM procedure was run from several random initializations close to simulation parameters. Convergence is deemed as achieved when the increment of log-likelihood is smaller than 0.001.

3.4.1 Simulation from our proposed model

In this simulation, we simulated data from the estimated parameters from the yeast dataset (Table 3.1) following (3.2.4). It consisted of 240 present proteins and 1760 absent proteins, where protein length $l_k \sim \exp(1/500)$. We estimated the parameters and the statuses of peptides and proteins using our models. The results show that parameters estimated from our models are close to the true parameters (Table 3.1).

Tradeoff between true calls and false calls

Because the true protein composition is unknown for realistic datasets, the performances of different methods can be compared by the tradeoff between the number of correct and incorrect calls made at various probability thresholds. As the practical interest is to get a



a. Peptide identification

b. Protein identification

Figure 3.4: The number of correct and incorrect calls in the simulation from our proposed model.

small number of false calls, the comparison should focus on the performance in this region.

At the peptide level, the peptide probabilities calculated from either of our models consistently identify at least 100 more true peptides than PeptideProphet probabilities, at any controlled number of false peptides in the range of 0-200. At the protein level, our models consistently identifies more true proteins (≥ 5) than both the naive product rule and the adjusted product rule, at any controlled number of false proteins in the range of 0-50 (Figure 3.4). Our full model, which takes the protein status and the protein length into account when modeling n_k, identifies slightly more proteins than our reduced model, which assumes the common distribution of n_k for all proteins. Both our methods and the product rules perform substantially better than the 2-peptide rule.

It is interesting to note that, though the adjusted PeptideProphet probabilities identify only slightly fewer true peptides than our method, the adjusted product rule identifies many fewer true proteins than our method, especially when the number of false calls is small. This shows a common phenomenon of *ad hoc* rules, i.e. fixing one part of the problem but introducing problems in another part, due to their lack of coherence.

Similarities between identifications

It is of interest to check how similar the identifications made by our method and the competing programs are. At protein level, our full model identifies all 207 proteins that are identified by product rules, and also identifies 13 proteins that are not identified by product rules, when FDR=0. At peptide level, our full model identifies 917 peptides our of 918 peptides that are identified by PeptideProphet, and also identifies 190 peptides that are not identified by PeptideProphet, when FDR=0.

Calibration of probabilities

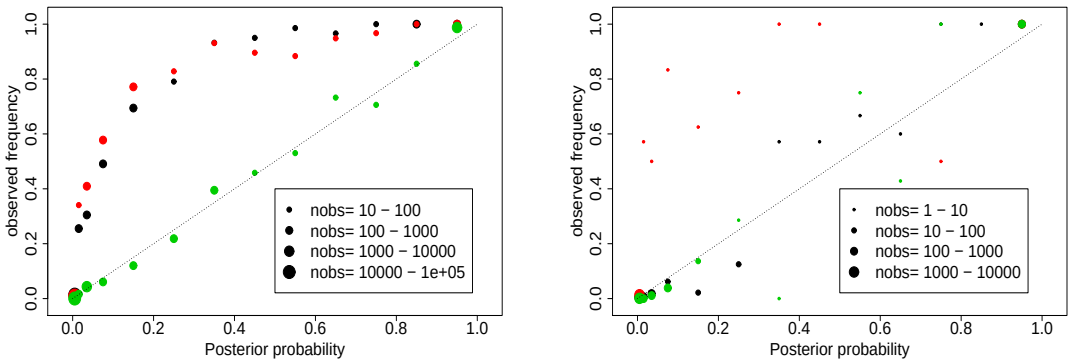
We hope to use the posterior probabilities estimated from our model as a way to assess the uncertainty of the calls. However, to justify this strategy, the model should ideally produce approximately calibrated probabilities.

To compare the calibration of prediction from our full model, PeptideProphet and product rules, the observations are binned by the assigned probabilities, then for each bin, the assigned probabilities are compared with the proportion of identifications that are actually correct.

The results show that the peptide probabilities calculated using our method are reasonably well calibrated on this simulated data (Figure 3.5b), but PeptideProphet probabilities are substantially smaller than the actual probabilities. Our method seems to also generate better-calibrated protein probabilities than ProteinProphet. However, as very few proteins are assigned probabilities $\in [0.2, 0.9]$, larger samples are needed for further confirmation.

Consistency of protein probabilities

To understand the behavior of the methods of comparison for different proteins, we examine how the deviation between the expected label and the true label ($P(T_k = 1) - T_k = E(T_k) - T_k$) varies with n_k for proteins with different statuses (Figure 3.6). The results show that all the methods underestimate the probability of being present for some present proteins that have small n_k. When n_k is large, the estimated label from our method approaches to the true label; the naive product rule overestimates probabilities for absent proteins even



a. Peptide identification

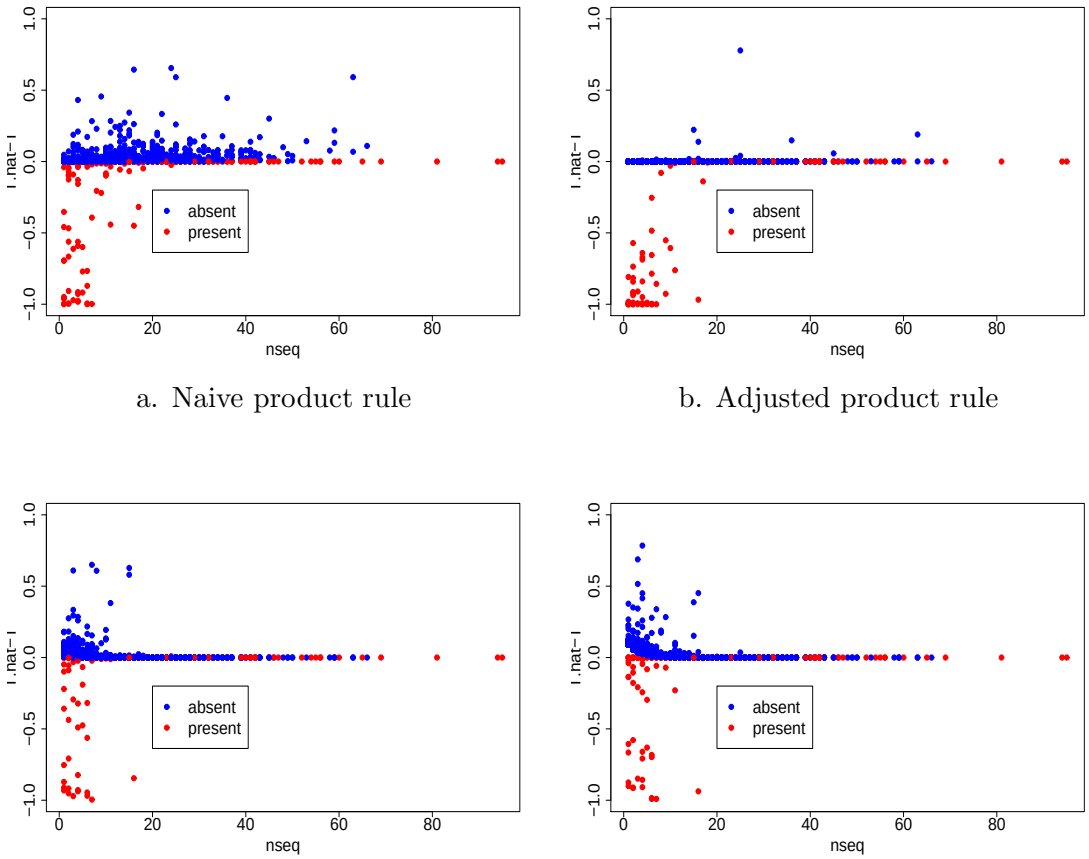
b. protein identification

Figure 3.5: Calibration of posterior probabilities in the simulation from our proposed model. Black: PeptideProphet (a) or naive product rule (b); Red: adjusted PeptideProphet probabilities (a) or adjusted produce rule (b); Green: our full model. The observations are binned by the assigned probabilities (bins are (0.00, 0.01, 0.02, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00)). The size of the points represents the number of observations in the bin.

when n_k is large. The adjusted product rule seems to estimate the probabilities well for absent proteins across different n_k, but it underestimates the probabilities for more present proteins with small n_k.

3.4.2 Practical effect of product rule

The product rule makes the assumption that the statuses of peptides on the same protein independently contribute to the protein probabilities. Since this assumption is blatantly wrong in practice, but is widely adopted by most existing approaches (Nesvizhskii et al., 2003; Price, 2007; Sadygov et al., 2004), it is of interest to explore the practical effect of this assumption on proteins with various characteristics, and compare with our approach. We first examine the scenarios that the product rules performs badly in the simulation in section 3.4.1. Then we simulate those scenarios to illustrate the performance degradation of the product rules.



a. Naive product rule

b. Adjusted product rule

c. Our full model

d. Our reduced model

Figure 3.6: Difference between estimated label and true label across different n_k in the simulation from our proposed model. Y-axis: $\hat{T} - T$; X-axis: number of unique peptide hits on a protein.

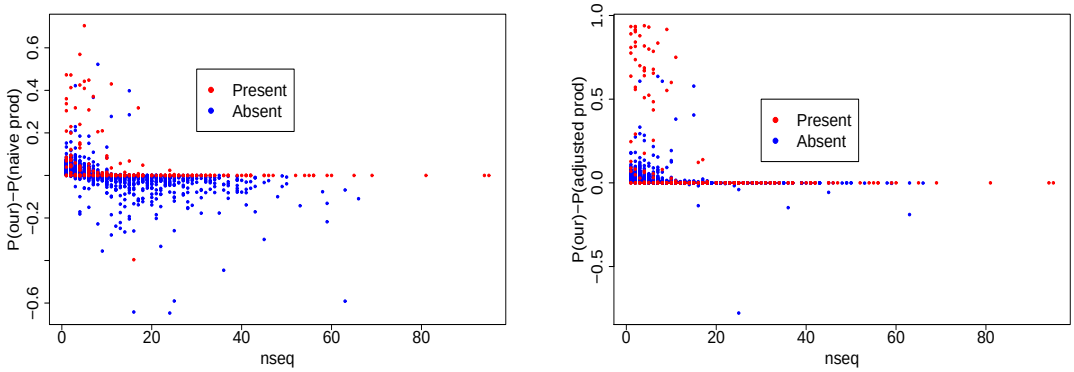
Deviation between protein probability assignment from our approach and product rules

We examined how probabilities assigned by our approach and by the product rules differ with the number of identified peptides n_k for proteins with different statuses, using the simulation in section 3.4.1. As shown in Figure 3.7a, when proteins are absent and have many identified peptides ($n_k > 10$), the naive product rules tend to overestimate the probabilities of being present; when proteins are present and have few identified peptides ($n_k < 10$), both product rules tend to underestimate the probabilities of being present. The adjusted product rule (Figure 3.7b) seems to reduce the inflation of probabilities for long proteins at the cost of further underestimating the probabilities of being present for short present proteins.

The behavior of naive product rules in the cases above is related to the independence assumption aforementioned: the product rules treat peptide statuses as independent when they are highly dependent, thus consequently overestimate certain type of probabilities consistently. When absent proteins that have many identified peptides contain a high-scored incorrect peptide, the product rules tend to call them present. For present proteins that have only several peptides, if they contain one or two correct peptides with mediocre scores and some incorrect ones, the product rules tend to call them absent. The examination of individual cases confirm that most mistakes made by the product rules belong to one of the two cases above.

Simulation against the product rules

Based on the observation from Figure 3.7, the combination of short present proteins and long absent proteins could maximize the difference between our approach and the product rule. So we simulate 1000 short present proteins ($l_k \in [100, 200]$), and 1000 long absent proteins ($l_k \in [1000, 2000]$) using the parameters in section 3.4.1 (Table 3.2), except we allow proteins that are absent to have high-scored incorrect peptide identifications occasionally (0.2%). The level of outliers was chosen according to the proportion of decoy peptides that are scored higher than the 90th percentile of the scores of the forward peptides in the yeast dataset. In the simulated dataset, the number of identified peptides are $n_k \in [1, 16]$



(a) naive product rule (b) adjusted product rule

Figure 3.7: Difference between the protein probabilities estimated from our full model and product rules in the simulation from our proposed model. Y-axis is $P_{our}(T = 1) - P_{prod}(T = 1)$.

for proteins that are present and $n_k \in [9, 51]$ for proteins that are absent. The parameter estimation is summarized in Table 3.2.

Tradeoff between true calls and false calls

In this dataset, as shown in Figure 3.8, our approaches substantially outperform both the adjusted and the naive product rules, in terms of the tradeoff between the numbers of true calls and false calls, at both the protein level and the peptide level. The gains of our approaches over the product rules are more pronounced in this simulation than the previous one at the protein level.

At the protein level, our full model consistently identifies at least 189 more proteins than both the adjusted and naive product rules at any number of false calls in the range of 0-50. The advantage of our approaches is more apparent, when only a small number of false calls is allowed.

At the peptide level, our models consistently identify at least 114 more peptides than PeptideProphet peptide probabilities. The adjusted peptide probabilities, unlike in the previous simulation, does not show much gain over the unadjusted PeptideProphet proba-

bilities on this dataset in the region of low false calls. The degraded gain of adjusted peptide probabilities on this dataset reveals that the heuristic adjustment of peptide probabilities is sensitive to the structure of data. Recall that the heuristic adjustment adjusts initial peptide probabilities according to the distribution of the estimated number of sibling peptides (denoted as NSP in ProteinProphet), defined as the sum of peptide probabilities for other peptides corresponding to the same protein. When a dataset consists of short true proteins and long false proteins, the distributions of this sum for true proteins and false proteins are less distinguishable, and consequently the adjustments provided are less efficient. However, our coherent methods are not affected by the structure of data.

It is also noted that our model with protein lengths performs substantially better than our model without protein lengths on this dataset. It confirms that incorporating protein lengths in the model helps distinguish proteins with different statuses. Although the gain from incorporating protein lengths here partially is due to the association between protein lengths and protein statuses in this simulation, we observe performance gain in other simulations where protein lengths have the same distribution regardless proteins statuses, though the gain is smaller than what is shown here.

Calibration of probabilities

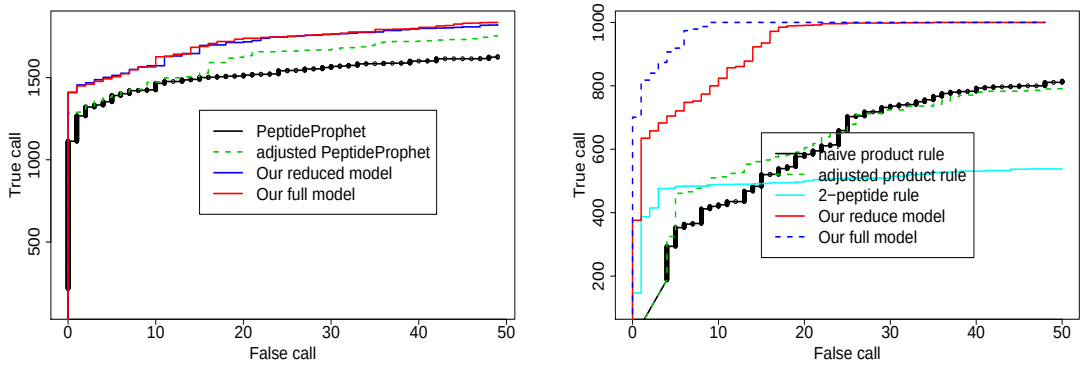
Similar to the previous simulation, our method calibrates the peptide probabilities very well, though PeptideProphet severely underestimates the probabilities (Figure 3.9a1-2). Both our approach and the product rules underestimate the protein probabilities between nominal probability of 0.2-0.9 (Figure 3.9b). Our method seems to be more conservative than the product rules in this range. However, less observations are assigned to this range by our method (5.2%) than by product rule (13.7%).

3.4.3 Sensitivity to violation of model assumptions

In our model, we assume that the proportion of incorrect peptides (π_1) is constant for the proteins that are present. However, this assumption may not hold in real data. It is of interest to check how sensitive our method is to the violation of this model assumption, and

Table 3.2: Simulation parameters and parameter estimation for the simulation consisting of 1000 short present proteins and 1000 long absent proteins. Gamma distribution is represented as Gamma(shape, scale, shift). π_0 is the proportion of incorrect peptides on the proteins that are absent in the simulation. π_0 is fixed at 1 in estimation. True: simulation parameters; Full: our full model; Reduced: our reduced model

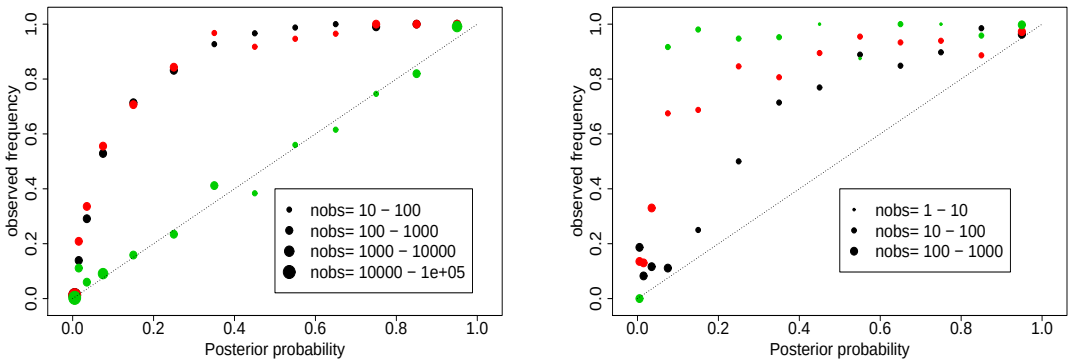
	π^*	c_0	c_1	π_0	π_1	f_0	f_1	$\log(L)$
True	0.5	0.018	0.033	0.998	0.58	$G(86.46, 0.093, -8.18)$	$N(3.63, 2.07^2)$	-51570.53
Full	0.55	0.018	0.034	-	0.56	$G(83.78, 0.096, -8.18)$	$N(3.71, 2.08^2)$	-51698.50
Reduced	0.55	-	-	-	0.54	$G(84.70, 0.095, -8.18)$	$N(3.69, 2.09^2)$	-



a. Peptide identification

b. Protein identification

Figure 3.8: The number of correct and incorrect calls in the simulation consisting of 1000 short present proteins and 1000 long absent proteins.



a. Peptide identification

b. Protein identification

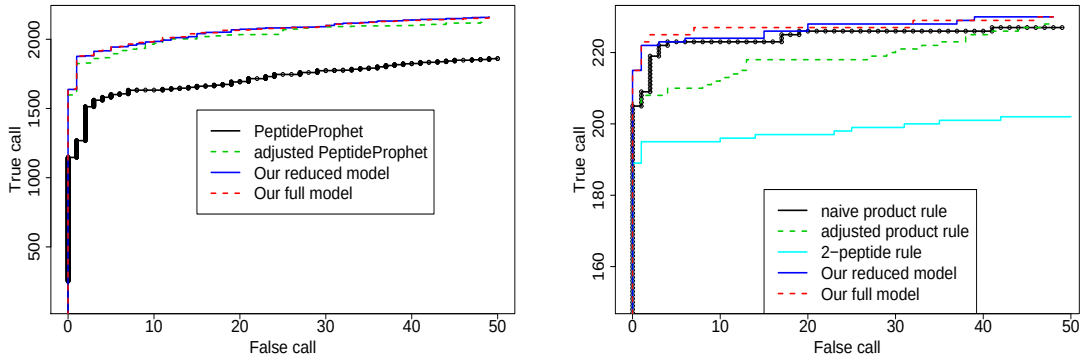
Figure 3.9: Calibration of posterior probabilities in the simulation consisting of 1000 short present proteins and 1000 long absent proteins. The size of the points represents the number of observations in the bin. Black: PeptideProphet in (a) or naive product rule in (b); Red: adjusted PeptideProphet probabilities in (a) or adjusted produce rule in (b); Green: our full model.

compare the performance with competing approaches in this scenario.

We did a simulation similar to section 3.4.1 with the same set of parameters, except $\pi_1 \sim unif(0, 0.8)$. The results show that, even with the violation of model assumptions, our method still produces reasonable parameter estimation (Table 3.3). The estimated proportion ($\hat{\pi}_1 = 0.4$) is the mean of the true distribution of π_1. Our method still outperforms PeptideProphet and the product rules in terms of the tradeoff between the numbers of true and false calls (Figure 3.10). The calibration results are similar to the results from the simulation without this violation (Figure 3.11). These indicates that our methods seem to be robust to this violation of model assumptions.

Table 3.3: Simulation parameters and parameter estimation for the simulation with heterogeneous $\pi_1 \sim Unif(0, 0.8)$. Protein length is simulated with exp(1/500). Gamma distribution is represented as Gamma(shape, scale, shift). π_0 is the proportion of incorrect peptides on the proteins that are absent in the simulation. π_0 is fixed at 1 in estimation. True: simulation parameters; Full: our full model; Reduced: our reduced model.

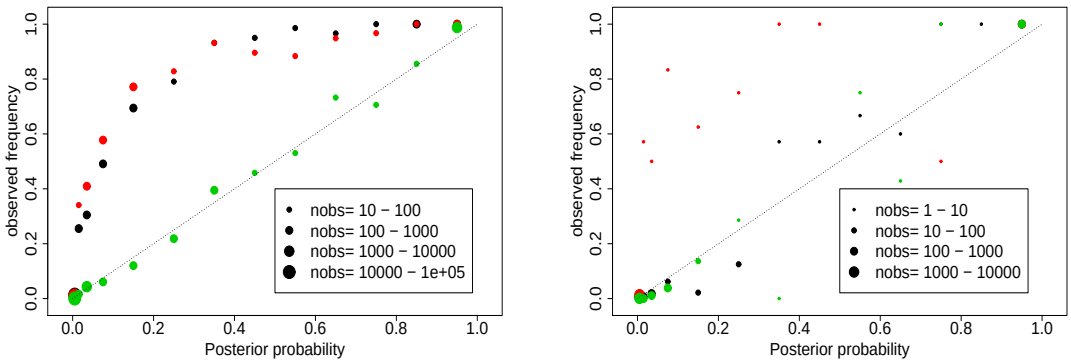
	π^*	c_0	c_1	π_0	π_1	f_0	f_1	$\log(L)$
True	0.88	0.018	0.033	1	$Unif(0, 0.8)$	$G(86.46, 0.093, -8.18)$	$N(3.63, 2.07^2)$	-37275.73
Full	0.88	0.018	0.034	-	0.40	$G(85.74, 0.094, -8.18)$	$N(3.68, 2.05^2)$	-37588.52
Reduced	0.87	-	-	-	0.40	$G(85.79, 0.094, -8.18)$	$N(3.68, 2.05^2)$	-



a. Peptide identification

b. Protein identification

Figure 3.10: The number of correct and incorrect calls for the simulation with $\pi_1 \sim Unif(0, 0.8)$. Our model outperforms PeptideProphet and the product rules in terms of the number of correct peptides and proteins identified at a given number of incorrect ones.



a. Peptide identification

b. Protein identification

Figure 3.11: Calibration of posterior probabilities on the simulation with $\pi_1 \sim Unif(0, 0.8)$. Parameters are estimated using the model with fixed π_1. The size of the points represents the number of observations in the bin. Black: PeptideProphet (a) or naive product rule (b); Red: adjusted PeptideProphet probabilities (a) or adjusted produce rule (b); Green: our full model.

3.5 Application on a yeast dataset

3.5.1 Description of data

We illustrate our method using a yeast dataset (Kall et al., 2007), which contains 140366 spectra. Because the true protein composition of this dataset is unknown, the spectra were searched against a database that is concatenated from the target database and a decoy database, which consists of dummy sequences created by permuting the sequences in the target database, using Sequest (Eng et al., 1994). After database search, we obtain 116264 unique putative peptide identifications, from which 12602 proteins are constructed using DTASelect (Tabb et al., 2002).

3.5.2 Performance comparison

We compared our algorithm with PeptideProphet for peptide inferences and actual ProteinProphet for protein inferences on this dataset. Because peptide sharing is present in this dataset, the overall performance reflects the combined results from the differences discussed earlier and differences due to handling of degeneracy and other heuristics in ProteinProphet.

To keep the comparisons on the same basis as much as possible, we followed several practices of data handling in PeptideProphet and ProteinProphet. For example, we used the same discriminant summary as in PeptideProphet to summarize peptide identification scores. Similar to ProteinProphet, we only kept one representative for each peptide that is identified by multiple spectra, because the spectra usually are highly correlated. However, the choices of representatives are different: ProteinProphet kept the one with the highest PeptideProphet probability, and we kept the one with the highest discriminant summary. Furthermore, we group the proteins that are indistinguishable by the identified peptides together and compute a group probability for each group, as in ProteinProphet. However, the group probability computed by ProteinProphet is the probability that any protein member is present in the sample, but ours is the highest protein probability in the group, which makes our group probability more conservative than the ones used by ProteinProphet. Though these two handlings report different group probabilities, we adopt this *ad hoc* grouping to keep comparisons on the same units (i.e. either protein or protein group). A more satisfactory approach is to coherently incorporate degeneracy into our model. We initialized our algorithm using the approach described in section 3.2.6, and stopped EM algorithm when the change of log-likelihood is smaller than 0.001. PeptideProphet and ProteinProphet were run with their default settings.

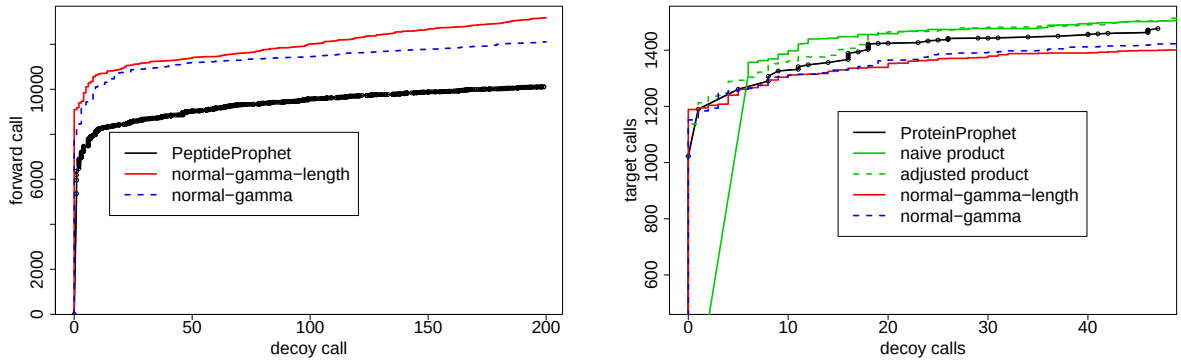
As the true identities are unknown, we assessed the accuracy of identification by comparing identifications marked by different methods as being present, at various score thresholds, in the target database and the decoy database. Note that the marks on decoy entries are false discoveries, and the ones on target entries contain *both true and false* discoveries, which is different from the simulation studies.

At a given number of decoy peptide calls, our model identified most target peptides identified by PeptideProphet, and a substantial number of target peptides that PeptideProphet does not (Figure 3.12a). For example, at FDR=0, our method identified 5362 peptides out of 5394 peptides that PeptideProphet identified, and additional 3709 peptides that PeptideProphet did not identify. At a given number of decoy protein calls, our model identified a similar number of target proteins (or protein groups) as ProteinProphet, when the number of decoy protein calls < 10 (Figure 3.12b). For example, at FDR=0, our methods identified 1190 proteins, and ProteinProphet identified 1023 proteins. Among them, 973 proteins are in common. Our method identified fewer proteins than ProteinProphet, when a larger number of false discovery is allowed (decoy protein > 10).

It is noted that the performance gain of our method over ProteinProphet in the simulation studies is not observed in this dataset. This may be due to the following reasons. First, our current approach makes nondegeneracy assumption, which is not true for real datasets. Second, in our experiment, we used a reshuffled database as the decoy database, where peptide sharing is destroyed when sequences are reshuffled in the generation of the decoy dataset; whereas, degeneracy exists in the target dataset. When ProteinProphet computes probabilities for the proteins with degeneracy, these proteins will be grouped together according to shared peptides, and assigned the probabilities that at least one of the peptide in any of the proteins in the group is present. Since this computation tends to overestimate the probabilities of being present for protein groups, the absence of degeneracy peptides in the reshuffled decoy database will cause ProteinProphet to identify more proteins from the target database at a given number of decoy calls.

3.5.3 Examination of model assumptions

Our models assume that the proportion of correct peptides is constant $(1 - \pi_1)$ for all the proteins that are present and is 0 for all the proteins that are absent. It is of interest to check if this assumption is approximately true in the real data. Thus we examine how the expected proportion of correct peptide identifications, which is computed as $\hat{\pi}^k = \frac{1}{n_k} \sum_i \hat{p}_{k,i}$,



a. Peptide identification b. Protein identification

Figure 3.12: The number of decoy calls and target calls on a yeast dataset calculated from different models. ProteinProphet: actual ProteinProphet; naive product: naive product rule; adjusted product: adjusted product rule; normal-gamma-length: our full model; normal-gamma: our reduced model.

a. number of identified peptides b. number of identified proteins

Figure 3.13: The numbers of common peptides and proteins identified by our method and PeptideProphet and ProteinProphet at FDR=0. Black: PeptideProphet or ProteinProphet; Red: our full method.

distributes for proteins with different statuses at FDR=0, using our full model (Figure 3.14a). The results show that the proportion is close to 0 for most decoy proteins, as what is assumed in our model for proteins that are absent. However, the proportion is fairly heterogeneous for target proteins that are called present. Though this indicates our assumption of a homogeneous π_1 is not realistic, the results from simulation studies in Section 3.4.2 seem to suggest that our method is robust to this departure of modeling assumption.

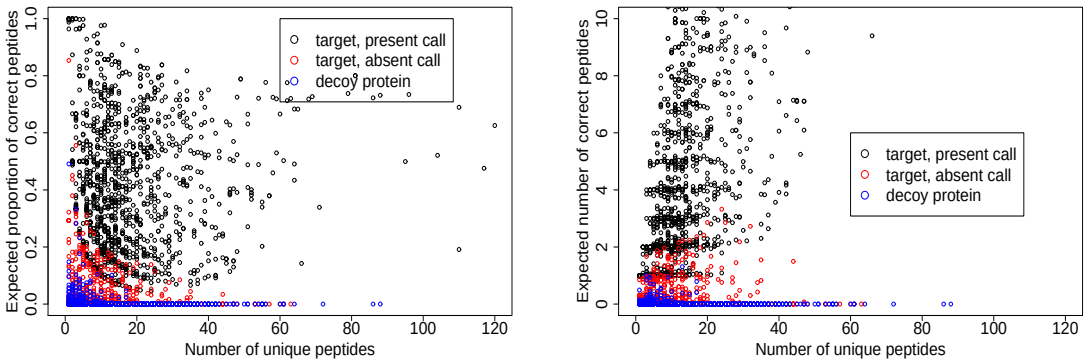
It is also noted that there is a notable curvature at the decision boundary, where proteins with smaller n_k's need to have a bigger proportion of correct peptides to be called as present. We offer two explanations for this phenomenon: First, the estimation of protein probabilities has bigger uncertainties at smaller n_k (for example, as shown in Figure 3.14a for decoy proteins). Second, the hyperbola shape of the curve suggests that the call is made when the expected number of correct peptides (i.e. the product of the values of x and y-axis) is over a certain value. This value actually is approximately 1, as shown in Figure 3.14b. It shows that our model requires at least one expected correct peptide for a protein to be called present, but the required amount of the expected correct peptide can come from *either one or multiple* peptides on the protein, rather than just from a *single* peptide as in the product rule of ProteinProphet.

3.6 Justification of model choices and modeling extensions

In this section, we provide justifications for some modeling choices and also present some additional modeling attempts that are not included in the previous sections. Though they were not adopted into the final model, they are included here to demonstrate the effects of these attempts and the rationale of our model choices.

3.6.1 Distribution choices for identification scores

In this section, we briefly describe how the distributions of identification scores (f_0 and f_1) are chosen.



a. Expected proportion of correct peptides b. Expected number of correct peptides

Figure 3.14: Relationship between expected proportion and expected number of high-scored peptide hits and the number of unique peptide hits. Computation is based on our full model. Black: target proteins identified at FDR=0, Red: target proteins that are not identified at FDR=0, Blue: decoy proteins.

As mentioned in Section 3.2.3, we choose the distributions of identification scores (f_0 and f_1) based on the shapes of the empirical observations, the goodness-of-fit between the selected distributions and the data, and the density ratio at the tails of the distributions. We provide more details here.

According to literature (Keller et al., 2002) and the empirical distribution of identification scores of decoy peptides, which approximate f_0, we consider shifted gamma or normal distributions for f_0. For f_1, we consider normal, shifted gamma and Gumbel distributions. A Gumbel distribution is considered because of its infinite support and asymmetric shape (a short left tail and a long right tail), which seems to be desirable for f_1.

To select from the choices above, we first fit different f_0 to the scores of decoy peptides. Based on BIC (Gamma: BIC= 141168.3; Normal: BIC=139820.9), a normal distribution fits better to the decoy data than a gamma distribution. We then fit a mixture model without protein level information for the scores of target peptides using different combinations of the choices above. It shows normal or gamma distribution fits better than Gumbel distribution

Table 3.4: Goodness-of-fit for fitting the distributions of identification scores.

f_0	f_1		
	Normal	shifted Gamma	Gumbel
Normal	371454.7	372287.7	374046.8
shifted Gamma	370154.7	371496.5	372648.6

for f_1 (Table 3.4).

Since the primary goal here is to estimate peptide labels, we further check the tail behavior of the distributions for estimating the posterior probabilities of the label. In order to assign peptide labels properly in the mixture model, it requires $f_0/f_1 > 1$ in the left tail of f_0 and $f_1/f_0 > 1$ in the right tail of f_1. Because a gamma distribution has a much shorter left tail than a normal distribution, observations on the far left tail of f_0 may have density ratio $f_1/f_0 > 1$, when f_0 is gamma and f_1 is normal (PeptideProphet's choice). As a result, those low-scored peptide identifications will pathologically receive high posterior probabilities of being correct, which is sometimes observed in the peptide assignments given by PeptideProphet. In addition, short left tail for f_1 is preferred to reduce overlap between f_0 and f_1.

Thus, we choose to model f_0 with a normal distribution and f_1 with a shifted gamma distribution. This is also confirmed by the estimation of peptide assignment using empirical data: our choice identifies more target peptides than the Gamma-Normal model at a given number of false calls (results not show).

We also consider a nonparametric approach, where both f_0 and f_1 are estimated using histogram density estimators. However, as shown in simulation studies (Figure 3.15), $\hat{f}_0$ and $\hat{f}_1$ estimated from this approach are always mixtures of f_0 and f_1 (regardless of the starting points), though the estimation of g_0 and g_1 are similar to their parametric counterparts. This indicates that the histogram estimators have difficulties distinguishing information from different layers of the nested structure.

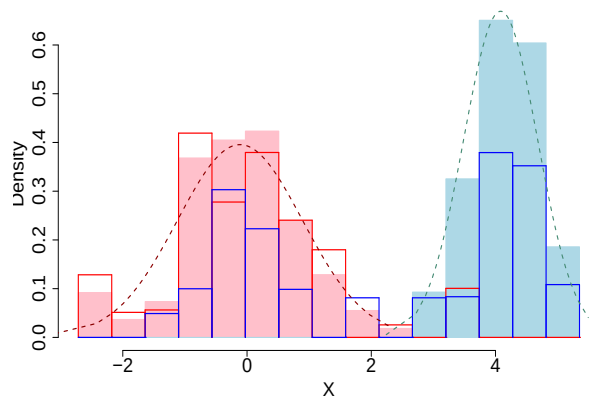


Figure 3.15: Score distributions estimated using normal or histogram density estimators in simulation studies. The pink/light blue filled histograms are the true distributions of f_0 and f_1 in a simulated dataset, respectively. The dark red/marine curves are the estimation from the normal estimators for f_0 and f_1, respectively. The red/blue histograms are the estimation from the histogram estimators for f_0 and f_1, respectively.

3.6.2 Model allowing outliers

In the models reported earlier, the identification scores of peptides in absent proteins are modeled using a low value component f_0. However, it is possible that incorrect peptides occasionally have high scores. To allowing this type of outliers, we modify g_0 in the previous model (3.2.4) as follows:

$$g_0(\mathbf{x_k}) \equiv P(\mathbf{X_k} \mid n_k, T_k = 0) = \prod_{i=1}^{n_k} [\pi_0 f_0(x_{k,i}) + (1 - \pi_0) f_1(x_{k,i})] \tag{3.6.1}$$

where $1 - \pi_0$ is the proportion of incorrect peptides with high scores on the proteins that are absent.

We then assess the performance of this model on the yeast dataset and estimate π_0 in the EM procedure along with other parameters. However, this model identifies many fewer target proteins than the previous model in (3.2.4), which actually is a special case of (3.6.1) with the constraint of $\pi_0 = 1$ (Figure 3.18). As shown in Figure 3.16, this model tends to draw a higher decision boundary on the expected proportion of correct peptides than

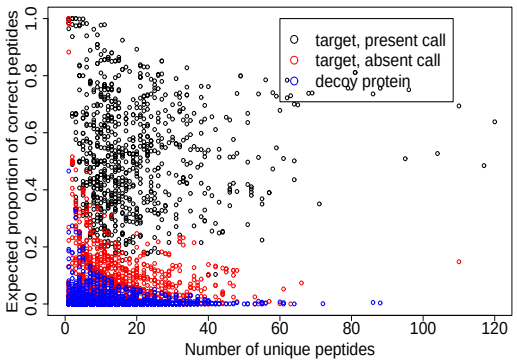


Figure 3.16: Relationship between the expected proportion of high-scored peptide hits and the number of unique peptide hits. Computation is based on our model allowing high-scored incorrect peptide identification in absent proteins. Black: target proteins identified at FDR=0, Red: target proteins that are not identified at FDR=0, Blue: decoy proteins.

the constrained model, when clustering the proteins. Consequently, it has the tendency to assign target proteins with low proportion of correct peptides as being absent, though they have much higher expected proportions of correct peptides than decoy proteins. A close examination of Figure 3.16 suggests that the decision boundary seems to be drawn to partition the proteins into two visually compact groups of $\hat{\pi}^k$. This model actually converges to a likelihood higher than the one from the constrained model (3.2.4). My conjecture is that the clustering method is misled by within-cluster variation in the less compact cluster, due to highly different compactness of the two clusters in π_i (i=0,1). The constraint of $\pi_0 = 1$ regulates the clustering method in some sense.

An examination on this yeast dataset shows that only 0.2% decoy peptides have identification scores higher than 90th percentile of the scores of target peptides. This indicates that π_0 is very close to 1 in this dataset. Given this, we suggest to use the constrained model.

3.6.3 Model with multiple clusters

In the view of model-based clustering, our models presented above essentially cluster the proteins into two clusters, one without correctly identified peptides and another with a proportion of $(1 - \pi_1)$ correctly identified peptides. The observed heterogeneity on π_1 for present proteins suggests that we may model them with multiple clusters.

As a first attempt, we fix the number of clusters as known and explore the effect of multiple clusters. Here, we consider two 10-cluster models: one uses prespecified proportions, which are fixed throughout of the estimation procedure, distributed evenly from 0.1 to 1; the other starts from the same proportions as in the former, but the proportions are updated during estimation. The chosen number of clusters is probably larger than what is present in the data. We purposely made this choice to ensure that the heterogeneity is captured by different clusters. The proteins assigned to the cluster with $\pi_i = 1$ are deemed as absent. In this model, we use $f_1 = Gumbel(mode, scale)$.

Figure 3.18 shows that the 10-cluster model with fixed π_i identifies a similar number of proteins as the 2-cluster methods at low decoy calls, and identifies more proteins than 2-cluster methods at higher numbers of decoy calls. However, the 10-cluster model with varied π_i performs worse than the 2-cluster model. The difference between the two variants may be explained by the regulation effect due to fixing the proportions of the clusters.

The parameters estimated at protein level (Table 3.5) and the 10 clusters (Figure 3.17) suggest that the data perhaps has fewer than 10 clusters. Because the interpretation of multiple clusters is less intuitive than the 2-cluster model and the performance gain is small, we do not proceed in this direction to find the optimal number of clusters.

3.6.4 Model with a varied mixing proportion

Empirical observations (Figure 3.17) show that long proteins that are present tend have lower proportion of correct peptide identifications than the short proteins that are present. This suggests that we may take account of some of the heterogeneity of π_1 by modeling it as a function of the protein length.

Table 3.5: Parameter estimation (protein-level) for two 10-cluster models.

		cluster									
		0	1	2	3	4	5	6	7	8	9
Fix	π^*	0.802	0.124	0.004	0.016	0.015	0.018	0.0001	0.013	0.006	0.0012
	π_i	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
	h	0.018	0.020	0.053	0.029	0.030	0.044	0.200	0.060	0.100	0.003
No Fix	π^*	0.742	0.131	0.051	0.035	0.016	0.006	0.001	0.007	0.009	0.003
	π_i	1.000	0.919	0.993	0.693	0.493	0.502	0.229	0.225	0.292	0.396
	h	0.018	0.019	0.030	0.029	0.038	0.052	0.157	0.091	0.057	0.060

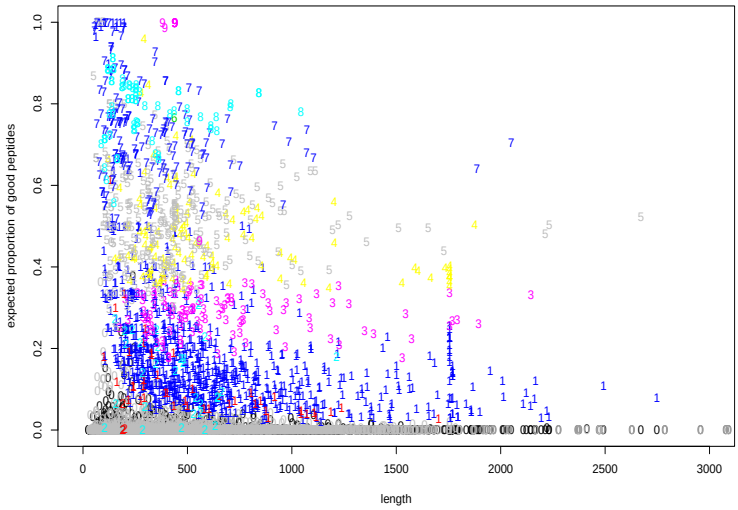


Figure 3.17: Relationship between the expected proportion of high-scored peptide hits and protein length for a 10-cluster Normal-Gumbel full model with fixed π_i. The cluster labeled 0 is the cluster with $\pi_i = 1$. The decoy proteins in cluster 0 is colored in grey. The decoy proteins in other clusters (cluster 1 or 2) are marked in red by their cluster numbers. Each point is one protein rather than a protein group.

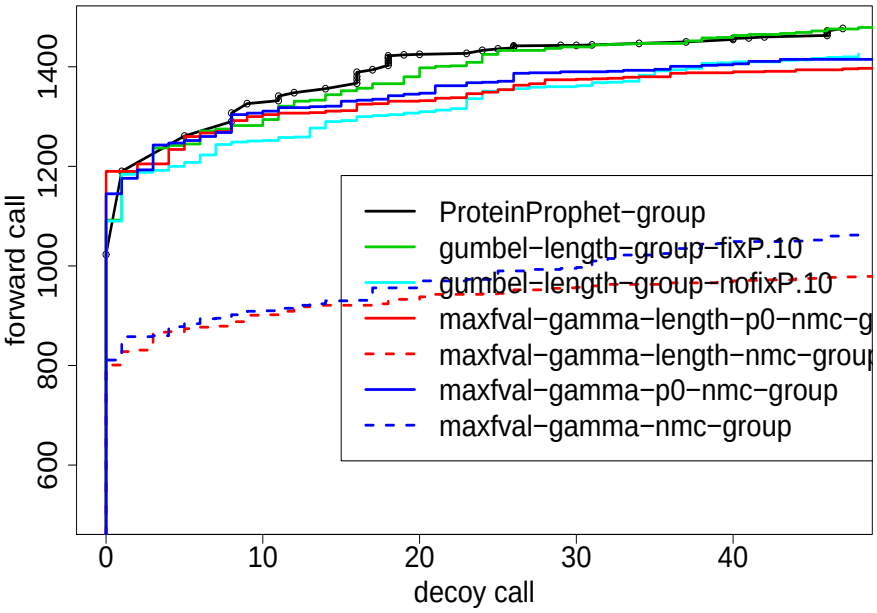


Figure 3.18: The number of decoy calls and target calls on the yeast dataset calculated from the Normal-Gumbel models with 10 clusters and Normal-Gamma models with 2 clusters. ProteinProphet-group: actual ProteinProphet; Gumbel-length-group-fixP.10: 10 clusters Normal-Gumbel full model with fixed π_i; Gumbel-length-group-nofixP.10: 10 clusters Normal-Gumbel full model with varied π_i; maxfval-gamma-length-p0-nmc-group: 2-cluster Normal-Gamma full model with fixed $\pi_0 = 1$; maxfval-gamma-p0-nmc-group: 2-cluster Normal-Gamma reduced model with fixed $\pi_0 = 1$; maxfval-gamma-length-nmc-group: 2-cluster Normal-Gamma full model and π_0 is estimated; maxfval-gamma-nmc-group: 2-cluster Normal-Gamma reduced model and π_0 is estimated.

One possibility is to associate π_j with length using a logistic regression, as follows. For a protein k with length l_k,

$$\log(\frac{\pi_j^k}{1-\pi_j^k}) = a_j l_k + b_j,$$

where a_j, b_j are parameters associated with the cluster $j(j > 0)$. The incorporation of this association slightly increases the number of identified target proteins at a given decoy identifications (Results not shown).

3.7 Discussion

We have presented a new statistical method for assessing evidence for presence of proteins and constituent peptides identified from mass spectra. Our approach essentially is a model-based clustering method, which simultaneously identifies which proteins are present, and which peptides are correctly identified. It provides a coherent framework to model the nested relationship between peptides and their parent proteins. It provides a properly-calibrated probability for peptide inference and protein inference. It shows improved accuracy over a widely-used program for protein inference in the simulation studies, and substantially increased accuracy over a widely-used program for peptide identification in both simulated studies and the real data we studied.

The key to our increased accuracy is the use of a coherent statistical framework, based on a nested mixture model, to incorporate the evidence feedback between proteins and the peptides nested on them. The feedback from protein evidence to peptide evidence helps distinguish peptides that are correctly identified but with weak scores from those that are incorrectly identified but with high scores.

The idea of combining peptide and protein inference is also reported in a recent paper (Shen et al., 2008), where they assess the confidence of peptides and proteins jointly using a hierarchical model. They used a generative model to generate proteins, peptides, spectra and identification scores for individual spectrum in a hierarchical structure, and estimated the parameters and latent variables using EM. However, in empirical comparisons Shen et al found their method identified many fewer proteins than PeptideProphet-ProteinProphet

two-stage approach.

Our model presented here assumes that present proteins have a homogeneous proportion of incorrect peptides (π_1), which is unlikely to be true in real data. However, this assumption can be relaxed. Our simulation results under a heterogeneous π_1 provide evidence that our method is relatively robust to the deviation from this assumption.

It is worth noting that the independence assumption in the product rule, which is adopted in several protein identification approaches (Nesvizhskii et al., 2003; Price, 2007), inflates the probability of being present for the proteins with many peptide hits and is sensitive to high-scored incorrect peptide identifications, due to its ignorance of the dependence between the correctness of peptides on the same proteins. Our simulation has confirmed that it has the tendency to identify long proteins with occasional high-scored incorrect peptides (Figures 3.7 and 3.8-b), as commented elsewhere (Sadygov et al., 2004; Shen et al., 2008); whereas, our method is relatively robust.

There are several possible improvements on our current work. The selection of features of identification scores is critical to the validation of peptide identification. Currently, we follow the practice of PeptideProphet, since the focus of our method is not feature selection. It is also possible to incorporate other features or adopt feature summaries from other packages, for example, the SVM score from Percolator (Kall et al., 2007).

Our approach is an unsupervised algorithm, which uses decoy results to initialize the incorrect peptide identifications. If the score distribution of identified decoy peptide approximates that of incorrect identification close enough, we may gain more discrimitive power by using a semi-supervised approach that models incorrect identification with decoy results. It is straightforward to alter our method into the semi-supervised way. This approach has been adopted in a semi-supervised version of PeptideProphet (Choi and Nesvizhskii, 2008b).

An important future work is to extend our approach to deal with degeneracy, which is prevalent in high-level organisms. Our current approach ignores the dependence between proteins that share peptides, which will overcount the shared peptides. The fact that our approach does not yet handle this issue properly may partly explain why the performance

gain of our method over ProteinProphet in the simulation studies is not observed in the real dataset. Currently, most existing approaches to handle degeneracy are based on heuristics. For example, ProteinProphet groups the proteins with shared peptides and assigns weights to degenerate peptides using heuristics. An exception is the hierarchical model used in Shen et al. (2008), which contains a layer in the hierarchy to indicate which protein(s) each degenerate peptide comes from and infers the indicator from the observed data. Though the empirical performance in Shen et al. (2008) is somewhat disappointing, this seems to be a sensible way to handle degeneracy, and it seems that out model could be extended in a similar way.

Appendix

Here we describe an EM algorithm for the estimation of $\Phi = (\pi_0^*, \pi_0, \pi_1, \mu, \sigma, \alpha, \beta, \gamma, c_0, c_1)^T$ and the protein statuses and the peptide statuses. To proceed, we use T_k and $(P_{k,1}, \ldots, P_{k,n_k})$ as latent variables, then the complete log-likelihood for the augmented data $Y_k \equiv (\mathbf{X_k}, n_k, T_k, (P_{k,1}, \ldots, P_{k,n_k}))$ is

$$
\begin{aligned}
& l^C(\Psi \mid \mathbf{Y}) \qquad (3.7.1) \\
& = \sum_{k=1}^{N} \Bigg\{ (1-T_k)[\log \pi_0^* + \log h_0(n_k \mid l_k, n_k > 0) + \sum_{i=1}^{n_k} (1-P_{k,i}) \log(\pi_0 f_0(x_{k,i})) + \\
& \sum_{i=1}^{n_k} P_{k,i} \log((1-\pi_0) f_1(x_{k,i}))] \Bigg\} \\
& + \sum_{k=1}^{N} \Bigg\{ T_k[\log(1-\pi_0^*) + \log h_1(n_k \mid l_k, n_k > 0) + \sum_{i=1}^{n_k} (1-P_{k,i}) \log(\pi_1 f_0(x_{k,i})) + \\
& \sum_{i=1}^{n_k} P_{k,i} \log((1-\pi_1) f_1(x_{k,i}))] \Bigg\}
\end{aligned}
$$

E-step:

$$Q(\Psi, \Psi^{(t)}) \equiv E(l^C(\Psi) \mid \mathbf{x}, \Psi^{(t)}) \tag{3.7.2}$$
$$= \sum_{k=1}^{N} P(T_k = 0)\{\log \pi_0^* + \log h_0(n_k \mid l_k, n_k > 0)$$
$$+ \sum_{i=1}^{n_k} P(P_{k,i} = 0 \mid T_k = 0) \log(\pi_0 f_0(x_{k,i})) + \sum_{i=1}^{n_k} P(P_{k,i} = 1 \mid T_k = 0) \log((1-\pi_0) f_1(x_{k,i}))\}$$
$$+ \sum_{k=1}^{N} P(T_k = 1)\{\log(1-\pi_0^*) + \log h_1(n_k \mid l_k, n_k > 0)$$
$$+ \sum_{i=1}^{n_k} P(P_{k,i} = 0 \mid T_k = 1) \log(\pi_1 f_0(x_{k,i})) + \sum_{i=1}^{n_k} P(P_{k,i} = 1 \mid T_k = 1) \log((1-\pi_1) f_1(x_{k,i}))\}$$

Then

$$\hat{T}_k^{(t)} \equiv E(T_k \mid \mathbf{x_k}, n_k, \Psi^{(t)}) \tag{3.7.3}$$
$$= \frac{P(T_k = 1, \mathbf{x_k}, n_k \mid \Psi^{(t)})}{P(\mathbf{x_k}, n_k \mid \Psi^{(t)})}$$
$$= \frac{(1-\pi_0^{*(t)}) g_1^{(t)}(\mathbf{x_k}, n_k \mid \Psi^{(t)}) h_1(n_k)}{\pi_0^{*(t)} g_0^{(t)}(\mathbf{x_k}, n_k \mid \Psi^{(t)}) h_0(n_k) + (1-\pi_0^{*(t)}) g_1^{(t)}(\mathbf{x_k}, n_k \mid \Psi^{(t)}) h_1(n_k)}$$

$$\hat{I}_0^{(t)}(P_{k,i}) \equiv E(P_{k,i} \mid T_k = 0, x_{k_i}, \Psi^{(t)}) \tag{3.7.4}$$
$$= \frac{P(P_{k,i} = 1, x_{k_i} \mid T_k = 0, \Psi^{(t)})}{P(x_{k_i} \mid T_k = 0, \Psi^{(t)})} = \frac{(1-\pi_0^{(t)}) f_1^{(t)}(x_{k_i})}{\pi_0^{(t)} f_0^{(t)}(x_{k_i}) + (1-\pi_0^{(t)}) f_1^{(t)}(x_{k_i})}$$
$$\hat{I}_1^{(t)}(P_{k,i}) \equiv E(P_{k,i} \mid T_k = 1, x_{k_i}, \Psi^{(t)}) = \frac{(1-\pi_1^{(t)}) f_1^{(t)}(x_{k_i})}{\pi_1^{(t)} f_0^{(t)}(x_{k_i}) + (1-\pi_1^{(t)}) f_1^{(t)}(x_{k_i})} \tag{3.7.5}$$
$$\tag{3.7.6}$$

M-step:

Now we need maximize $Q(\Psi, \Psi^{(t)})$. Since the mixing proportions and the distribution parameters can be factorized into independent terms, we can optimize them separately. The MLE of the mixing proportion π_0^* is:

$$\pi_0^{*(t+1)} = \frac{\sum_{k=1}^{N}(1-\hat{T}_k^{(t)})}{N} \tag{3.7.7}$$

$$\pi_0^{(t+1)} = \frac{\sum_{k=1}^{N}[(1-\hat{T}_k^{(t)})\sum_{i=1}^{n_k}(1-\hat{I}_0^{(t)}(P_{k,i}))]}{\sum_{k=1}^{N}(1-\hat{T}_k^{(t)})n_k} \tag{3.7.8}$$

$$\pi_1^{(t+1)} = \frac{\sum_{k=1}^{N}[\hat{T}_k^{(t)}\sum_{i=1}^{n_k}(1-\hat{I}_1^{(t)}(P_{k,i}))]}{\sum_{k=1}^{N}\hat{T}_k^{(t)}n_k} \tag{3.7.9}$$

If incorporating ancillary features of peptides, $f_j(x_{k_i}) = f_j(x_{k_i}^S)f_j^{nmc}(x_{k,i}^{nmc})f_j^{ntt}(x_{k,i}^{ntt})$ as in Section 3.2.4, where $x_{k_i}^S$ is the identification score, $x_{k,i}^{nmc}$ is the number of missed cleavage and $x_{k,i}^{ntt}$ is the number of tryptic termini. As described in Section 3.2.3, $f_0^S \sim N(\mu, \sigma^2)$ and $f_1^S \sim Gamma(\alpha, \beta, \gamma)$. We can obtain closed form estimators for f_0^S as follows, and estimate f_1^S using the numerical optimizer optimize() in R.

$$\mu = \frac{\sum_{k=1}^{N}\sum_{i=1}^{n_k}\left[(1-\hat{T}_k^{(t)})(1-\hat{I}_0^{(t)}(P_{k,i})) + \hat{T}_k^{(t)}(1-\hat{I}_1^{(t)}(P_{k,i}))\right]x_{k_i}}{\sum_{k=1}^{N}\sum_{i=1}^{n_k}\left[(1-\hat{T}_k^{(t)})(1-\hat{I}_0^{(t)}(P_{k,i})) + \hat{T}_k^{(t)}(1-\hat{I}_1^{(t)}(P_{k,i}))\right]} \tag{3.7.10}$$

$$\sigma^2 = \frac{\sum_{k=1}^{N}\sum_{i=1}^{n_k}\left[(1-\hat{T}_k^{(t)})(1-\hat{I}_0^{(t)}(P_{k,i})) + \hat{T}_k^{(t)}(1-\hat{I}_1^{(t)}(P_{k,i}))\right](x_{k_i}-\mu_0)^2}{\sum_{k=1}^{N}\sum_{i=1}^{n_k}\left[(1-\hat{T}_k^{(t)})(1-\hat{I}_0^{(t)}(P_{k,i})) + \hat{T}_k^{(t)}(1-\hat{I}_1^{(t)}(P_{k,i}))\right]} \tag{3.7.11}$$

So the MLE of f_0^{nmc} is:

$$f_0^{nmc}(x_{k,i}^{nmc}) = \frac{w_j^{(t)}}{\sum_{j=0}^{1} w_j^{(t)}} \tag{3.7.12}$$

where

$$w_j^{(t)} = \sum_{k=1}^{N}\sum_{i=1}^{n_k}(1-\hat{T}_k^{(t)})(1-\hat{I}_0^{(t)}(P_{k,i}))1(x_{k,i}^{nmc}=j) + \sum_{k=1}^{N}\sum_{i=1}^{n_k}\hat{T}_k^{(t)}(1-\hat{I}_1^{(t)}(P_{k,i}))1(x_{k,i}^{nmc}=j) \tag{3.7.13}$$

Similarly, the MLE of f_1^{nmc} is:

$$f_1^{nmc}(x_{k,i}^{nmc}) = \frac{v_j^{(t)}}{\sum_{j=0}^{1} v_j^{(t)}} \tag{3.7.14}$$

where

$$v_j^{(t)} = \sum_{k=1}^{N}\sum_{i=1}^{n_k}(1-\hat{T}_k^{(t)})\hat{I}_0^{(t)}(P_{k,i})1(x_{k,i}^{nmc}=j) + \sum_{k=1}^{N}\sum_{i=1}^{n_k}\hat{T}_k^{(t)}\hat{I}_1^{(t)}(P_{k,i})1(x_{k,i}^{nmc}=j) \quad (3.7.15)$$

The MLE of f_j^{ntt} takes the same format as f_j^{nmc}, j=0,1.

For h_0 and h_1, the terms related to h_0 and h_1 in $Q(\Psi, \Psi^t)$ are:

$$\sum_{k=1}^{N}(1-\hat{T}_k)\log h_0(n_k) = \sum_{k=1}^{N}(1-\hat{T}_k)\log\frac{\exp(-c_0 l_k)(c_0 l_k)^{n_k}}{n_k!(1-\exp(-c_0 l_k))} \quad (3.7.16)$$

$$\sum_{k=1}^{N}\hat{T}_k\log h_1(n_k) = \sum_{k=1}^{N}\hat{T}_k\log\frac{\exp(-c_1 l_k)(c_1 l_k)^{n_k}}{n_k!(1-\exp(-c_1 l_k))} \quad (3.7.17)$$

The MLE of the above does not have close form, so we estimate them using optimiz() in R.

Chapter 4

SUMMARY AND FUTURE DIRECTIONS

In this thesis, we developed statistical methods for peptides and proteins identification using mass spectra. We showed that statistical models that carefully model the noise structure were well suited for revealing underlying structures from these complex biological data sets. Here, I summarize the main contributions of the dissertation, go over some important future work, connect our model to applications in other fields where similar structures are encountered, and briefly mention some statistical problems in other proteomics research that is based on mass spectrometry.

4.1 Main contributions of the dissertation

4.1.1 Peptide identification

We have developed a likelihood-based scoring algorithm based on a generative model, which scores each candidate sequence by the likelihood that the observed spectrum arises from its theoretical spectrum. By explicitly modeling the noise structure, our probability model takes account of multiple sources of noise in the data, e.g. variable peak intensities and errors in peak locations. This attribute enables our method to extract some of the more subtle signals which other methods miss, such as peak intensities and information on sophisticated theoretical prediction. Our results demonstrate that incorporating peak intensities improves the accuracy of peptide identification.

We also provide two statistical measures for assessing the uncertainty of each identification for our likelihood-based scores. These measures allow one to determine the uncertainty of identification with respect to all the candidates of a given observed spectrum.

4.1.2 Protein identification

We have developed a new statistical approach for assessing evidence for presence of proteins and constituent peptides identified from mass spectra. Our approach is essentially a model-based clustering method, based on a nested mixture model. In contrast to commonly-used two-stage approaches, our model provides a one-stage solution that simultaneously identifies which proteins are present, and which peptides are correctly identified. In this way, our model provides a coherent framework to incorporate the evidence feedback between proteins and their constituent peptides. As a result, our method provides properly calibrated probability for peptide inference and protein inference. In the comparison with widely-used two-stage approaches, our single-stage approach yields consistently more accurate results for peptide identification. For protein identification, our method and the existing method have similar accuracy in most settings, although we exhibit some scenarios in which the existing method perform poorly.

4.2 Handling degenerate peptides

We developed a unified statistical method for assessing evidence for presence of proteins and constituent peptides identified from mass spectra. However, our current model does not deal with the degeneracy case, which is prevalent in high-level organisms. One possibility is to add an additional layer between proteins and peptides to indicate which protein(s) the degenerate peptide comes from, then infer this latent variable from the observed data. This idea is also used in a recent paper by Shen et al. (2008) to handle degeneracy.

Another possibility is to view the collection of putative peptides and proteins as a bipartite graph. In this graph, the nodes involved in homolog proteins are densely connected and the connections between other proteins are sparse. As many proteins do not have homologs, the graph contains many independent proteins and some groups of dependent proteins (i.e. homologs). Factor graphs (Frey, 2003; Kschischang et al., 2001) seem to be suitable for this data structure. Given the size of nodes in this type of data, we may use variational methods for inference (Wainwright and Jordan, 2005).

4.3 Inference on nested structures

We developed a nested mixture model for making inference on proteins and their constituent peptides based on the information from peptides. Mixture models with nested structures are also seen elsewhere, such as in text mining, social science, etc.

In social science, the structure of the multilevel latent class model (Vermunt, 2003) bears some similarities to our model. However, there are obvious distinctions. In that model, a lower-level cluster usually is a random sample from a upper-level cluster. The hierarchical structure is a device to divide the observations into relatively homogeneous groups. The primary interest often is not the assignment of the membership. Whereas, in our model, the lower-level elements are constituent component of the upper-level elements. Though there is a nested relationship between the upper-level *elements* and their constituent lower-level *elements* in our model, the *clustering* of lower-level elements is not nested under the *clustering* of upper-level elements. Our goal is to cluster the observations at both levels.

Mixture models with hierarchical structures are often used for categorizing documents into hierarchies of topics, for example, the hierarchical topic model (Blei et al., 2004), the mixed-membership model (Erosheva et al., 2004). However, their focus often is to learn the structure of the hierarchy or to cluster documents into latent topics. In many cases, the number of clusters is assumed infinite, then Dirichlet process is used for estimation, which is very different from our application.

4.4 Statistical problems in other proteomics research

As proteomics is still new to statisticians, here I briefly mention some statistical problems in other MS-based proteomics research. The choice of topics is purely based on my experience and personal opinion.

4.4.1 Signal extraction from mass spectra

Some statistical work has been done on extracting signals from MS1 data (e.g. Figure 1.1E, without fragmentation), such as MALDI, where measurements are taken on unfragmented

peptides and signals are present in irregular peaks. Currently, the main statistical focus is to use nonparametric approaches (J.S. et al., 2008; Harezlak et al., 2008; Randolph et al., 2005), such as wavelet or functional data analysis, to detect and extract peaks.

4.4.2 Quantitative proteomics

Mass spectrometry is increasingly used for relative or absolute quantification of peptides and proteins. The goal of quantitative proteomics experiments, which somewhat is similar to microarray experiments, is to quantify changes in the abundance of features of interest across the samples that are compared. The signals from these type of experiments are similar to those from two-channel microarray.

However, the error structure in quantitative proteomics data are more complicated than microarray experiments due to differences in experimental procedures. For example, protein samples often need undergo separation steps that operate on a continuous scale, before the measurements of interest are taken. As samples elute from the separation device in a continuous gradient, the measurements from neighboring separation fractions usually are spatial correlated. This type of correlation is not present in most microarray experiments, where measurements are taken from discrete spots. In addition, unlike microarray experiments, where the measurements are taken from known identities, the identities of the measurements in the proteomic experiments often need to be identified, i.e. involving a step of protein or peptide identification. This step introduces an additional level of error, which often be quite substantial, as shown in this dissertation. Li et al. (2008) shows an example of this sort.

Though some statistical methods developed for microarray experiments may be applied for some quantitative experiments, these complicated error structures introduce many open problems and impose new statistical challenges. Some problems have tangible underlining structure and are statistically interesting, such as the spatial correlation problem mentioned above. However, due to the amount of errors propagated from different sources in many steps, majority of discoveries from current quantitative proteomics experiments are not

reliable. This unreliability mainly is due to the immaturity of technologies, biological or technical variation in the sample, or poor experimental design or sample processing. Statisticians need bear in mind of the limitations of technology when choosing problems to work on!

BIBLIOGRAPHY

Aebersold, R. and M. Mann (2003). Mass-spectrometry-base proteomics. *Nature 422*, 198–207.

Bafna, V. and N. Edwards (2001). Scope: a probabilistic model for scoring tandem mass spectra aganist a peptide database. *Bioinformatics 17 (Suppl. 1)*, S13–S21.

Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc., Ser. B 57*, 289–300.

Blei, D., T. Gri, M. Jordan, and J. Tenenbaum (2004). Hierarchical topic models and the nested chinese restaurant process. In *NIPS*.

Choi, H. and A. I. Nesvizhskii (2008a). Semisupervised model-based validation of peptide identifications in mass spectrometry-based proteomics. *J. Proteome Res. 7*, 254–265.

Choi, H. and A. I. Nesvizhskii (2008b). Semisupervised model-based validation of peptide identifications in mass spectrometry-based proteomics. *J. Proteome Res. 7*, 254–265.

Coon, J. J., J. E. Syka, J. Shabanowitz, and D. Hunt (2005). Tandem mass spectrometry for peptide and proteins sequence analysis. *BioTechniques 38*, 519–521.

Dempster, A., N. Laird, and D. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *J. R. Statist. Soc. B 39*(1), 1–38.

Elias, J., F. Gibbon, O. King, F. Roth, and S. Gygi (2004). Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nature Biotechnol. 22*, 214–219.

Elias, J. and S. Gugi (2007). Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods 4*, 207–214.

Eng, J., A. McCormack, and J. I. Yates (1994). An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom 5*, 976–989.

Erosheva, E. A., S. E. Fienberg, and J. Lafferty (2004). Mixed-membership models of scientific publications. *Proceedsings of the national academy of sciences 101(Suppl. 1)*, 5220–5227.

Frey, B. J. (2003). Extending factor graphs so as to unify directed and undirected graphical models. In *Proceedings of the 19th conference in uncertainty in artificial intelligence*, Volume Aug 7-10, pp. 257–264.

Harezlak, J., M. Wu, M. Wang, A. Schwartzman, D. Christiani, and X. Lin (2008). Biomarker discovery for arsenic exposure using functional data: Analysis and feature learning of mass spectrometry proteomic data. *J Proteome Research 7*, 217–224.

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

Hernandez, P., M. Muller, and R. D. Appel (2006). Automated protein identification by tandem mass spectrometry: issues and strategies. *Mass Spectrometry Reviews 25*, 235–254.

J.S., M., B. P.J., H. R.C., K. Baggerly, and C. K.R. (2008). Bayesian analysis of mass spectrometry data using wavelet based functional mixed models. *Biometrics 2008*, 479–489.

Kall, L., J. Canterbury, J. Weston, and M. J. Noble, W. S.and MacCoss (2007). A semi-supervised machine learning technique for peptide identification from shotgun proteomics datasets. *Nature Methods 4*, 923–925.

Keller, A. (2002). Experimental protein mixture for validating tandem mass spectral analysis. *Omics 6*, 207–12.

Keller, A., A. Nesvizhskii, E. Kolker, and R. Aebersold (2002). Empirical statistical model to estimate the accuracy of peptide identifications made by ms/ms and database search. *Anal. Chem. 74*, 5383–5392.

Kinter, M. and N. E. Sherman (2003). *Protein sequencing and identification using tandem mass spectrometry*. Wiley.

Klammer, A. A., S. Reynolds, M. J. MacCoss, J. Bilmes, and W. Noble (2008). Modelling peptide fragmentation with dynamic bayesian networks for peptide identification. *Bioinformatics In press*.

Kschischang, F. R., B. J. Frey, and H. A. Loeliger (2001). Factor graphs and the sum-product algorithm. *IEEE transactions on information theory 47*, 498–519.

Li, Q., M. Fitzgibbon, and M. McIntosh (2008). Statistical methods for detecting differentially expressed protein isoforms in high-throughput quantitative experiments. In preparation.

Li, Q., Q. Xia, T. Wang, M. Meila, and M. Hackett (2006). Analysis of the stochastic variation in ltq single scan mass spectra. *Rapid Communications in Mass Spectrometry 20*, 1551–1557.

McLachlan, G. and D. Peel (2000). *Finite Mixture Models*. Wiley.

Nesvizhskii, A. and R. Aebersold (2005). Interpretation of shotgun proteomic data: the protein inference problem. *Mol. Cell. Proteomics 4*, 1419–1440.

Nesvizhskii, A., A. Keller, E. Kolker, and R. Aebersold (2003). A statistical model for identifying proteins by tandem mass spectrometry. *Anal. Chem. 75*, 4646–4653.

Nesvizhskii, A., O. Vitek, and R. Aebersold (2007). Analysis and validation of proteomic data generated by tandem mass spectrometry. *Nature Methods 4*, 787–797.

Nesvizhskii, A. I. and R. Aebersold (2004). Analysis, statistical validation and dissermination of large-scale proteomics datasets generated by tandem ms. *Drug Discovery Todays 9*, 173–181.

Perkins, D. N., D. J. Pappin, D. M. Creasy, and J. Cottrell (1999). Probability-based protein identification by searching sequence database using mass spectrometry data. *Electrophoresis 20*, 3551–3567.

Price, T. e. a. (2007). Ebp, a program for protein identification using multiple tandem mass spectrometry data sets. *Mol. Cell. Proteomics 6*, 537–536.

Randolph, T. W., B. L. Mitchell, D. F. McLerran, P. D. Lampe, and Z. Feng (2005). Quantifying peptide signal in maldi-tof mass spectrometry data. *Molecular and Cellular Proteomics 4*, 1990–1999.

Sadygov, R., D. Cociorva, and J. Yates (2004). Large-scale database searching using tandem mass spectra: looking up the answer in the back of the book. *Nature methods 1*, 195–202.

Sadygov, R., H. Liu, and J. Yates (2004). Statistical models for protein validation using tandem mass spectral data and protein amino acid sequence databases. *Anal Chem 76*, 1664–1671.

Sadygov, R. and J. Yates (2003). A hypergeometric probability model for protein identification and validation using tandem mass spectral data and protein sequence databases. *Anal. Chem. 75*, 3792–3798.

Schutz, F., E. A. Kapp, R. J. Simpson, and T. P. Speed (2003). Deriving statistical models for predicting peptide tandem ms product ion intensitiyes. *Biochemical Society Transactions 31*, 1479–1483.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics 6*, 461–464.

Shen, C., Z. Wang, G. Shankar, X. Zhang, and L. Li (2008). A hierarchical statistical model

to assess the confidence of peptides and proteins inferred from tandem mass spectrometry. *Bioinformatics 24*, 202–208.

Steen, H. and M. Mann (2004). The abc's (and xyz's) of peptide sequencing. *Nature Reviews 5*, 699–712.

Tabb, D., C. Fernando, and M. Chambers (2007). Myrimatch: highly accurate tandem mass spectral peptide identification by multivariate hypergeometric analysis. *Journal of proteome research 6*, 654–661.

Tabb, D., H. McDonald, and J. I. Yates (2002). Dtaselect and contrast: tools for assembling and comparing protein identifications from shotgun proteomics. *J. Proteome Res. 1*, 21–36.

Tanner, S., H. Shu, A. Frank, L.-C. Wang, E. Zandi, M. Mumby, P. Pevzner, , and V. Bafna (2005). Inspect: Fast and accurate identification of post-translationally modified peptides from tand em mass spectra. *Anal. Chem. 77*, 4626–4639.

Vermunt, J. K. (2003). Multilevel latent class models. *Sociological Methodology 33*, 213–239.

Wainwright, M. J. and M. Jordan (2005). *A variational principle for graphical model*, Chapter 11. MIT Press.

Wan, Y., A. Yang, and T. Chen (2006). Pephmm: A hidden markov model based scoring function for mass spectrometry database search. *Anal. Chem. 78*, 432–437.

Wysocki, V. H., G. Tsaprsilis, L. Smith, and L. A. Breci (2000). Mobile and localized protons: a framework for understanding peptide dissociation. *J. Mass Spectrom 35*, 1399–1406.

Zhang, Z. (2004). Prediction of low-energy collision-induced dissociation spectra of peptides. *Analytical Chemistry 76*, 3908–3922.

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

VITA

Qunhua Li was born in Wuhan, China. She received her Bachelor of Science in Biology from Wuhan University in July 1997. From August 1997 to January 1999, she studied genetics in the Ph.D program in Texas A&M University. She then switched to the statistics program in Texas A&M University and received the Master of Science degree in statistics in 2000. After working in Eli Lilly and company for one year, she joined the Ph.D program in statistics in University of Washington in 2001. In 2008, she graduated with a Doctor of Philosophy in Statistics.