11-22-2017 2:30 PM

NBPMF: Novel Network-Based Inference Methods for Peptide Mass Fingerprinting

Zhewei Liang
The University of Western Ontario

Supervisor
Zhang, Kaizhong
The University of Western Ontario

Graduate Program in Computer Science
A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

Abstract

Proteins are large, complex molecules that perform a vast array of functions in every living cell. A proteome is a set of proteins produced in an organism, and proteomics is the large-scale study of proteomes. Several high-throughput technologies have been developed in proteomics, where the most commonly applied are mass spectrometry (MS) based approaches. MS is an analytical technique for determining the composition of a sample. Recently it has become a primary tool for protein identification, quantification, and post translational modification (PTM) characterization in proteomics research. There are usually two different ways to identify proteins: top-down and bottom-up. Top-down approaches are based on subjecting intact protein ions and large fragment ions to tandem MS directly, while bottom-up methods are based on mass spectrometric analysis of peptides derived from proteolytic digestion, usually with trypsin.

In bottom-up techniques, peptide mass fingerprinting (PMF) is widely used to identify proteins from MS dataset. Conventional PMF representatives such as probabilistic MOWSE algorithm, is based on mass distribution of tryptic peptides. In this thesis, we developed a novel network-based inference software termed NBPMF. By analyzing peptide-protein bipartite network, we designed new peptide protein matching score functions. We present two methods: the static one, ProbS, is based on an independent probability framework; and the dynamic one, HeatS, depicts input dataset as dependent peptides. Moreover, we use linear regression to adjust the matching score according to the masses of proteins. In addition, we consider the order of retention time to further correct the score function. In the post processing, we design two algorithms: assignment of peaks, and protein filtration. The former restricts that a peak can only be assigned to one peptide in order to reduce random matches; and the latter assumes each peak can only be assigned to one protein. In the result validation, we propose two new target-decoy search strategies to estimate the false discovery rate (FDR). The experiments on simulated, authentic, and simulated authentic dataset demonstrate that our NBPMF approaches lead to significantly improved performance compared to several state-of-the-art methods.

Keywords: Network-Based Inference, Peptide Mass Fingerprinting, Proteomics, Protein Identification, Mass Spectrometry, Knowledge-Discovery in Databases, Recommender Systems, Bioinformatics, Data Mining.

To My Family

&

For My Grandmother's 99th Birthday

Acknowlegements

First of all, I am extremely proud to be a student supervised by my supervisor Dr. Kaizhong Zhang. To me, he is the most inspirational researcher in my life. He unconditionally and persistently supported my research over the past years. I am extremely overwhelmed with delight when considering him as my mentor and lifetime friend. Without his patience, dedication, and guidance I could not make this research through to its completion.

I have always felt very fortunate that I have spent my graduate studies at Western in London, Ontario, Canada. It was definitely a remarkable eight years of my life which I will never forget.

I would like to thank all my dear colleagues, Weiming Li, Yi Liu, Weiping Sun, Fang Han, Lin He, Yan Yan, Yu Shan, Shaofeng Jiang, Yiwei Li, Qin Dong, Yu Qian, Wenjing Wan, in room 222 at Middlesex College, the great friends, the brightest faculty, and the best staff members in the Department of Computer Science at Western.

I would like to thank our collaborator, Dr. Gilles Lajoie, for his insightful guidance on my research topics survey. Thanks are also owed to the examiners of my thesis proposal, Dr. Lila Kari and Dr. Mark Daley, for their useful advice to improve this project at the earlier stage. Thanks as well to my thesis examiners, (in alphabetical order) Dr. Lucian Ilie, Dr. Shun-Cheng Li, Dr. Bin Ma, and Dr. Roberto Solis-Oba, for their helpful suggestions concerning this work.

Last but not least, I am extremely thankful to my family for their love and support.

Contents

List of Figures

List of Tables

List of Abbreviations

NBPMF Novel Network-Based Inference Methods for Peptide Mass Fingerprinting

MS Mass Spectrometry

PTM Post-Translational Modification

LC Liquid Chromatography

DDA Data-Dependent Acquisition

DIA Data-Independent Acquisition

LC-MS Liquid Chromatography and Mass Spectrometry

LC-MS/MS Liquid Chromatography and Tandem Mass Spectrometry

SRM Selected Reaction Monitoring

AMT Accurate Mass and Time

RT Retention Time

PMF Peptide Mass Fingerprinting

MALDI Matrix-Assisted Laser Desorption Ionization

TOF Time-Of-Flight

HCP Host Cell Proteins

DNA Deoxyribonucleic Acid

NGS Next-Generation Sequencing

RNA Ribonucleic Acid

mRNA messenger RNA

rRNA ribosomal RNA

tRNA transfer RNA

NMR Nuclear Magnetic Resonance

NHGRI National Human Genome Research Institute

PPI Protein-Protein Interaction

ESI Electrospray Ionization

IT Ion Trap

FTICR Fourier Transform Ion Cyclotron Resonance

Da Dalton

FWHM Full Width at Half Maximum

mDa milli Dalton

mmu milli mass units

ppm parts per million

Q Quadrupole

LIT Linear Ion Trap

QIT Quadrupole Ion Trap

CID Collision Induced Dissociation

CAD Collision Activated Dissociation

IRMPD Infrared Multiphoton Dissociation

BIRD Blackbody Infrared Dissociation

SORI Sustained Off Resonance Irradiation

ECD Electron Capture Dissociation

SID Surfaceinduced Dissociation

ETD Electron Transfer Dissociation

HCD Higher-energy C-trap Dissociation

MS/MS Tandem Mass Spectrometry

PFF Peptide Fragment Fingerprinting

FDR False Discovery Rate

NIST National Institute of Standards and Technology

SNP Single Nucleotide Polymorphism

HPLC High-Performance Liquid Chromatography

OPM Ordered Peptide Match

NBI Network-Based Inference

ProbS Probabilistic Spreading

HeatS Heat Spreading

IDE Integrated Development Environment

TP True Positive

FP False Positive

UniProtKB UniProt Knowledgebase

TERF1 Telomeric Repeat-binding Factor 1

sPRG Proteomics Standards Research Group

BIC sPRG Bioinformatics Committee

LR Linear Regression

ORT Ordered Retention Time

HI Hydrophobicity Index

AP Assignment of Peaks

PF Protein Filtration

PSM Peptide-Spectrum Match

TC Total Correct

TI Total Incorrect

FN False Negative

TN True Negative

CDF Cumulative Distribution Function

GEV Generalized Extreme Value

Chapter 1

Introduction

1.1 Research Objective and Motivation

Proteins are a class of organic compounds that play many critical roles in all living organisms. Proteins such as skin, hair, and muscles, can hold together, protect, and provide structure to the body's tissues and organs. Proteins are made up of hundreds or thousands of amino acids, which are linked by peptide bonds. Additionally, there are 20 different types of amino acids that are common in humans and animals [152].

Proteomics is the large-scale study of proteomes, and a proteome is a set of proteins produced in an organism, system, or biological context [147]. Proteomics can be used to investigate post-translational modifications (PTMs) such as phosphorylation. Moreover, proteomics can provide significant information for many biological problems. Consequently, several high-throughput technologies have been developed to investigate proteomes in depth, where mass spectrometry and micro arrays are the common methods used in proteomics.

Mass spectrometry (MS) is one of the most informative techniques for determining the composition of a sample. It is true that for obtaining detailed structural information, MS cannot compete with methods such as nuclear magnetic resonance and X-ray crystallography. However, MS is much easier to automate and use as a large-scale technique [83]. Recently, MS has

become a primary tool for protein identification, quantification, and PTM characterization in proteomics research.

There are usually two different approaches by MS to identify proteins: top-down and bottom-up. In top-down proteomics, intact protein ions can be generated by electrospray mass spectrometry, then introduced into a mass analyzer and subjected to gas-phase fragmentation. Top-down MS has the ability to sequence intact proteins, especially for the analysis of PTMs. In conventional bottom-up method, protein identification is based on mass spectrometric analysis of peptides derived from proteolytic digestion, usually with trypsin. Here, proteolysis is the breakdown of proteins into smaller peptides. Intact protein sequencing [107] was difficult due to its large molecular weight, and requires more expensive MS instrumentation, which hinders the implementation of this technique in a typical academic laboratory. However nowadays, some top-down approaches are used in proteomics such as analyzing PTMs [65], mapping intact protein isoforms [124], etc. Some researchers also combined bottom-up and top-down mass spectrometry analysis of the histone PTMs, and their combinatorial patterns [18].

In traditional bottom-up approach, the proteins may first be purified by gel electrophoresis resulting in one or a few proteins in each proteolytic digest. Alternatively, in shotgun proteomics, the crude protein extract (prepared by removal of cellular debris generated by cell lysis) is digested directly, followed by one or more dimensions of separation of the peptides by liquid chromatography (LC) coupled to MS [156]. There are usually two modes for bottom-up approaches, the most widespread one is *data-dependent acquisition* (DDA), where selected peptide precursors (usually are the peptide peaks with high intensity) following chromatographic separation are fragmented by MS/MS [74]. Another mode is *data-independent acquisition* (DIA), where all ions within a selected *m/z* range (a small window that contains several peptide peaks) are fragmented and analyzed in tandem MS. DIA is an alternative to DDA, where a fixed number of precursor ions are selected and analyzed by tandem MS [156].

In wet-lab procedures for protein identification based on the most used DDA mode, a sample is first digested by enzyme. Then liquid chromatography and tandem mass spectrometry

(LC-MS/MS) are used for analyzing the resultant peptides. This bottom-up approach attempts to reconstruct the original protein sample based on identified peptides, since they can be surrogates for their parent proteins (peptides can be used to identify proteins). In order to analyze the dataset, we should have a protein sequence database that contains all target proteins. Each MS/MS scan is used to identify a peptide-spectrum match (given a mass tolerance threshold) from it; finally, these peptides are searched against the database to identify the proteins.

For tandem mass spectra in DDA mode, there are roughly four ways to interpret the dataset and identify the fragmentation of proteins: sequence database searching, spectral library searching, database-independent approach (*de novo* sequencing), and the hybrid interpretation algorithms. In the first approach, peptide identification is performed by correlating experimental MS/MS spectra with theoretical spectra predicted for each peptide contained in a protein sequence database (database search approach) [17, 26, 37, 97, 100, 103, 84]. In the second approach, instead of searching acquired MS/MS spectra against theoretically predicted spectra, one can assign MS/MS spectra to peptides by matching against a spectral library (spectral library searching) [132, 27, 42, 63]. In the third approach, peptide sequences are extracted directly from the spectra, i.e., without referring to a sequence database for help (*de novo* sequencing approach) [122, 82, 81]. The forth category is hybrid approaches, such as those based on the extraction of short sequence tags (a piece of information about a peptide that contains three to five amino acid residues) followed by "error-tolerant" database searching, where the error tolerant search of uninterpreted MS/MS data is a powerful way of finding additional peptide matches [85, 121, 41]. Moreover, the different approaches can get bonus from the combination of each other. For example, the performance of *de novo* sequencing can be greatly improved, with the use of spectra pairs from the same peptide under different fragmentation modes [130].

Each of the approaches has its own strengths and weaknesses. The database searching approaches are good at "reidentifying" proteins and peptides, which is especially difficult in *de novo* interpretation if the ion series are incomplete (some ions are missed). The advantage of the *de novo* sequencing approach over the database search method is that it can make new

discoveries. However, *de novo* analysis is computationally intensive and requires high-quality MS/MS spectra. In the high-throughput proteomics environment, researchers are not always interested in the peptides where there is no exact match in the database. As a result, the computational analysis starts typically with database searching, and only then, if desired, *de novo* sequencing tools are used to the remaining unassigned spectra [83]. Therefore, the combination of database searching and *de novo* sequencing is often used in some softwares, such as PEAKS [82].

Certain challenges will arise when the above enzymatic digestion LC-MS/MS work flow is applied to complex protein samples, such as plasma or a whole-cell lysate. For example, after digestion a sample of proteins can produce a multitude of peptides, including expected, missed cleavages (the allowing of internal trypsin cleavage sites), and PTMs. This will lead to a peptide under-sampling problem, which means not all of the peptides can be identified. Even with thorough sample preparation and chromatographic separation, the introductions of peptides into the mass spectrometer are still faster than their isolations and fragmentations. Therefore, the majority of peptides in the sample are often left unanalyzed. Even in another alternative DIA mode, such as selected reaction monitoring (SRM) [3, 104] or accurate mass and time (AMT) tags [116], under-sampling is unlikely eliminated completely.

To avoid the above problems, advances in LC and MS technologies make it possible to identify peptides solely on their MS masses and retention time (RT) without MS/MS, where RT is the amount of time a compound spends on the column after it has been injected. These advances require instrumentation capable of high-accuracy measurements, LC systems with sufficient RT precision, as well as precise prediction algorithms for relative RT [62, 94, 93]. This technique is analogous to traditional peptide mass fingerprinting (PMF), which has long been used for protein identification separated by gel-electrophoresis [112]. PMF (also known as protein fingerprinting) is an analytical technique to identify protein, where the unknown protein of interest is first cleaved into smaller peptides, whose absolute masses can be accurately measured with a mass spectrometer [24].

PMF has some advantages over MS/MS. For example, Matrix-Assisted Laser Desorption Ionization Time-Of-Flight (MALDI-TOF) MS is significantly less expensive and faster than MS/MS, followed by PMF in the protein identification [89]. Another application where PMF is advantageous over MS/MS is in the host cell proteins (HCPs) detection. HCPs are those produced or encoded by the organisms and unrelated to the intended recombinant product [127]. These HCPs usually have small number of proteins, and their abundance is too low to be identified by MS/MS. Therefore, we should use MS and PMF to detect HCPs. Additionally, for some proteins such as histones, they would be cleaved to short peptides (3-4 amino acid residues) after digestion, which are hardly to be identified by MS/MS. Consequently, MS and PMF should be applied for them.

Although PMF has these advantages, due to the lack of specificity with a low-accuracy dataset in peptide identification, PMF has been limited to low complexity samples. The reason for the limitation is that each mass used for fingerprinting can typically be assigned to several peptides from different proteins. Therefore in a complex sample, it becomes impossible to infer potential presented proteins.

Traditional PMF was first introduced in 1993, and the representative was MOWSE [100]. Until now, the most popular PMF methods are Mascot (1999) [103], MS-Fit (1999) [24], ProFound (2000) [137], and Aldente (2003) [125]. Among them, Mascot performed the best in several studies [20, 109]. More recently, PMF was tackled as an optimization problem assuming each peptide peak can only be assigned to one protein (2010) [50]. In addition to the peptide mass, the RT was also used to facilitate protein identification (2011) [17].

Mascot uses a proprietary scoring function based on the probabilistic MOWSE algorithm, which calculates distribution of a tryptic peptide mass across an entire search database. Mascot also computes a probability for each observed peak for a match (the probability represents that whether the observed match is a random event), and the lowest one is reported as the best match. In practice, an event is significant if it would be expected to occur at random with a frequency of less than 5%. Therefore, whether it is also a significant match depends on

the size of the database [103]. This means that searches in smaller protein databases, such as bacterial, will generally have lower threshold scores for confidence than those conducted in larger databases for higher organisms [86]. As a result, it is worthy of finding other PMF solutions relies less on whether the size of database.

Another consideration for traditional PMF methods is that they assume the score of peptide to be independent. Additionally, the score for each peptide can be calculated only from the probability. As we know, statistics is based on the past knowledge, while each time the MS dataset comes from proteolytic digests of complex protein mixtures. There should be a relationship between these peptides and proteins. For example, when many peptides in a protein are matched, these matching peptides will be dependent, consequently all their scores for this protein could be increased for more accurate computation. Therefore, by mining useful information from current input datasets dynamically, we could get better performance.

When treating the relationship between peptides and proteins in the database as a bipartite graph or network (peptides and proteins), we may use network analysis techniques to discover important peptide-protein information between dependent peptides. Here we present novel network-based inference methods for PMF, termed *NBPMF* [69], to carry out such analyses. We also have two methods: ProbS, which is based on a probability framework; and HeatS, for a dynamic analysis. We propose that with accurate mass measurements, and suitable RT prediction algorithms, it is possible to identify proteins confidently in a digest of medium complexity sample (not too complex with several thousand proteins) based on LC-MS characteristics alone. We also use linear regression to adjust the matching score according to the masses of proteins. In the post processing, we restrict that a peak can only be assigned to one peptide in order to reduce random matches. Additionally, we try to filter out false positive proteins, assuming each peak can only be assigned to one protein. Finally, we use target-decoy strategy for finding a proper threshold to select the number of identified proteins. The experiments on simulated and real data demonstrate that our NBPMF approaches lead to significantly improved performance compared to several state-of-the-art methods [70].

By avoiding MS/MS, our approaches not only share the advantage of analyzing an identical set of peptide features like DIA, but also sample peptides more efficiently than DDA. This software capable of identifying proteins in a yeast digest with results that are comparable to common LC-MS/MS methods, if the sample is not too complex, such as with several hundred proteins. Given the increasing prevalence of high mass accuracy instrumentation, this approach should be widely applicable in the proteomics community, either alone or as a complement to other techniques.

Our NBPMF software also has the ability to solve protein inference problem [96] with MS/MS. For those identified peptides by MS/MS, if they are unique peptides, they could have high weights; otherwise, they would have medium weights and be adjusted by our NBPMF software. Besides the identified peptides from MS/MS, we also have the mass information for unidentified peptides from MS, which would have the low weights in our system. Now, we could combine the unique identified peptides, the other identified peptides, and the unidentified peptides, to infer proteins. Consequently, their scores would be adjusted by NBPMF approaches. As a result, these adjusted scores could be used for the protein inference, where the performance has the possibility to be improved.

1.2 Chapter Outlines

This thesis is organized into the following Chapters:

Chapter 1 gives an introduction to our research objective and motivation.

Chapter 2 introduces some basic biology knowledge in bioinformatics, including DNA, genes, chromosomes, RNA, amino acids, peptides, and proteins. We also give a general summary on mass spectrometry (MS) instruments. Additionally, we compare several protein identification methods by mass spectrometry, including top-down, peptide mass fingerprinting (PMF), and tandem mass spectrometry.

Chapter 3 reviews the development of PMF, and formulates PMF problem mathemati-

cally. Consequently, we introduce some state-of-the-art PMF methods, especially for MOWSE, optimization-based PMF, and PMF by the order of retention time.

Chapter 4 presents the concept of recommender systems, and formulates network-based inference approaches mathematically. We develop two novel network-based inference methods for PMF: probabilistic spreading (ProbS), which is based on a probability framework; and heat spreading (HeatS), for a dynamic analysis. Additionally, we did the experiments on simulated, authentic, and simulated authentic dataset. The results demonstrate that our network based inference approaches achieved improved performance.

Chapter 5 shows how to adjust the raw scores. The first method used is linear regression, which is trying to tackle the bias between larger proteins and smaller proteins. The second approach is using the order of retention time with dynamic programming, for further adjustment of the protein score.

Chapter 6 designs post processing algorithms to correct the scores with two approaches. The first method is the assignment of peaks, which restricts that a peak can only be assigned to one peptide in order to reduce random matches. The second approach is protein filtration, assuming each peak can only be assigned to one protein.

Chapter 7 uses target decoy search strategy to estimated the false discovery rate (FDR) for a score threshold. We review target-decoy search strategy for tandem mass spectrometry, and introduce the concept of decoy database. Moreover, we design two new target-decoy strategies for PMF, including random dataset and decoy dataset.

Chapter 8 draws the conclusion on this thesis. Finally, we give a discussion about the possible research work in the future.

Chapter 2

Background

Bioinformatics is an interdisciplinary field that develops approaches and software tools to understand biological data. As an interdisciplinary field of science, bioinformatics combines Computer Science, Biology, Mathematics, and Engineering for analyzing and interpreting biological data [156].

In this chapter, we will review some basic biology knowledge, and hope this would facilitate the readers (especially those without a biology degree) to understand the biological data processed in this thesis. We will introduce the following background knowledge, including DNA, genes, chromosomes, RNA, amino acids, peptides, proteins. We also give a general summary on mass spectrometry instruments. Consequently, we review several methods for protein identification by mass spectrometry, such as top-down, peptide mass fingerprinting, and tandem mass spectrometry.

2.1 DNA, Genes, and Chromosomes

All organisms consist of cells, and a cell in every multicellular organism has a cell nucleus. This nucleus contains the deoxyribonucleic acid (DNA), the hereditary material. DNA is made out of two long, twisted strands that contain complementary genetic information. A *gene* is a segment of DNA that is passed down from parents to children and confers a trait to the off-

spring. Genes are organized and packaged in units called "chromosomes". A *chromosome* is a DNA molecule with part or all of the genetic material (genome) of an organism. A *genome* is the name for all genetic material that is characteristically present in one organism. Therefore, all the different chromosomes of an organism together make up that organism's genome. Figure 2.1 illustrate the relationship of DNA, genes, and chromosomes, where DNA is neatly packed into chromosomes. A *telomere* is a region of repetitive nucleotide sequences at each end of a chromosome, which protects the end of the chromosome. A *centromere* is a part of a chromosome that links sister chromatids or a dyad, where a chromatid is one of two identical halves of a replicated chromosome. *Histones* are highly alkaline proteins found in eukaryotic cell nuclei that package and order the DNA into nucleosomes (basic units of DNA packaging in eukaryotes). Humans have 46 chromosomes, fruit flies have 8, and the flour beetle has 20. Most animals are diploid, which means there are two copies of each chromosome. For example, human cells have 22 different types of autosome (not a sex chromosome), each present as two copies, and two sex chromosomes. This gives 46 chromosomes in total, 23 originate from the mother and 23 originate from the father. These chromosomes contain all the hereditary information in the form of double stranded DNA [149, 153].

DNA is a molecule that carries the genetic instructions used in the growth, development, functioning and reproduction of all known living organisms and many viruses [156]. The DNA in every nucleus of an organism is the same in all cells. The only exceptions are the sperm cells and the eggs, they only contain half of the DNA that a normal cell contains. For example, sperm and eggs in humans contain only 23 of the 46 chromosomes.

DNA is made up from four different nucleotides (bases). Nucleotides that compose DNA are called deoxyribonucleotides, Figure 2.2 illustrates the structure of nucleotides. The three components of a deoxyribonucleotide are a five-carbon sugar called deoxyribose, a phosphate group, and a nitrogenous base, a nitrogen-containing ring structure that is responsible for complementary base pairing between nucleic acid strands. The carbon atoms of the five-carbon deoxyribose are numbered 1', 2', 3', 4', and 5'. A nucleoside comprises the five-carbon sugar

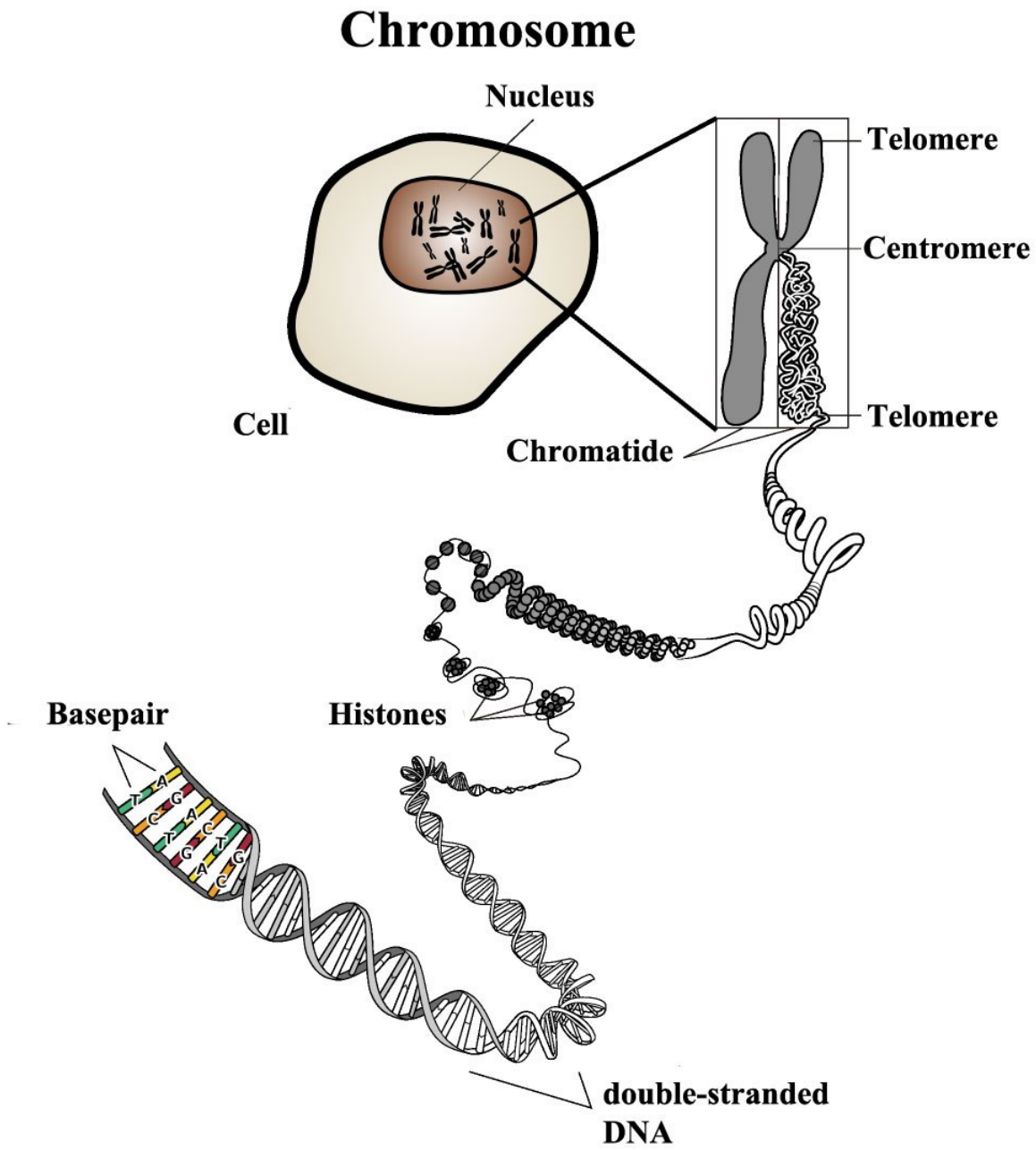

Figure 2.1: DNA, genes, and chromosomes [153].

and nitrogenous base. The deoxyribonucleotide is named according to the nitrogenous bases, which is shown in Figure 2.3. Each nucleotide is composed of one of four nitrogen-containing nucleobases — cytosine (C), guanine (G), adenine (A), or thymine (T) — a sugar called deoxyribose, and a phosphate group. Watson and Crick proposed the double helix model for DNA, which is shown in Figure 2.4. In Figure 2.4 (c), the 5' end is the one where carbon #5 is not bound to another nucleotide; the 3' end is the one where carbon #3 is not bound to another nucleotide. The bases on one strand of DNA form base pairs with a second strand of DNA to form the double helix. But the base pairs that can be formed are limited, according to base pairing rules: adenine (A) can only form a base pair with thymine (T) and guanine (G) can only form a base pair with cytosine (C). Therefore, when we know the sequence of bases on 1 strand of DNA, we also know the sequence of bases on the other strand of DNA. The order of bases is referred to as the sequence. An example of a short sequence of a single strand of DNA is: ATTGCTCAT. Because we know the sequence of this strand we also know which bases are on the other strand: TAACGAGTA [150, 153].

Modern DNA sequencing technology such as Sanger sequencing [110], is the process of determining the precise order of nucleotides within a DNA molecule. Therefore, we can know the sequence of all the DNA in all the chromosomes of an organism. Recent advances such as next-generation DNA sequencing (NGS) [111, 87], refers to non-Sanger-based high-throughput DNA sequencing technologies. Nowadays, by using NGS an entire human genome can be sequenced within a single day, although it is not perfect. In contrast, the previous Sanger sequencing technology, used to decipher the human genome, required over a decade to deliver the final draft [9]. For analyzing NGS data, some algorithms and software were developed for genome assembly [54], read correction [91], and short read mapping [31].

Actually DNA is not identical for all individuals of a species, so the human genome that is available online is not identical to an individual genome. But the most important parts of the genome vary considerably less than the less important ones, which is influenced by natural selection. Take for example eye color, it is not important for survival whether you have

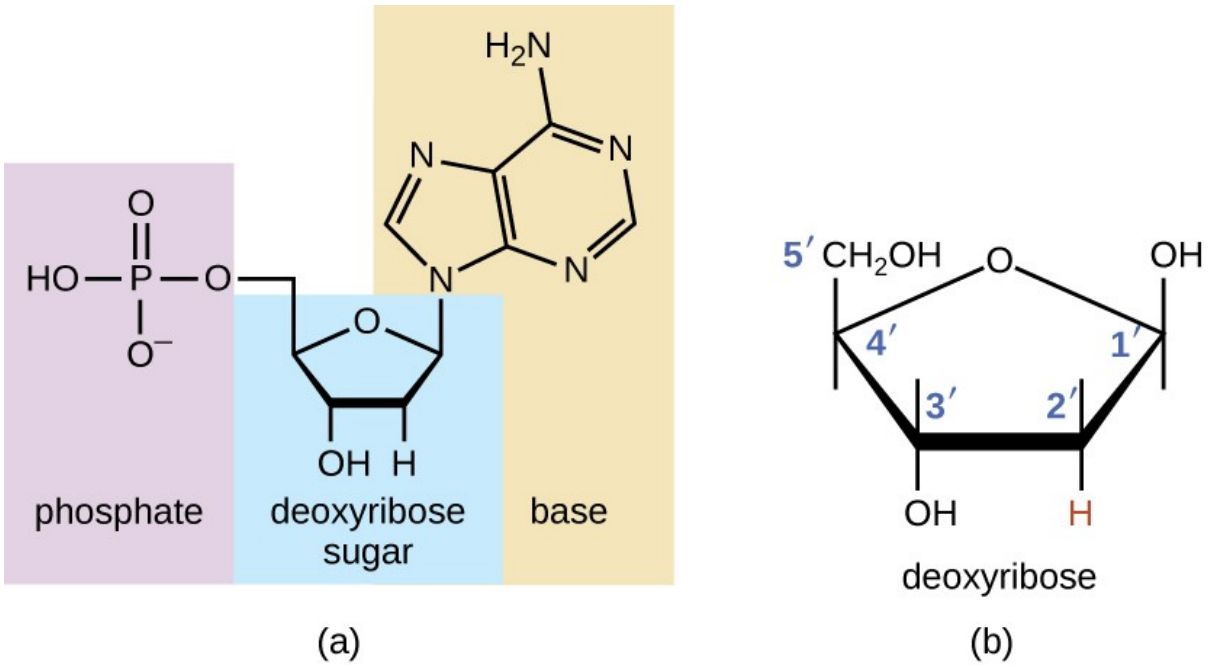

Figure 2.2: The structure of DNA nucleotides [150]. (a) Each deoxyribonucleotide is made up of a sugar called deoxyribose, a phosphate group, and a nitrogenous base – in this case, adenine. (b) The five carbons within deoxyribose are designated as 1', 2', 3', 4', and 5'.

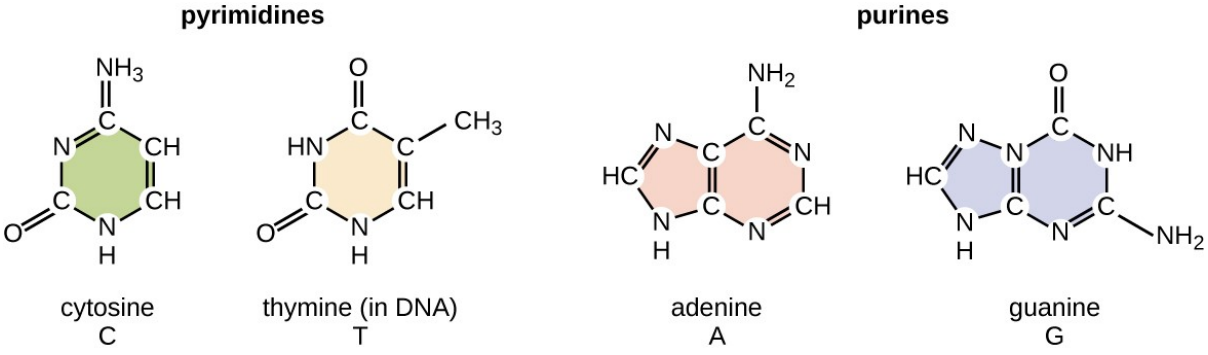

Figure 2.3: The structure of nitrogenous bases [150]. Nitrogenous bases within DNA are categorized into the two-ringed purines adenine and guanine and the single-ringed pyrimidines cytosine and thymine. Thymine is unique to DNA.

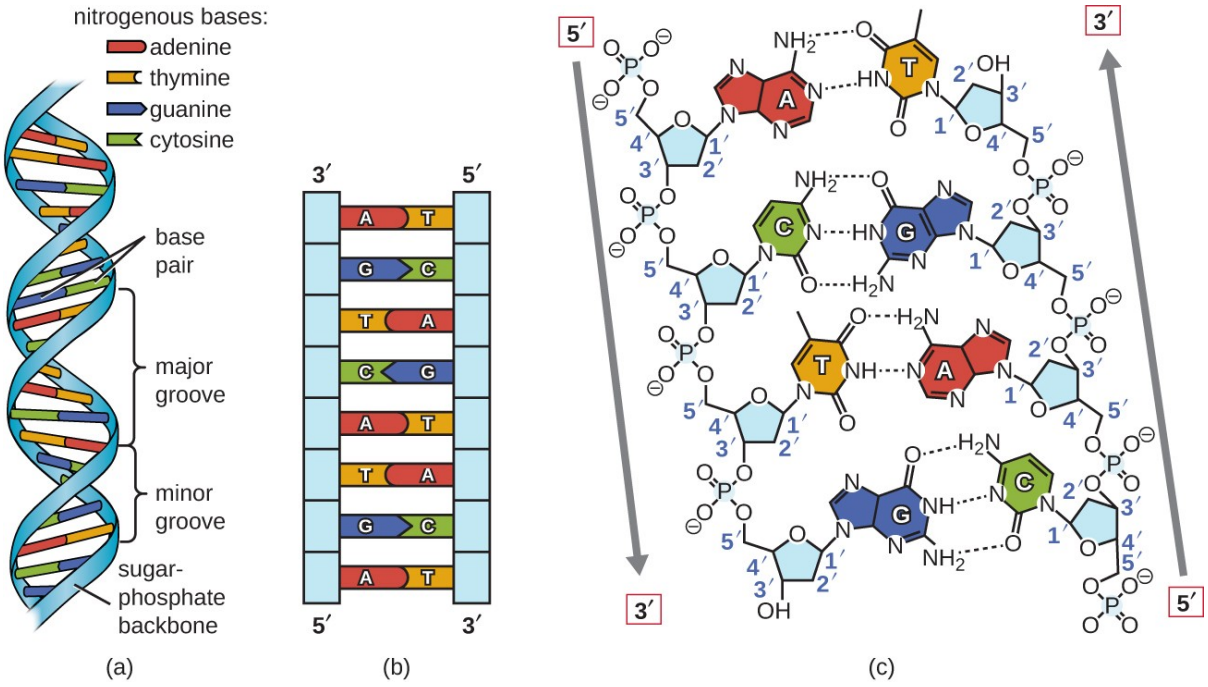

Figure 2.4: Watson and Crick proposed the double helix model for DNA [150]. (a) The sugar-phosphate backbones are on the outside of the double helix and purines and pyrimidines form the "rung" of the DNA helix ladder. (b) The two DNA strands are antiparallel to each other. (c) The direction of each strand is identified by numbering the carbons (1 through 5) in each sugar molecule.

blue or brown eyes, so this is a less important character. The red blood cells that are able to transport oxygen on the other hand are very important; people with red blood cells that are unable to transport oxygen will not survive. Therefore, variation in a character as important as the ability to transport oxygen is sort of “not tolerate”, which means this gene should be stable and identical [153].

2.2 RNA

The central dogma of molecular biology is an explanation of the flow of genetic information within a biological system. It was first stated by Francis Crick in 1958 [28]. To make protein from DNA we first need to code RNA from DNA. Ribonucleic acid (RNA) is a polymeric molecule essential in various biological roles in coding, decoding, regulation, and expression of genes. RNA is synthesized in the nucleus and is very similar to DNA. The synthesis of RNA also involves the use of bases, but in RNA synthesis no thymine (T) is used but uracil (U) is used instead [153]. The sequence of RNA corresponds to the sequence of DNA from which the RNA is synthesized, which is shown in Figure 2.5, where the central dogma is often stated as“DNA makes RNA and RNA makes protein”.

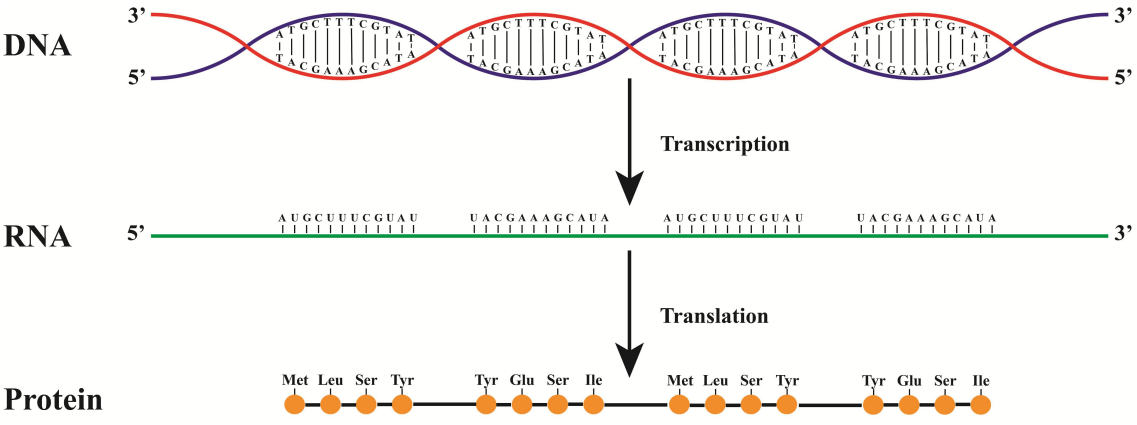

Figure 2.5: Central dogma [153].

The synthesis of RNA from DNA is called *transcription*. In Figure 2.5 the RNA is being

synthesized from the red strand of DNA, which serves as template. This strand of DNA starts with the base T, and the RNA strand starts with the only base that can form a base pair with this T, the A. This continues until the complete sequence of RNA is synthesized. Since the red strand serves as template, the sequence of RNA will be identical to the blue strand of DNA, only with the base U instead of the base T. Therefore, now we have a RNA strand. After transcription the RNA is relocated to the cytoplasm of the cell, and the protein will be synthesized from it. This is called *translation*, which means RNA is translated into protein [153].

Recall that, a gene is a segment of DNA which is transcribed into *messenger RNA (mRNA)* and then translated into a protein. Messenger RNA is a large family of RNA molecules that convey genetic information from DNA to the ribosome. Genes consist of different regions, include untranslated regions (UTRs) on both ends (5' - end and the 3' - end), exons and introns. Figure 2.6 shows this situation. The UTRs are not translated into protein. These regions are however involved in determining the activity of the gene. The exons are the only parts of the gene that can be translated into protein, the introns are spliced out of the mRNA before it leaves the nucleus [153].

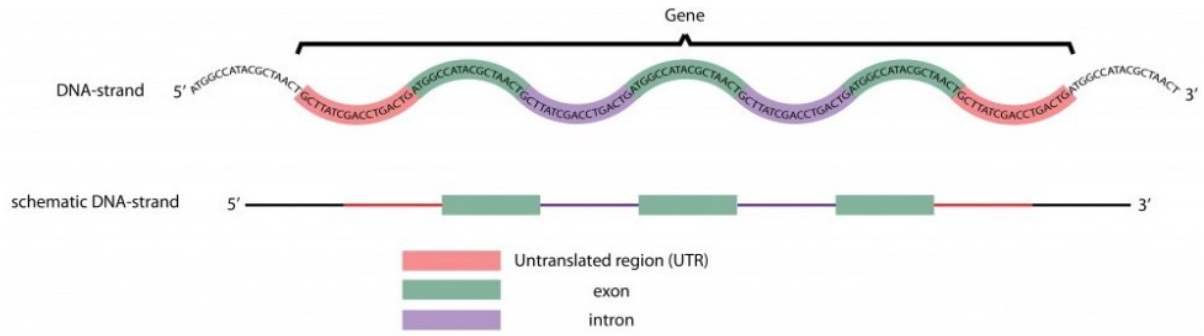

Figure 2.6: Gene, untranslated region, exon, and intron [153].

When genes are "transcribed", it means that the sequences of A,C,G and T's are being copied into RNA molecules with the same sequences but a U instead of a T. This process does not discriminate between the different regions (UTR, exon and intron) of the gene, it simply

copies. After copying a gene, the resulting RNA sequence is called a pre-mRNA. This is because it still contains the introns, which are not used to make protein. These are removed before the mRNA leaves the nucleus in a process called *splicing*, which is illustrated in Figure 2.7. Splicing is performed by the spliceosome, which is a complex of different RNA molecules (small nuclear RNA or snRNA) and several proteins. It simply recognizes the boundaries of exons and introns, folds up the intron, cuts it out and connects the 2 exons together. Next, a 5' cap is added to the 5' end of the mRNA, this increases the stability of the molecule once it leaves the nucleus. Finally, a string of A's is added to the 3' end of the mRNA, which is called the poly-A tail that is between 100 and 250 residues long. The poly-A tail can increase the stability of the mRNA and prevents its degradation. Additionally, the poly-A tail allows the mature mRNA molecule to be exported from the nucleus and translated into a protein [153].

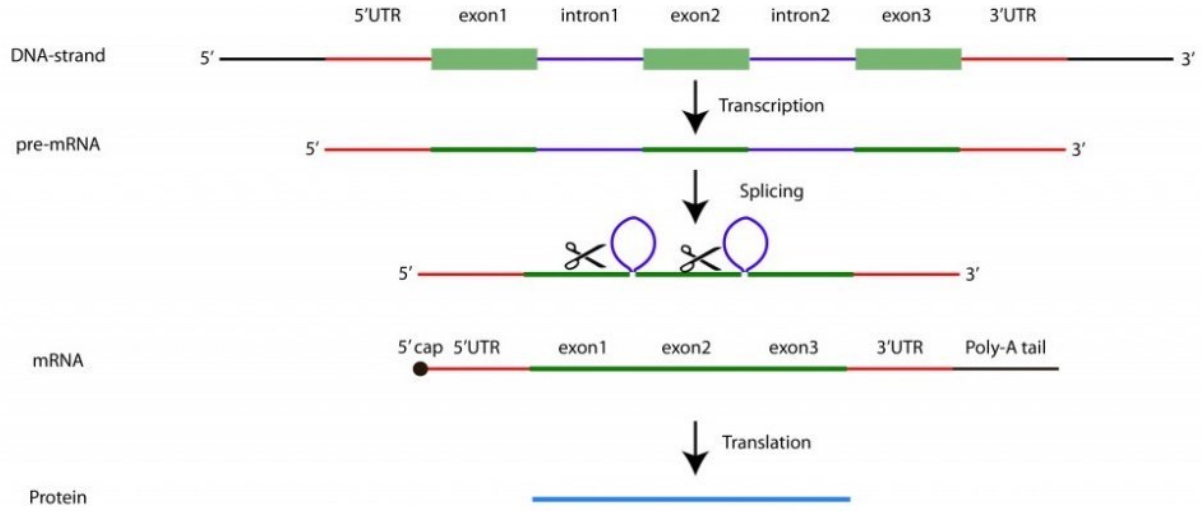

Figure 2.7: Splicing [153].

In fact, it is possible to make different proteins by including or leaving out certain exons of genes. This process is called *alternative splicing*, and it can produce different proteins from the same gene. These different protein versions from the same gene are called *isoforms*. In this way, our bodies can produce over 100,000 proteins from only 20,000 genes. Here the principle is the same as *splicing*. But the only difference is, instead of just splicing the introns out, some exons might also be spliced out. This leads to different proteins in Figure 2.8, one in which

all 3 exons are used and one in which only exon 1 and 3 are used. Therefore, one gene can produce two proteins [153].

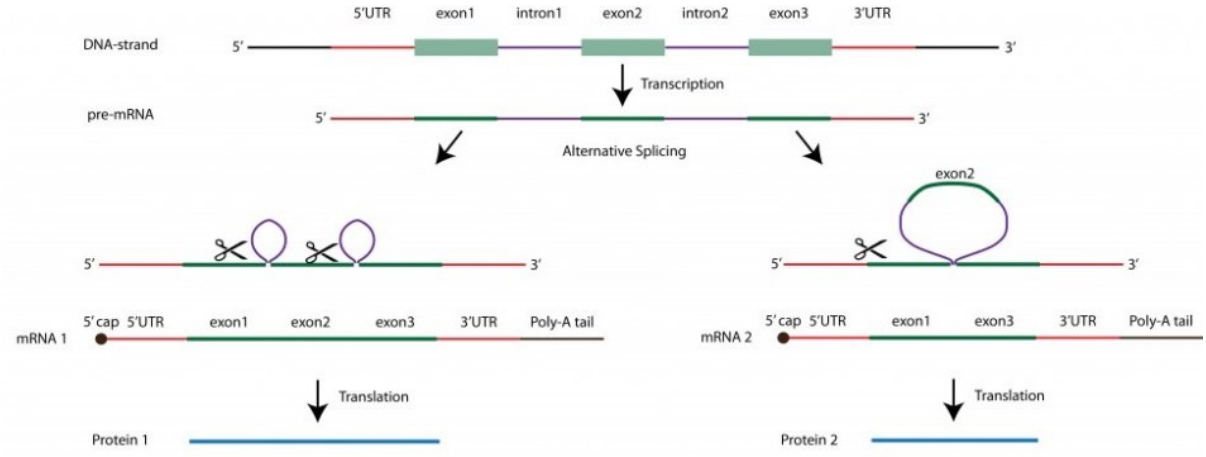

Figure 2.8: Alternative splicing [153].

Taking the human genome as example, less than 2% of the DNA sequence comprises protein-coding exons. The rest of the genome is non-coding and was previously regarded as junk DNA. However, recent studies have unveiled that these regions contain cis-regulatory elements (CREs), such as promoters, enhancers, silencers, insulators, etc. CREs are regions of non-coding DNA which regulate the transcription of nearby genes. CREs are vital components of genetic regulatory networks, which in turn control morphogenesis, the development of anatomy, and other aspects of developmental biology [156]. A genome-wide CRE discovery method based on promoter sequences and gene co-expression networks was introduced in [43], also some machine learning approaches include deep learning were developed for predicting CREs [66].

RNA is quite similar to DNA in structure. However, RNA molecules are much shorter and are typically single stranded, whereas DNA molecules are typically long and double stranded. RNA is made of ribonucleotides. A ribonucleotide contains ribose (the pentose sugar), one of the four nitrogenous bases (A, U, G, and C), and a phosphate group. Figure 2.9 shows the structures of deoxyribose and ribose. Besides the double helix of DNA, the subtle structural difference between the sugars gives DNA added stability, making DNA more suitable for stor-

age of genetic information, whereas the relative instability of RNA makes it more suitable for its more short-term functions. RNA molecules perform a variety of roles in the cell but are mainly involved in the process of translation and its regulation. Even though RNA is single stranded, most types of RNA molecules show extensive intramolecular base pairing between complementary sequences within the RNA strand, creating a predictable three-dimensional structure essential for their function. Figure 2.10 shows the difference between DNA and RNA [150].

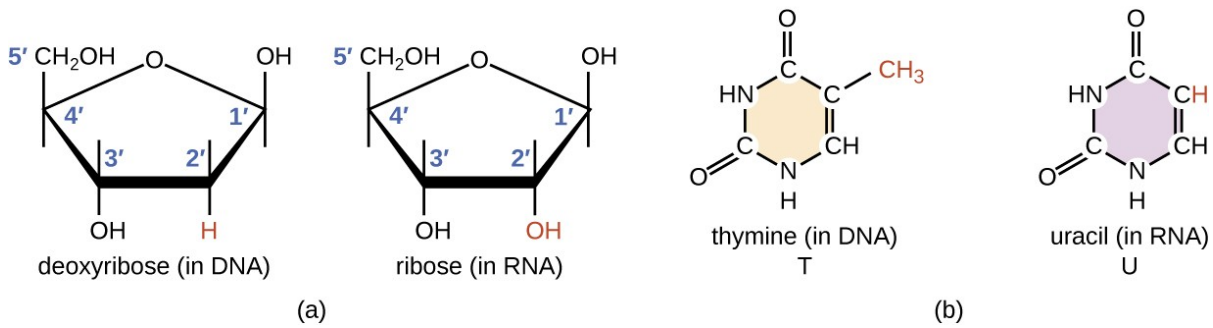

Figure 2.9: The difference between DNA and RNA [150]. (a) Ribonucleotides contain the pentose sugar ribose instead of the deoxyribose found in deoxyribonucleotides. (b) RNA contains the pyrimidine uracil in place of thymine found in DNA.

The three main types of RNA directly involved in protein synthesis are *mRNA, ribosomal RNA (rRNA)*, and *transfer RNA (tRNA)*. In 1961, French scientists Francois Jacob and Jacques Monod hypothesized the existence of an intermediary between DNA and its protein products, which they called messenger RNA [108]. Evidence supporting their hypothesis was gathered soon afterwards showing that information from DNA is transmitted to the ribosome for protein synthesis using mRNA. The mRNA carries the message from the DNA, If a cell requires a certain protein to be synthesized, the gene for this product is “turned on” and the mRNA is synthesized through the process of transcription. In protein synthesis, rRNA and tRNA are stable types of RNA. In prokaryotes and eukaryotes, tRNA and rRNA are encoded in the DNA, then copied into long RNA molecules that are cut to release smaller fragments containing the individual mature RNA species. Ribosomal RNA is the second type of RNA and a major con-

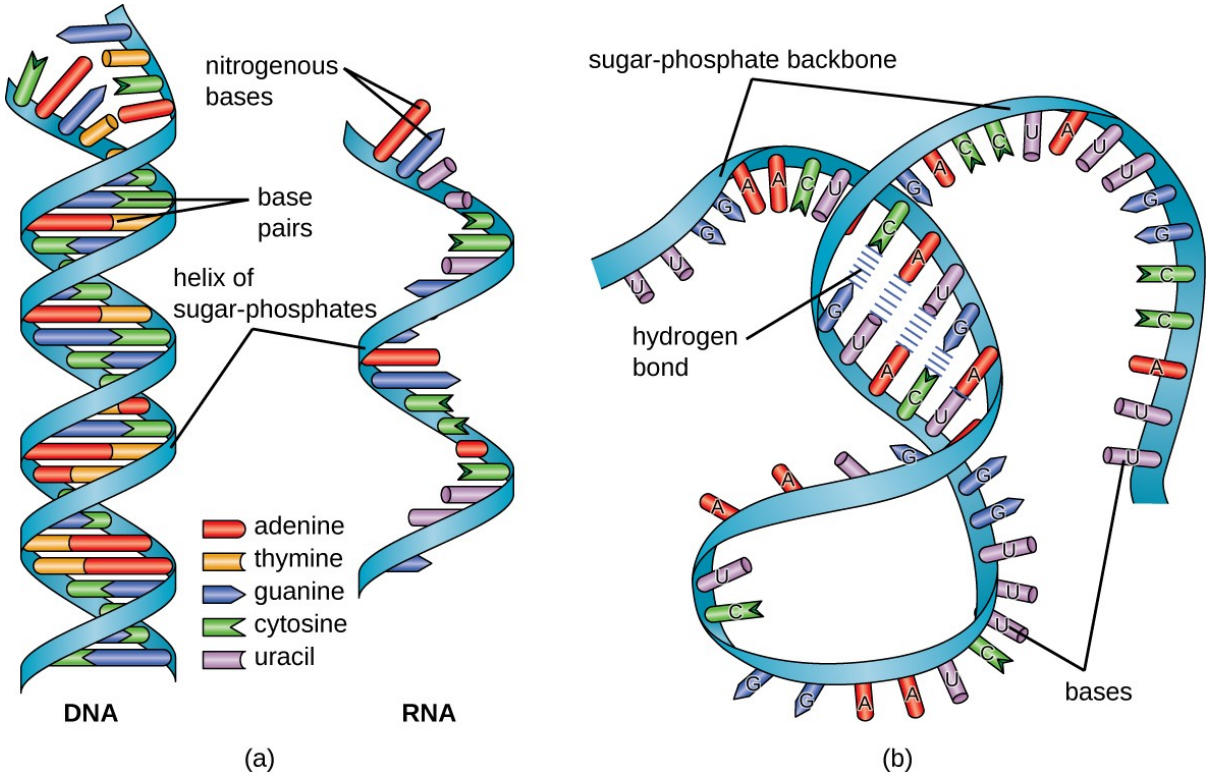

Figure 2.10: The structures of DNA and RNA [150]. (a) DNA is typically double stranded, whereas RNA is typically single stranded. (b) RNA can fold upon itself, with the folds stabilized by short areas of complementary base pairing within the molecule, forming a three-dimensional structure.

stituent of ribosomes, composing up to about 60% of the ribosome by mass and providing the location where the mRNA binds. The rRNA ensures the proper alignment of the mRNA, tRNA, and the ribosomes; the rRNA of the ribosome also has an enzymatic activity and catalyzes the formation of the peptide bonds between two aligned amino acids during protein synthesis [98]. Figure 2.11 shows how mRNA and tRNA are used in protein synthesis. Here, a ribosome consists of two major components: the small ribosomal subunit, which reads the mRNA; and the large subunit, which joins amino acids to form a polypeptide chain. Transfer RNA is the third main type of RNA and one of the smallest, usually only 70-90 nucleotides long. It carries the correct amino acid to the site of protein synthesis in the ribosome. The 2-dimensional and 3-dimensional structure of tRNA is shown in Figure 2.12. It is the base pairing between the tRNA and mRNA that allows for the correct amino acid to be inserted in the polypeptide chain being synthesized [150].

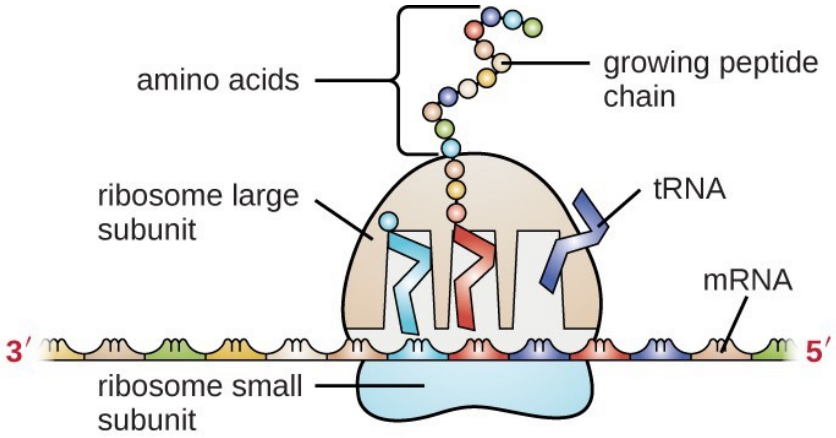

Figure 2.11: A generalized illustration of how mRNA and tRNA are used in protein synthesis within a cell [150].

RNA molecules' secondary structures (2-dimensional structure of RNA) could be represented as ordered labelled trees, and Figure 2.13 shows an example of the RNA GI:2347024 structure. Ordered labelled trees are trees where each node has a label and the left-to-right order among siblings is significant. An ordered labelled forest is a sequence of ordered labelled trees. Algorithms for the edit distance between two forests (trees) [135, 34] could be used to measure

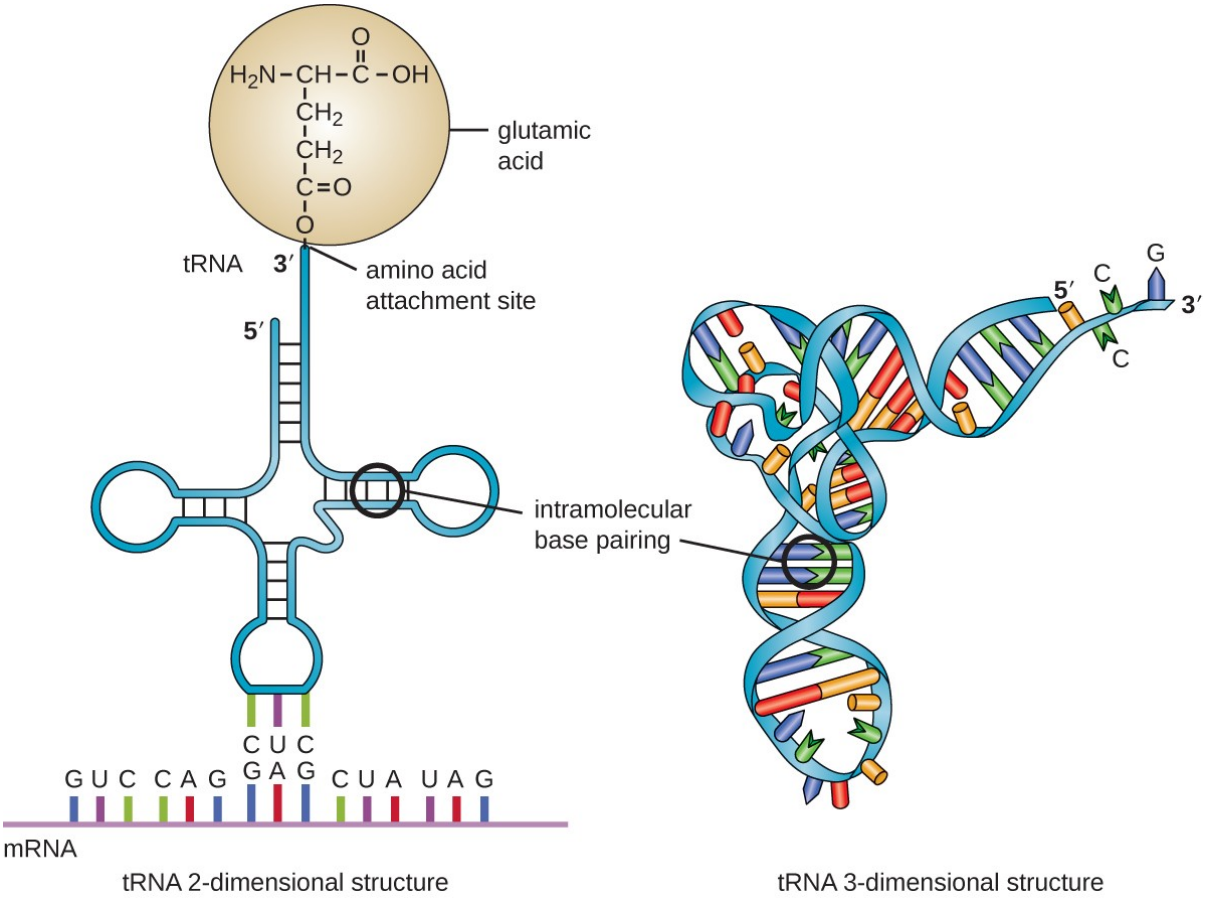

Figure 2.12: The 2-dimensional and 3-dimensional structure of tRNA [150]. A tRNA molecule is a single-stranded molecule that exhibits significant intracellular base pairing, giving it its characteristic three-dimensional shape.

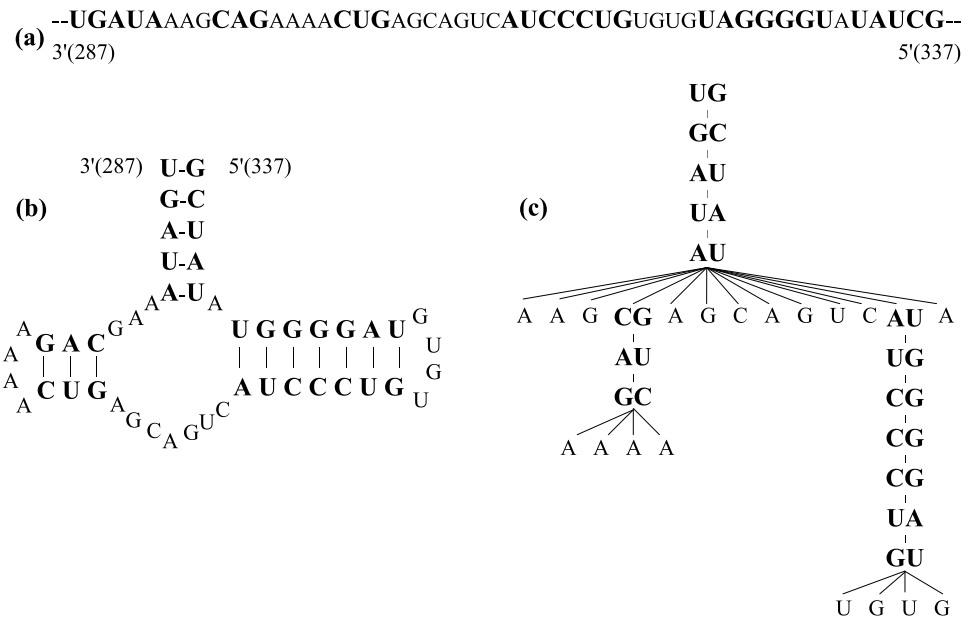

Figure 2.13: RNA structures and forest representation [102]. (a) A segment of the RNA GI: 2347024 primary structure [151], (b) its secondary structure, (c) its forest representation.

the global similarity of forests (trees). Recently the *Forest (tree) Pattern Matching* problem and the *Local Forest (tree) Similarity* problem attracted much attention [102, 57, 136, 68, 71, 72].

2.3 Amino Acids, Peptides, and Proteins

Proteins, come from the Greek *proteios*, meaning "primary" or "holding first place". They are a class of organic compounds which perform a vast array of functions within every living cell. Proteins such as skin, hair, and muscles, can hold together, protect, and provide structure to the body of a multi-celled organism. In a form of enzymes, hormones, and antibodies, they also catalyze, regulate, and protect the body chemistry. A protein is a polymer of amino acids, linked by amide bonds. The length and specific amino acid sequence of a protein are major determinants of its shape, and the shape of a protein is critical to its function. The primary protein structure is simply the sequence of amino acids that make up the polypeptide chain. Figure 2.14 depicts the primary structure of a protein.

The translation of RNA to protein is different than the synthesis of RNA from DNA. When

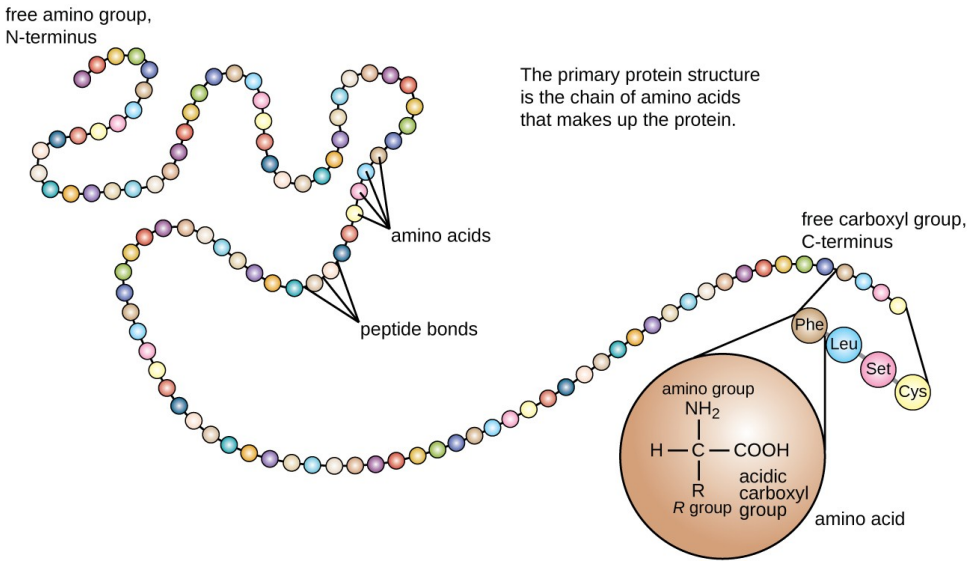

Figure 2.14: The primary structure of a protein is the sequence of amino acids [150].

the DNA was transcribed into RNA, one base of DNA corresponded to one base of RNA, but this one to one relation is not used in the translation to protein. During the translation, an amino acid is added to the protein strand for every three bases in the RNA. Therefore, a RNA sequence of 48 bases codes for a protein strand of 16 amino acids. A certain combination of three bases always gives the same amino acids, and the three-nucleotide code means that there is a total of 64 possible combinations, which is shown in Figure 2.15. This number is greater than the number of amino acids, and a given amino acid is encoded by more than one codon (a sequence of three DNA or RNA nucleotides). This redundancy in the genetic code is called degeneracy. We take the first three bases from Figure 2.5 as example, which are AUG. The first base is A, we look it up on the left side of the table, which is the 3^{rd} row of the table. The second base is U, we look it up on the top of the table, which is the 1^{st} column and 3^{rd} row. There we see our third base and our combination of AUG codes for the amino acid Methionine (Met or M). In this way we can translate the complete RNA sequence into the protein sequence [153]. Whereas 61 of the 64 possible triplets code for amino acids, three of the 64 codons do not code for an amino acid; they terminate protein synthesis, releasing the polypeptide from the translation machinery. These are called stop codons or nonsense codons. Another

codon, AUG, typically serves as the start codon to initiate translation, and it also specifies the amino acid Methionine (Met or M). Then the *reading frame*, the way nucleotides in mRNA are grouped into codons, for translation is set by the AUG start codon near the 5' end of the mRNA. Each set of three nucleotides following this start codon is a codon in the mRNA message [150]. From above description, DNA codons can be decoded unambiguously into amino acids, but it is not possible to predict a specific DNA codon from an amino acid. The reason is that there are 61 different DNA (and mRNA) codons specifying only 20 amino acids.

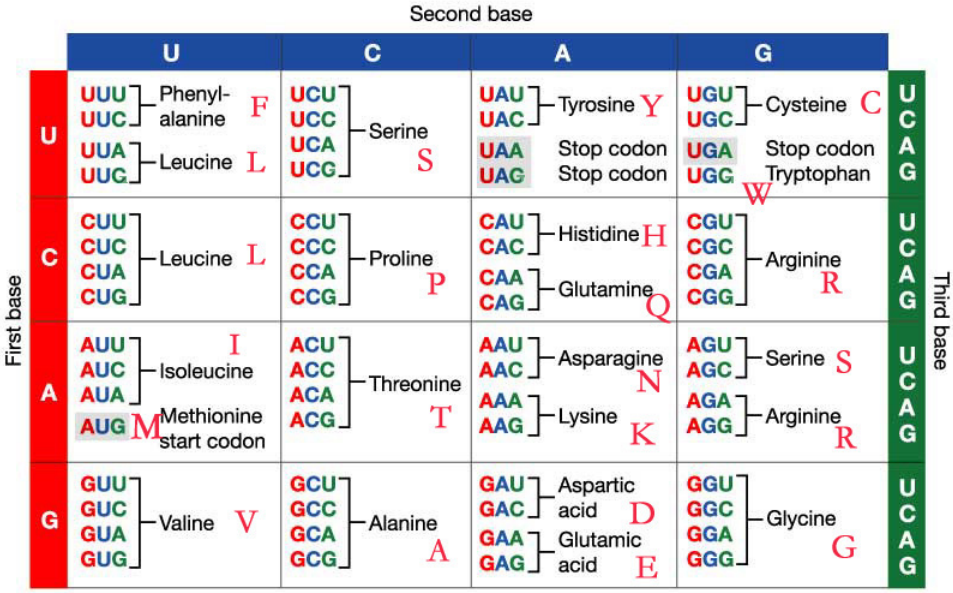

Figure 2.15: Messenger RNA codons and amino acids [144].

As the name implies, *amino acids* are organic molecules where a hydrogen atom, an amino group (-NH_2), and a carboxyl group (-COOH), are all bonded to the same carbon atom, the so-called α carbon. The fourth group bonded to the α carbon varies among the different amino acids and is called a *residue* or a *side chain*, represented in structural formulas by the letter R. A residue is a monomer that results when two or more amino acids combine and remove water molecules. The primary structure of a protein is a peptide chain, which is made of amino

acid residues. The unique characteristics of the functional groups and R groups allow these components of the amino acids to form hydrogen, ionic, and disulfide bonds, along with polar/nonpolar interactions needed to form secondary, tertiary, and quaternary protein structures. These groups are composed primarily of carbon, hydrogen, oxygen, nitrogen, and sulfur, in the form of hydrocarbons, acids, amides, alcohols, and amines. Figure 2.16 provides a few examples illustrating these possibilities. Amino acids may chemically bond together by reaction of the carboxylic acid group of one molecule with the amine group of another [150]. This reaction forms a peptide bond and a water molecule, which is another example of dehydration synthesis. Figure 2.17 illustrates the formation of a peptide bond, where a molecule of water is released in this process. The 20 amino acid residue abbreviations, masses, and their compositions are shown in Table 2.1. For example, amino acid residue alanine (Ala or A) has monoisotopic mass of 71.03711 and average mass of 71.08, with the composition of C_3H_5NO.

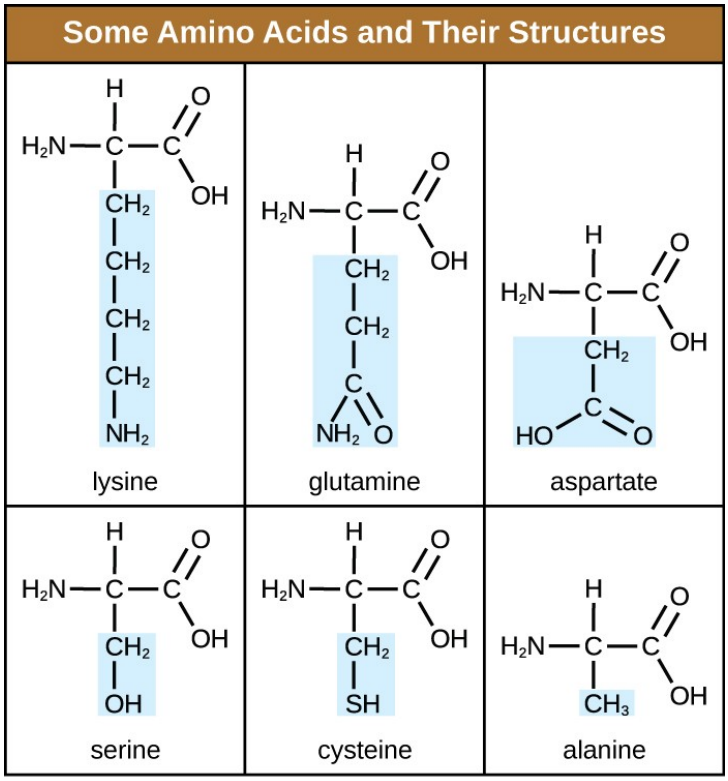

Figure 2.16: Some amino acid and their structures [150].

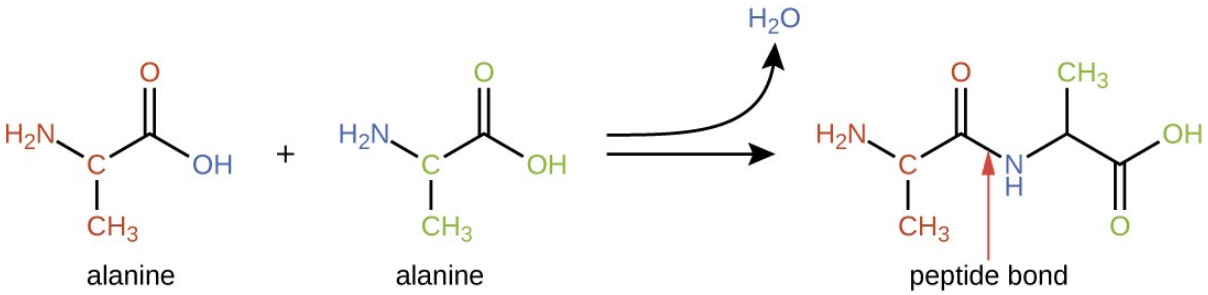

Figure 2.17: Peptide bond formation is a dehydration synthesis reaction [150]. The carboxyl group of the first amino acid (alanine) is linked to the amino group of the incoming second amino acid (alanine).

A *polypeptide* is a linear chain of *amino acid residues*. Proteins are composed of one or more polypeptides. In other words, peptides are "short" proteins. Short polypeptides, containing less than 20-30 residues, are rarely considered to be proteins and are commonly called peptides, or sometimes oligopeptides. The individual amino acid residues are bonded together by peptide bonds and adjacent amino acid residues. The sequence of amino acid residues in a protein is defined by the sequence of a gene, which is encoded in the genetic code. In general, the genetic code specifies 20 standard amino acids. Shortly after or even during synthesis, the residues in a protein are often chemically modified by post-translational modification (PTM), which alters the physical and chemical properties, folding, stability, activity, and ultimately, the function of the proteins [156].

In reality complex interactions between amino acids lead to three-dimensional forms, which are essential for the functioning of the protein. Protein structure is the three-dimensional arrangement of atoms in a protein molecule. Protein structure can be categorized in terms of four levels: primary, secondary, tertiary, and quaternary. Figure 2.18 describes them. The summaries below are based on definitions found in [115].

- *Primary structure* — the linear sequence of residues (amino acids) in a polypeptide chain.

- *Secondary structure* — the arrangement of a polypeptide chain into more or less regular

Table 2.1: The 20 amino acid residue abbreviations, masses, and compositions [146].

Name	3-letter code	1-letter code	Monoisotopic Mass	Average Mass	Composition
Alanine	Ala	A	71.03711	71.08	C_3H_5NO
Arginine	Arg	R	156.10111	156.2	$C_6H_{12}N_4O$
Asparagine	Asn	N	114.04293	114.1	$C_4H_6N_2O_2$
Aspartic Acid	Asp	D	115.02694	115.1	$C_4H_5NO_3$
Cysteine	Cys	C	103.00919	103.1	C_3H_5NOS
Glutamic Acid	Glu	E	129.04259	129.1	$C_5H_7NO_3$
Glutamine	Gln	Q	128.05858	128.1	$C_5H_8N_2O_2$
Glycine	Gly	G	57.02146	57.05	C_2H_3NO
Histidine	His	H	137.05891	137.1	$C_6H_7N_3O$
Isoleucine	Ile	I	113.08406	113.2	$C_6H_{11}NO$
Leucine	Leu	L	113.08406	113.2	$C_6H_{11}NO$
Lysine	Lys	K	128.09496	128.2	$C_6H_{12}N_2O$
Methionine	Met	M	131.04049	131.2	C_5H_9NOS
Phenylalanine	Phe	F	147.06841	147.2	C_9H_9NO
Proline	Pro	P	97.05276	97.12	C_5H_7NO
Serine	Ser	S	87.03203	87.08	$C_3H_5NO_2$
Threonine	Thr	T	101.04768	101.1	$C_4H_7NO_2$
Tryptophan	Trp	W	186.07931	186.2	$C_{11}H_{10}N_2O$
Tyrosine	Tyr	Y	163.06333	163.2	$C_9H_9NO_2$
Valine	Val	V	99.06841	99.13	C_5H_9NO

hydrogen-bonded structures, which has two basic elements:

  - *Alpha helix* — spiral configuration of a polypeptide chain with 3.6 residues (amino acids) per turn. The helix may be left-handed or right-handed, and the latter is more common.
  - *Beta strand* —two adjacent polypeptide strands that are bonded together. Two or more strands may interact to form a *beta sheet*, which is a common motif of regular secondary structure.

- *Tertiary structure* — the level of protein structure at which an entire polypeptide chain has folded into a three-dimensional structure. In multi-chain proteins, the term tertiary structure applies to the individual chains.

- *Quaternary structure* — the fourth order of complexity of structural organization exhibited by protein molecules, and refers to the arrangement in space of the complete protein, without regard to the internal geometry of the subunits. Quatenary structure is possessed only when the molecule is made of at least two subunits that are separable.

The three-dimensional structure of a protein is determined by techniques such as X-ray crystallography and nuclear magnetic resonance (NMR).

In fact, proteins rarely act alone. Instead, most biological characteristics arise from complex interactions among the cell's numerous constituents, such as proteins, DNA, RNA, and small molecules. Many times proteins team up into "molecular machines" and have intricate physicochemical dynamic connections to undertake biological functions at both cellular and systems levels [7]. Protein-protein interactions (PPIs) are the physical contacts of high specificity established between two or more protein molecules as a result of biochemical events steered by electrostatic forces including the hydrophobic effect [156]. Many are physical contacts with molecular associations between chains that occur in a cell or in a living organism in a specific biomolecular context [33, 123]. A protein interaction network in yeast is shown in Figure 2.19, which illustrates that a few highly connected nodes hold the network together. In this figure, the largest cluster that contains ~78% of all proteins is shown. Many computational methods can predict protein-protein interactions, such as SPRINT, which is a new sequence-based algorithm and tool [67].

2.4 Mass Spectrometry Instruments

Mass spectrometry (MS) is an analytical technique that ionizes chemical species and sorts the ions based on their mass-to-charge ratio. In other words, a mass spectrum measures the masses within a sample. MS is used in many different fields and is applied to pure samples as well as complex mixtures [156]. Here, we will introduce the MS instruments, including mass spectrometers, tandem mass spectrometers, and mass spectrometer configurations.

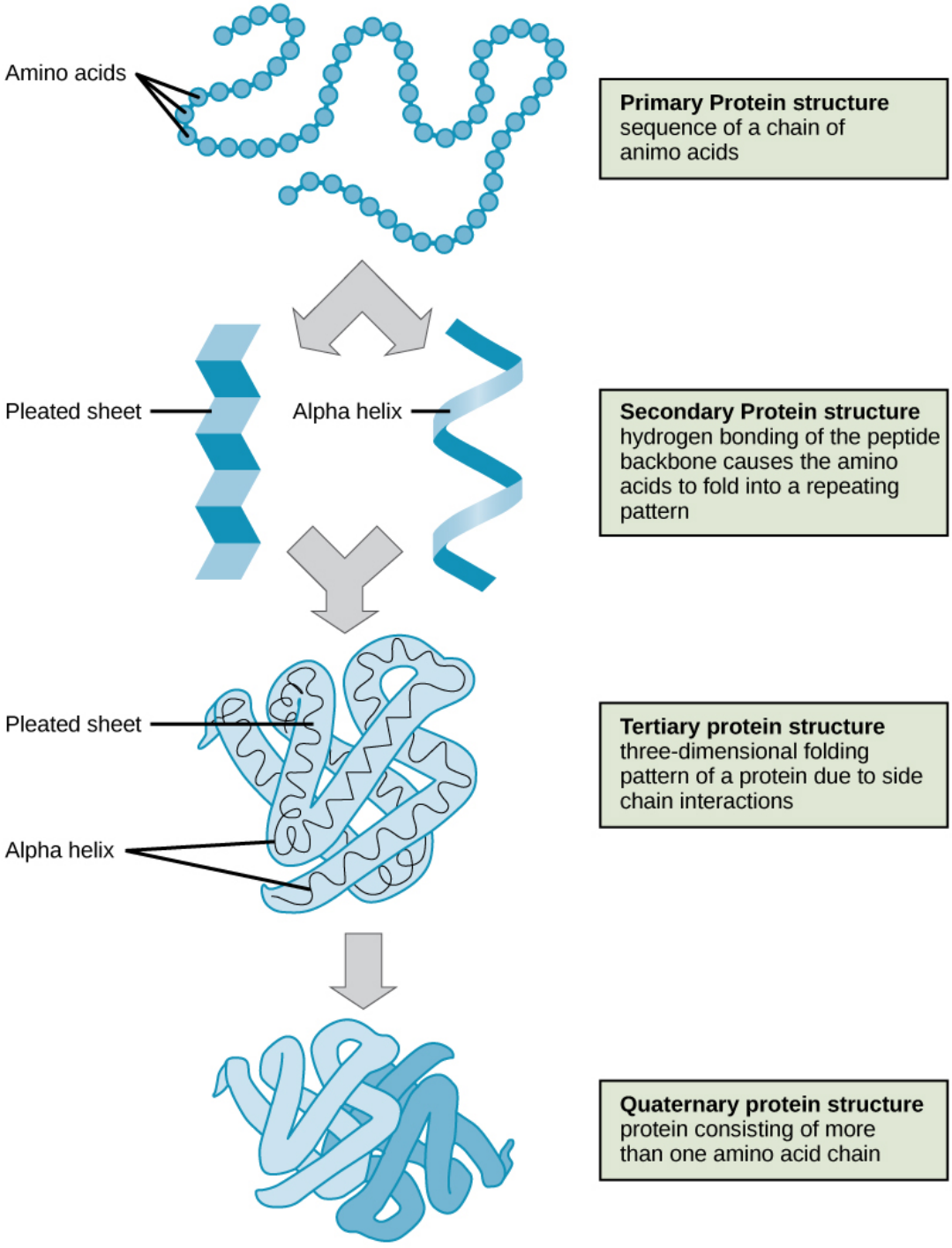

Figure 2.18: Three-Dimensional Protein Structures. Image source: National Human Genome Research Institute (NHGRI).

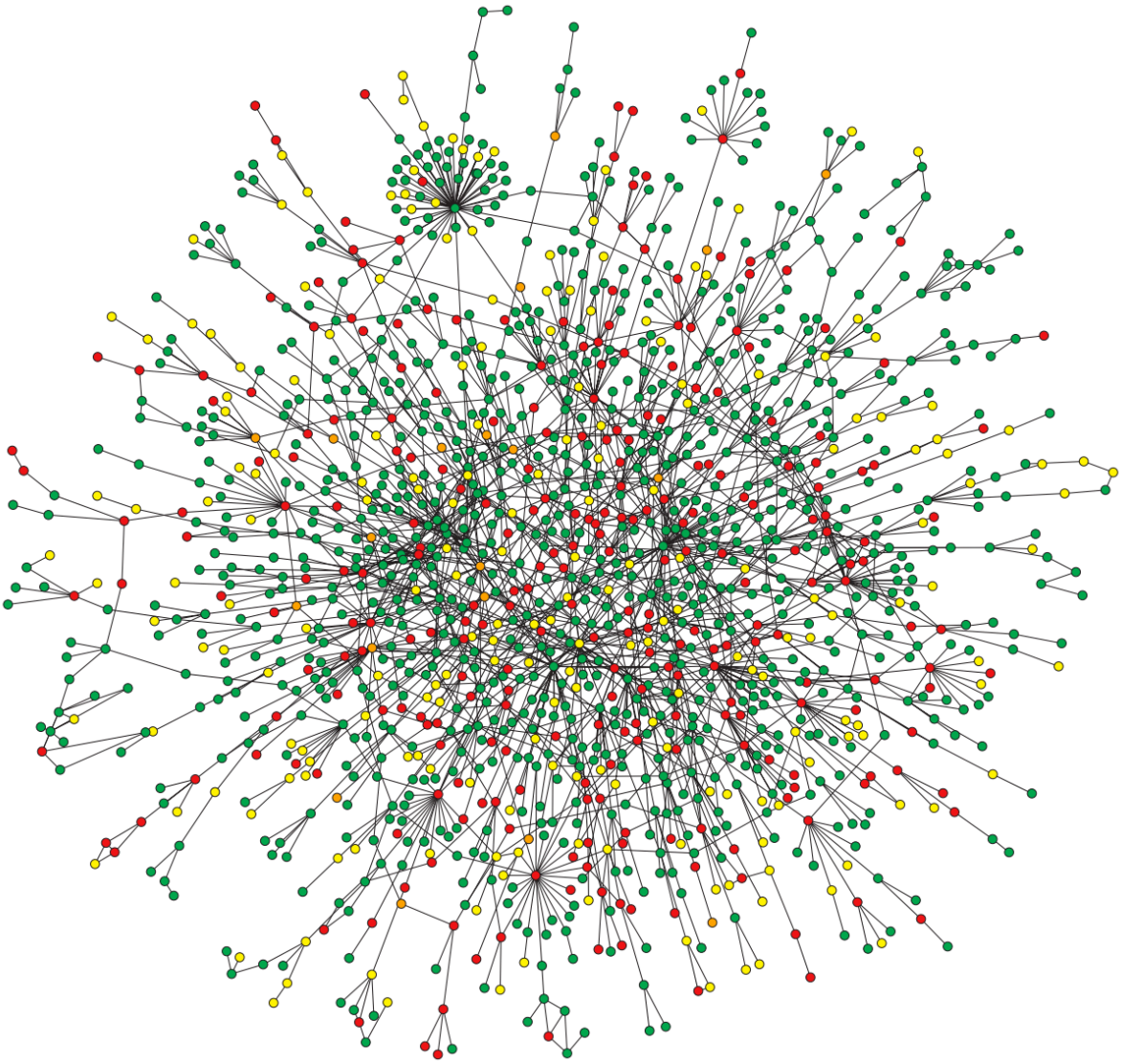

Figure 2.19: Yeast protein interaction network [7]. A map of proteinprotein interactions in Saccharomyces cerevisiae, which is based on early yeast two-hybrid measurements.

2.4.1 Mass Spectrometers

Mass spectrometric measurements are carried out in the gas phase on ionized analytes. By definition, a mass spectrometer consists of an ion source, a mass analyser that measures the mass-to-charge ratio (*m/z*) of the ionized analytes, and a detector that registers the number of ions at each *m/z* value. Figure 2.20 outlines a mass spectrometer, where the mass analyzer and detector are always within the high-vacuum region. A bunch of molecules are first ionized with the ionizer. Several ionization methods exist, but the most commonly used methods in proteomics are electrospray ionization (ESI) and matrix-assisted laser desorption and ionization (MALDI). In the second step, the ions are separated in the mass analyzer according to their *m/z*. Ion sources can be combined with different mass analyzers, such as MALDI-time-of-flight (TOF), ESI-ion trap (IT), and ESI-Fourier transform ion cyclotron resonance (FTICR). The limitation of *m/z* is determined by the mass range of a mass analyzer. The *m/z* range in proteomics can be typically from 50 to 4,000 *u* [48], where *u* is the unified atomic mass unit which was defined as 1/12 the mass of an atom of ^{12}C in 1961. Another unit of mass is dalton (Da), which has been accepted as an alternate name for the unified atomic mass unit [16]. Finally, the ions are detected by the detector and the *m/z* of the detected ions are calculated and stored in a computer. Each type of ions with the same *m/z* will form a peak in the resulting dataset (called a *mass spectrum*). Figure 2.21 shows an example. The intensity of a peak indicates the ion counts detected by the detector at the *m/z*, which is related to the abundance of the corresponding type of molecules in the original sample. However, because different molecules have different ionization efficiencies, the abundances of two different molecules cannot be compared solely by their peak intensities [1, 80, 83].

In mass spectrometry, four terms—*mass resolving power, mass resolution, mass accuracy, and mass precision*—are used to characterize the performance of high-resolution, accurate-mass mass spectrometers. The 2013 IUPAC recommendations [90] clear up the controversial usage of mass resolution and mass resolving power. In 1997 IUPAC, *resolving power in mass spectrometry* is defined as the ability of an instrument to distinguish between two peaks at *m/z*

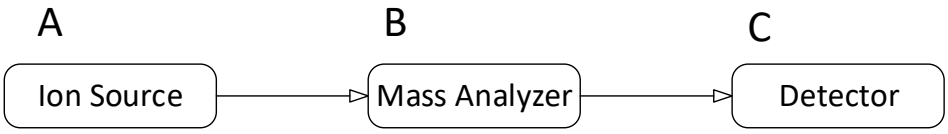

Figure 2.20: Outline of a mass spectrometer [101]. (A) The ion source for electrospray ionization is at atmospheric pressure, and the source for MALDI is under vacuum. (B) The mass analyzer can be a TOF, an ion trap, a quadrupole, a FTICR, or a hybrid of the aforementioned analyzers. (C) The detector is normally an electron multiplier.

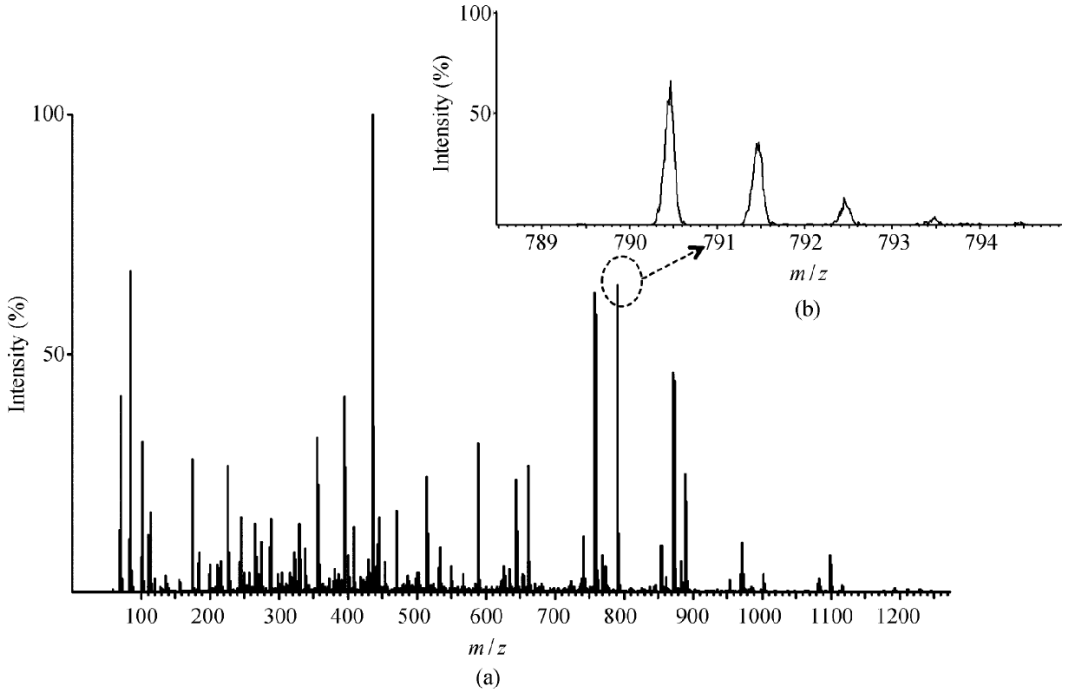

Figure 2.21: Mass spectrum. (a) Exemplary mass spectrum. (b) Zooming in a peak shows more details. In particular, each peak spans a width on the *m/z* direction [80].

values differing by a small amount and expressed as the peak width in mass units (*Deltam*). But *mass resolving power* is defined separately as $m/\Delta m$ in a manner similar to *mass resolution*. These definitions of *resolving power in mass spectrometry* and *mass resolving power* are contradictory, the former is expressed as a mass and the latter as a dimensionless ratio. Consequently, the 2013 IUPAC definition for *mass resolving power* is a "measure of the ability of a mass spectrometer to provide a specified value of mass resolution", while *mass resolution* is defined as the observed *m/z* value divided by the smallest difference $(\Delta)m/z$ for two peaks that can be separated: $(m/z)/(\Delta)m/z$. There are several ways to define the minimum peak separation, and the two most widely used are the valley definition and the peak width definition. Figure 2.22 pictorially illustrates the two different definitions of resolution. In the valley definition, Δm is the closest spacing of two peaks of equal intensity with the valley (lowest value of signal) between them less than a specified fraction of the peak height. Typical valley values are 10% or 50%. The value obtained from a 5% peak width is roughly equivalent to a 10% valley. In the peak width definition, the value of Δm is the width of the peak measured at a specified fraction of the peak height, for example 0.5%, 5%, or 50%. Additionally, the latter is called the full width at half maximum (FWHM), which is usually used in orbitrap technology [156]. *Mass accuracy* is the closeness of the agreement between the result of a measurement and a true value (exact mass). When a measurement is close to the true value we say it is accurate and when it is not we say it is inaccurate. *Mass precision* is the closeness of agreement between independent mass measurement results [155]. When a set of mass measurements of one ion species lie close together we say the measurements are precise, and when not we say the measurements are imprecise. Figure 2.23 illustrates these four terms: (a) Mass resolution $R = m/\Delta m$ at FWHM; (b) *accuracy* is the proximity of the experimental measurement (blue vertical line) to the exact mass (red vertical line); *precision* is the repeatability of the measurement reflecting random errors.

With the above definitions, in an example of a peak at *m/z* 400.0000, with a peak width of 0.002 FWHM, the mass resolution is $R = m/\Delta m = 200{,}000$. As a consequence, two peaks of

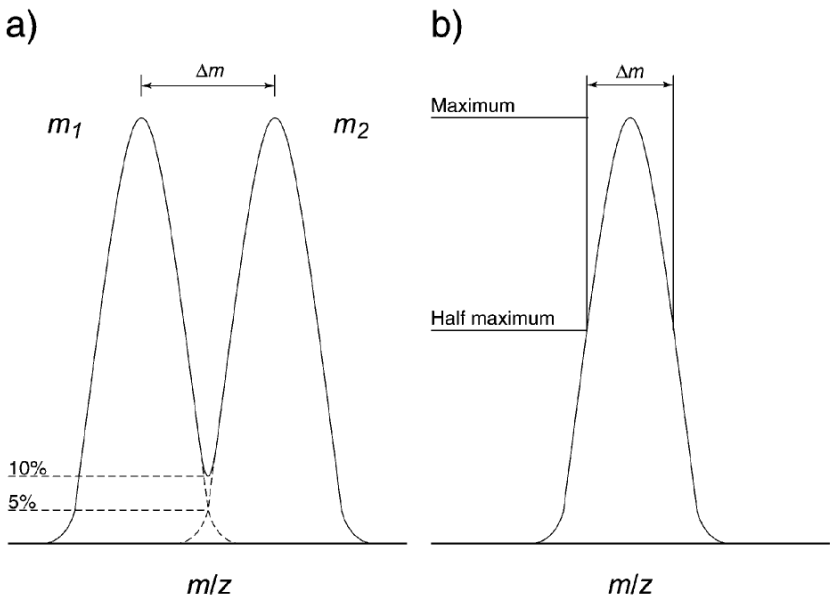

Figure 2.22: The definitions of resolution [8]. (a) "10% valley": two separate peaks of similar intensities overlap at a peak height of 5% of their maxima, therefore summing to be 10% at the overlap. (b) "full width at half maximum": most typically used in conjunction with TOF and FTICR mass spectrometers.

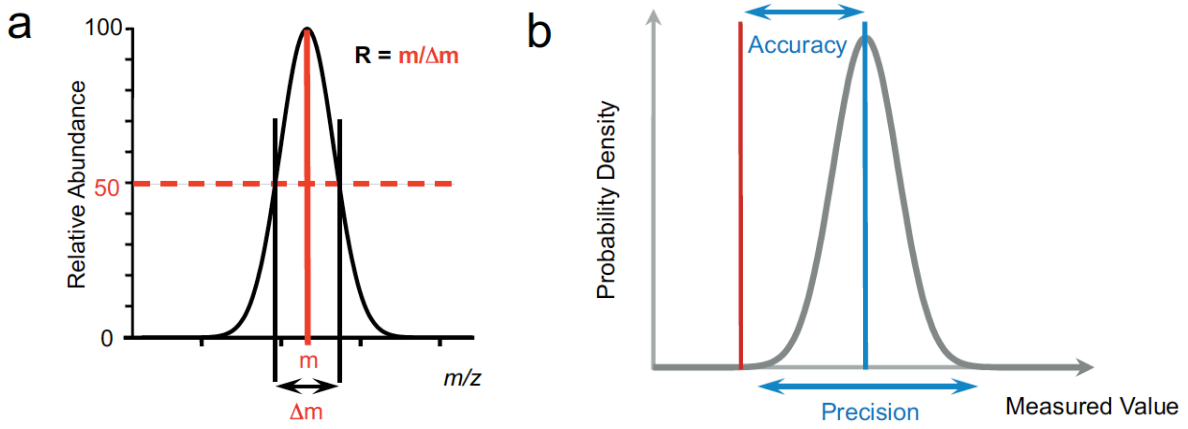

Figure 2.23: Some fundamental MS definitions [155]. (a) Mass resolution $R = m/\Delta m$ at FWHM, (b) accuracy and precision of mass determination, red vertical line is exact mass, and blue vertical line is experimental measurement.

equal height at *m/z* 400.0000 and 400.002 cannot be baseline resolved, which means we cannot distinguish them. Only two peaks of equal height at m/z 400.0000 and 400.004 can be baseline resolved, if the mass spectrometer delivers a peak width of 0.002 u (FWHM) at these given *m/z* values [155]. Mass accuracy is normally measured by *milli mass units (mmu)* or *parts per million (ppm)*. The *mmu* is equivalent to 10^{-3} u. A more modern name for the *mmu* is the *milli dalton (mDa)*, since nowadays the unified atomic mass unit is more and more displaced by the unit *dalton*. Consequently, the *ppm* is equivalent to 10^{-6}. Then the mass accuracy can be measured in *Da*, *mDa*, and *ppm* as

$$\begin{aligned}\Delta m_i &= (m_i - m_a)\ in\ Da \\ &= (m_i - m_a) \times 10^3\ in\ mDa \\ &= ((m_i - m_a)/m_a) \times 10^6\ in\ ppm,\end{aligned}$$

where m_a is the exact or theoretical mass, m_i is the measured or observed mass from MS, and Δm_i can be positive or negative [13]. Based on the above formulas, for an ion on m_a = 400.0000 Da, the mass accuracy for a measurement on m_i = 400.0013 is $\Delta m_i = 0.0013(Da) = 1.3(mDa) = 3.25(ppm)$. Additionally, if m_i = 399.9987, we will have $\Delta m_i = -0.0013(Da) = -1.3(mDa) = -3.25(ppm)$.

2.4.2 Tandem Mass Spectrometers

A tandem mass spectrometer is needed to obtain tandem mass spectra (or *MS/MS spectra*). A tandem mass spectrometer has two mass analyzers (or two sequential analyses in the same analyzer). The first analyzer selects ions at a certain *m/z* window (usually a very small window so that only copies of the same ion are selected). This selected peptide ion is called the *precursor ion* or the *parent ion*. MS/MS can be one of the two types: tandem-in-space and tandem-in-time.

In tandem-in-space MS, the first mass analyzer separates and isolates the precursor ion of interest. The isolated ion is transmitted to the collision cell where it is fragmented. The

fragments are transmitted and analyzed in the second mass analyzer (tandem in space). In tandem in time, the precursor ion selection, fragmentation, and the separation of the fragments all occur in one mass analyzer at different time-points. The precursor ions with the highest intensity and with a charge state of +2 or more are generally preferred because they give the best fragmentation spectra. The isolated precursor ion is then fragmented by different methods [83].

Finally, the fragment ions are measured as usual to form a *MS/MS spectrum*. The MS/MS spectrum of peptides contains sequence information. Figure 2.24 (a) illustrates the possible fragmentation sites of a peptide. For example, when the fragmentation occurs at the peptide bond (between C and N atoms), The b-ions appear to extend from the amino terminus, sometimes called the N-terminus (for example, b_2), and y-ions appear to extend from the carboxyl terminus (for example, y_2), or C-terminus. The subscript k of the y_k ion indicates the number of residues in the fragment. Fig.2.24 (b) shows an annotated MS/MS spectrum. Because amino acid residues have different mass values (except for Leucine and Isoleucine), the m/z values of the fragment ions can be used to identify peptides [80].

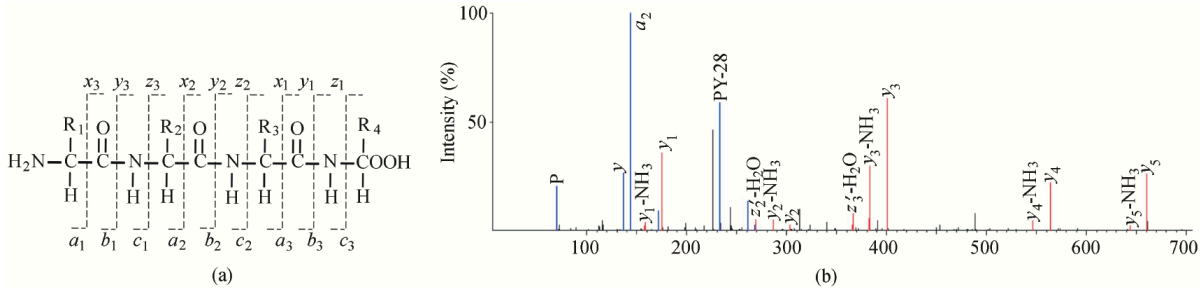

Figure 2.24: Tandem mass spectrum [80]. (a) Fragmentation of a four-residue peptide in MS/MS. The fragmentation can happen at each bond on the peptide backbone, resulting in different fragment ion types. (b) Annotated CID MS/MS spectrum of a peptide GLPYPQR. CID produces mostly y and b ions.

2.4.3 Mass Spectrometer Configurations

Each of the three aforementioned components (the ionizer, the mass analyzer and the detector) of a mass spectrometer can be made with different technologies, causing different properties of the data. In proteomics MS/MS dataset analysis, one cares mostly about the ionizer type, the mass analyzer type, and the peptide fragmentation method.

Two types of ionizers are commonly used in proteomics. These are matrix-assisted laser desorption/ionization (MALDI) and electrospray ionization (ESI). The difference for the ions produced is that MALDI produces singly charged ions ($z = 1$) and ESI produces singly and multiply charged ions ($z \geq 1$). ESI ionizes the analytes out of a solution and is therefore readily coupled to liquid-based (for example, chromatographic and electrophoretic) separation tools. Figure 2.25 illustrates generic MS-based proteomics experiment. MALDI sublimates and ionizes the samples out of a dry, crystalline matrix via laser pulses. Integrated liquid-chromatography ESI-MS systems (LC-MS) are preferred for the analysis of complex samples, whereas MALDI-MS is normally used to analyse relatively simple peptide mixtures [1, 80].

Five basic types of mass analysers are commonly used in proteomics research. These are quadrupole (Q), ion trap (linear ion trap, LIT/LTQ; quadrupole ion trap, QIT), time-of-flight (TOF), Fourier transform ion cyclotron (FTICR), and orbitrap analysers. The difference in mass analyzers mostly affects the resolution and mass accuracy of the data. Normally the order of performance in terms of resolution and accuracy is ion trap $\approx$ quadrupole < TOF < FTICR $\approx$ orbitrap. Table 2.2 summarizes their analytical characteristics and capabilities [48]. These analysers can be stand alone or, in some cases, put together in tandem to take advantage of the strengths of each other [1, 80]. Figure 2.26 illustrates several mass spectrometers used in proteome research.

In our wet-lab experiments, we use Thermo Scientific Orbitrap Elite mass spectrometer, which combines a dual-pressure linear ion trap with a high-field Orbitrap mass analyzer, to create the ultimate analytical instrument. The schematic of the Orbitrap Elite Hybrid MS is demonstrated in Figure 2.27, where the mass resolution is >240,000.

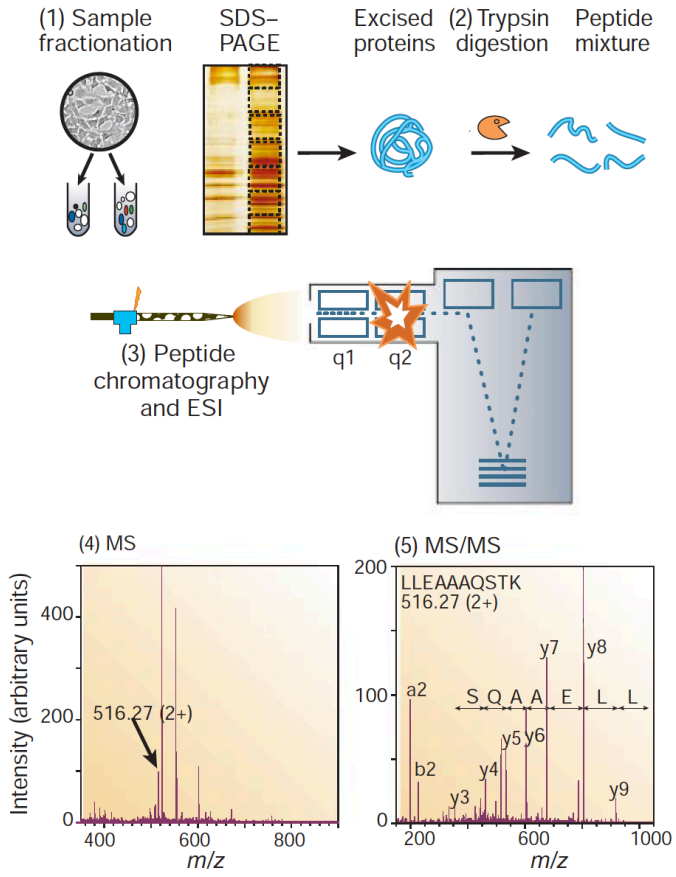

Figure 2.25: Generic MS-based proteomics experiment [1]. The typical proteomics experiment consists of five stages.

Table 2.2: The comparison of typical performance characteristics of commonly used mass spectrometers in proteomics

Instrument	Mass resolution	Mass accuracy (ppm)	*m/z* range	Scan rate
QIT	1,000	100-1,000	50-2,000; 200-4,000	Moderate
LTQ	2,000	100-500	50-2,000; 200-4,000	Fast
TOF	10,000-20,000	10-20; <5	No upper limit	Fast
FTICR	50,000-750,000	<2	50-2,000; 200-4,000	Slow
LTQ-Orbitrap	30,000-100,000	<5	50-2,000; 200-4,000	Fast

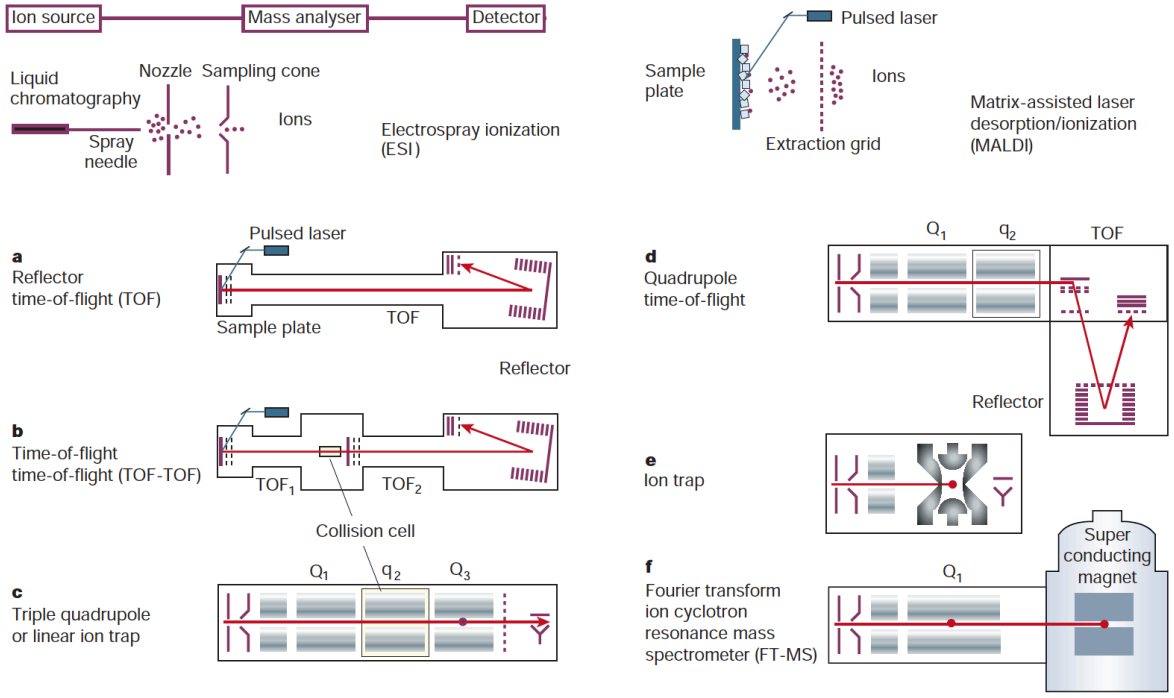

Figure 2.26: Mass spectrometers used in proteome research [1]. The left and right upper panels depict the ionization and sample introduction process in ESI and MALDI. The different instrumental configurations ($a \rightarrow f$) are shown with their typical ion source.

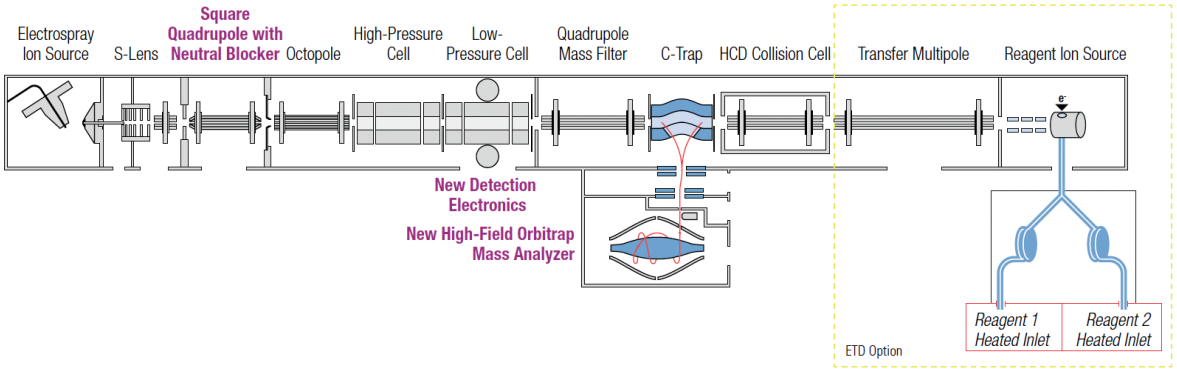

Figure 2.27: Schematic of the Orbitrap Elite Hybrid MS [145].

There are a few fragmentation methods for tandem mass spectrometers: collision induced dissociation (CID), collision activated dissociation (CAD), infrared multiphoton dissociation (IRMPD), blackbody infrared dissociation (BIRD), sustained off resonance irradiation collision induced dissociation (SORI-CID), electron capture dissociation (ECD), surfaceinduced dissociation (SID), electron transfer dissociation (ETD), and higher-energy C-trap dissociation (HCD). CID was traditional most used for fragmentation, while a more recent type is HCD. These approaches tend to fragment at different sites of a peptide, and often generate different types of fragment ions and therefore significantly different spectra for the same peptide [80].

One cannot be ignorant to the characteristics of the different detectors in analyzing MS data. For example, detector saturation effects (the state reached when an increase in applied external magnetic field cannot increase the magnetization of the material further) can skew the near Gaussian mass peaks of intense ions downward, so the experimental mass of intense ions will have negative mass error. An overview of the different instrument techniques that can be used is summarized in Table 2.3 [83], where the instrumental methods can be combined in many different ways.

Table 2.3: Overview of different instrument settings

Ion sources	Mass analyzers	Fragmentation		Detectors
MALDI	Quardrupole	CID/CAD		Faraday cup
ESI	TOF	ECD	Bottom up	Scintillation counter
ESSI	Ion trap	ETD		Electron multiplier
SELDI	FTICR	HCD		HED
FAB	Orbitrap	IRMPD	Top down	MCP
PD		BIRD		FTICR
LDI		SID		

2.5 Protein Identification by Mass Spectrometry

Protein identification is a key and essential step in the field of proteomics. Mass spectrometry has emerged as the primary tool for protein identification, which is by far the most popular

application of mass spectrometry in proteomics today [19].

There are usually two different methods to identify proteins: top-down and bottom-up. For top-down sequencing, intact proteins are fragmented directly into the mass spectrometer. For bottom-up technologies, protein identification is based on mass spectrometric analysis of peptides derived from proteolytic digestion, usually with trypsin.

In the bottom-up approaches for tandem mass spectra, there are two modes for protein identification: *data-dependent acquisition* (DDA) and *data-independent acquisition* (DIA). The most widespread mode is DDA, where selected peptide precursors following chromatographic separation are fragmented by MS/MS [74]. DIA is a method of molecular structure determination where all ions within a selected *m/z* range are fragmented and analyzed in a second stage of tandem MS [156]. Tandem mass spectra are acquired either by fragmenting all ions that enter the mass spectrometer at a given time (called broadband DIA) or by sequentially isolating and fragmenting at the selected ranges of *m/z*.

In this section, we will review several different approaches for protein identification: top-down MS, peptide mass fingerprinting and bottom-up tandem mass spectra. For the tandem mass spectra in DDA mode, there are roughly four ways to interpret the dataset and identify the fragmentation of proteins: sequence database searching, spectral library searching, database-independent approach (*de novo* sequencing), and the hybrid interpretation algorithms.

2.5.1 Top-Down Protein Identification

Top-down mass spectrometry is an emerging approach for the analysis of intact proteins. Top-down MS offers the ability to sequence intact proteins, especially for the analysis of PTMs, but is not yet a high-throughput method [114]. In the top-down approach, intact proteins are introduced into the mass spectrometer, so important information about combinatorial PTMs (individual histone PTMs and their combinatorial patterns) is retained. A key step in enabling top down approaches has been the ability to assign tandem mass spectrometer product ion identities, which can be done either via high resolving power or through product ion charge

state manipulation [107].

The work flows of top-down and bottom-up approaches for protein analysis are shown in Figure 2.28. The mass range of top-down has been extended to proteins as large as 229 kDa [49], and one can detect increasingly larger numbers of intact proteins in a single analysis. But top-down technique is still mainly for analyzing single purified proteins or simple mixtures, while bottom-up methods are used to the analysis of complex mixtures and proteins. Although the bottom-up approach offers swift identification of a protein by combining tandem MS dataset and database searching, there are some drawbacks for it. For example, one cannot completely recover peptides from protein digestion for MS measurements, leading to information loss. Top-down sequencing can overcome these information-loss problems. The isoforms can be separated by the spectrometer, because measurement of the intact proteins gives information for possible protein isoforms provided they have sufficiently different masses. Given that the analyst has selected the protein of interest according to its *m/z* prior to activation, as a MS/MS method, top-down sequencing always makes the connection between the intact protein and the fragments [29].

2.5.2 Protein Identification by Peptide Mass Fingerprinting

Peptide mass fingerprinting is an effective way of identifying, e.g., gel-separated or LC-MS, proteins, by matching experimentally obtained peptide mass dataset against large databases. The experimental dataset used in the PMF-based protein identification strategy are mass lists derived from matrix-assisted laser desorption/ionization time-of-flight (MALDI-TOF) MS spectrum of an enzymatic-digested protein. Also, theoretical spectra each comprising the list of masses expected by an enzymatic digestion of each protein sequence in the reference database are provided [22]. Additionally, experimental dataset are used as query in PMF search programs. Consequently, each protein in the database is given a score, dependent on how well the theoretical mass list correlates with the experimental data. Those with the highest score are then most likely to be present in the sample in question. This sounds like a simple task if

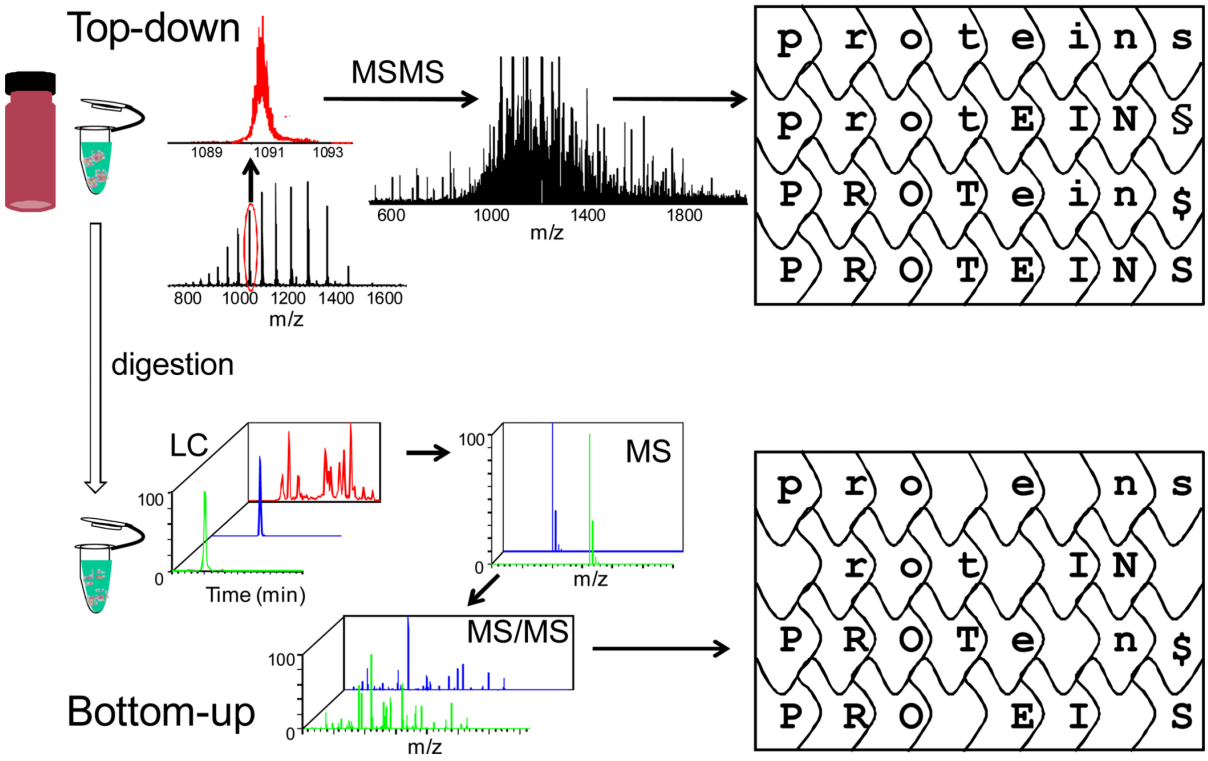

Figure 2.28: The comparison of top-down and bottom-up workflows [29].

the protein in question is present in the database. However, several phenomena influence the matching of the experimental dataset to the theoretical ones, making PMF a method that is not always reliable. This can happen for many reasons, such as unexpected PTM, splice variants, individual sequence variants (single nucleotide polymorphisms, etc), or omissions and errors in the database [86]. In addition, only peptides with a mass within the recorded mass range will be observed, e.g., between 700 and 3,500. Also, additional signals can arise from unwanted components in the sample, e.g., keratin from hair and dust. Figure 2.29 lists additional reasons why a complete match of all peaks should not be expected [83].

Various software tools can be used for PMF searches [103, 24, 137, 125]. The outcome is most often a ranked list of proteins of which the top hit represents the protein/proteins most likely to be present in the samples analyzed. Many of the algorithms behind these search tools take into account features like the size of the database, the distribution frequency of a particular peptide mass within a given protein size, and the distribution of the mass accuracy. The user

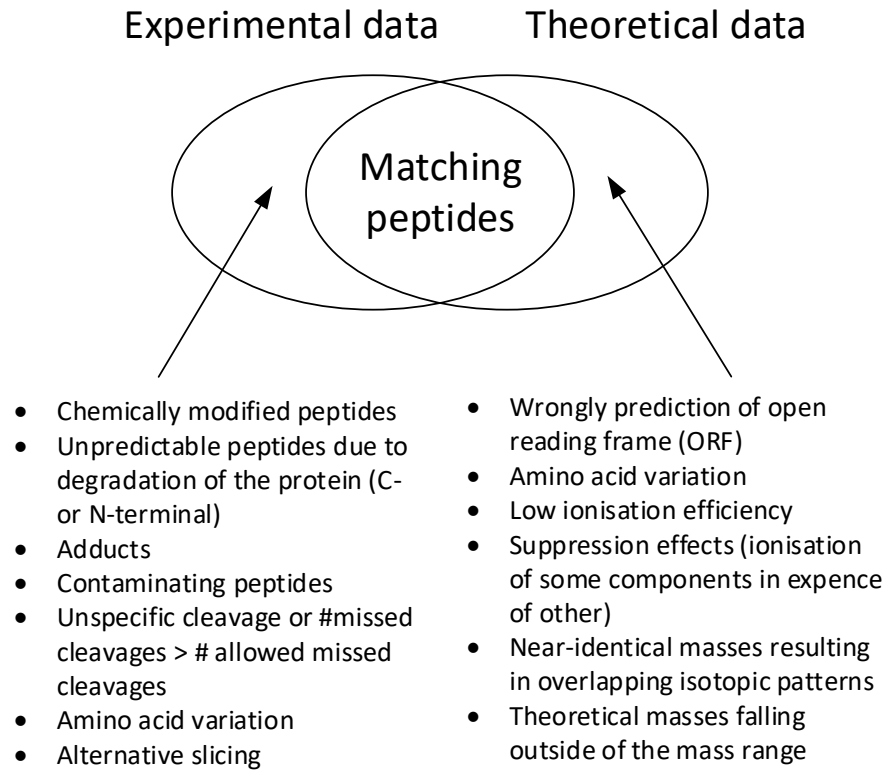

Figure 2.29: The Venn diagram of matching peptides [83]. There are several reasons why a perfect match should not be expected, when experimental peptide mass fingerprinting dataset is compared to the theoretically predicted dataset.

can also set some parameters, such as the number of missed cleavages sites allowed (there may be some sites that should be cleaved by the reagent that are not completely cleaved during the course of the experiment), the modifications expected, the sequence database against which the dataset should be searched, and the mass accuracy. The specificity of the search is determined by these parameters. For example, if the mass tolerance is two times higher than the true mass accuracy of the data, then the specificity of the search is low, and the risk of having false-positive hits is high. On the other hand, if the mass tolerance is two times lower than the true mass accuracy of the data, then the specificity of the search is high, and the search program will not be able to report the true-positive hit [83]. The next step is to decide on a threshold: proteins whose scores are above this threshold are said to be identified. The choice of a suitable threshold is difficult because setting a threshold too low will incorrectly identify a number of proteins (false positives), while setting the threshold too high may result in the correct protein not being identified (false negative) [86].

One of the limitations of PMF is its sensitivity to database size. There is a direct effect on the statistical confidence a PMF algorithm can ascribe to protein identification as the search database grows. A larger database has an elevated chance of the experimental masses randomly matching theoretical peptide masses in these databases, thereby decreasing the confidence of protein identifications using PMF [86].

2.5.3 Protein Identification by Tandem Mass Spectrometry

In the last few years, the shotgun (bottom-up) proteomics method [74, 47, 128, 6, 106] has become the approach of choice for identifying and quantifying proteins in most large-scale studies [1]. This strategy is based on digesting proteins into peptides followed by peptide sequencing using tandem mass spectrometry (MS/MS) and automated database searching. The shotgun method is also called peptide fragment fingerprinting (PFF). Compared with methods of analysis based on extensive protein separation prior to MS-based identification, such as two-dimensional (2D) gels [46], LC-MS/MS (shotgun) proteomics allows higher dataset throughput

and better protein detection sensitivity, and it is presently the method of choice for identifying proteins in most large-scale studies. Our research is also based on shotgun proteomics, which involves several major steps schematically illustrated in Figure 2.30.

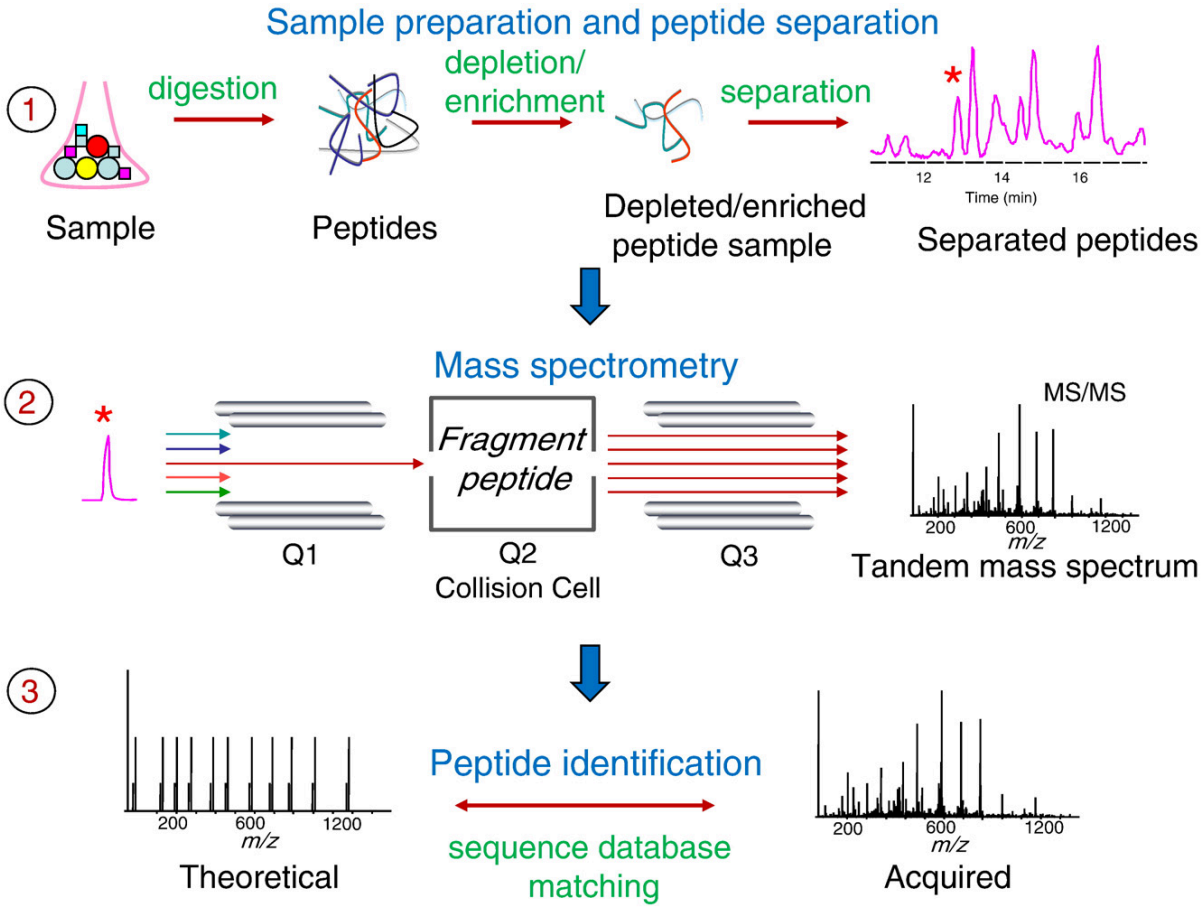

Figure 2.30: Overview of shotgun proteomics [95]. 1) Sample proteins are digested into peptides using enzymes such as trypsin. Resulting peptide mixtures are then separated using a LC system coupled online to a mass spectrometer. 2) Peptides are subjected to MS/MS analysis that results in the acquisition of MS/MS spectra. 3) The correct assignment of MS/MS spectra to peptide sequences is the first step in proteomic dataset processing.

In the shotgun proteomic approach, the first step is the digestion of sample proteins into peptides using proteolytic enzymes such as trypsin. The resulting peptide mixtures can be very complex, since each protein produces multiple peptides after digestion. Peptide samples are then separated by one- or multidimensional liquid chromatography (LC) and subjected to MS/MS analysis to sequence the peptides [118]. In quantitative proteomics, peptides are also

encoded with a stable isotope tag, which can determinate the relative protein abundances with respect to a control sample [1]. Peptides are then ionized, and selected ions are subjected to sequencing to produce signature MS/MS spectra. The MS/MS dataset acquisition process consists of two stages. The first stage involves reading all peptide ions that are introduced into the instrument at any given time (MS spectrum). In the second stage, some selected peptides called *precursor* or *parent ions* are fragmented into smaller pieces (fragment ions) in the collision cell of the mass spectrometer in the process termed CID. The acquired MS/MS spectrum is thus a record of mass-to-charge ratios (*m/z* values) and intensities of all the resulting fragment ions generated from an isolated precursor ion [23]. One can identify the amino acid sequence of the peptide that produced it, which is from the fragmentation pattern encoded by the MS/MS spectrum. Figure 2.31 demonstrates an example of a tandem mass spectrometry spectrum. After the desired amount of MS dataset is collected, the effort shifts toward the computational analysis [83].

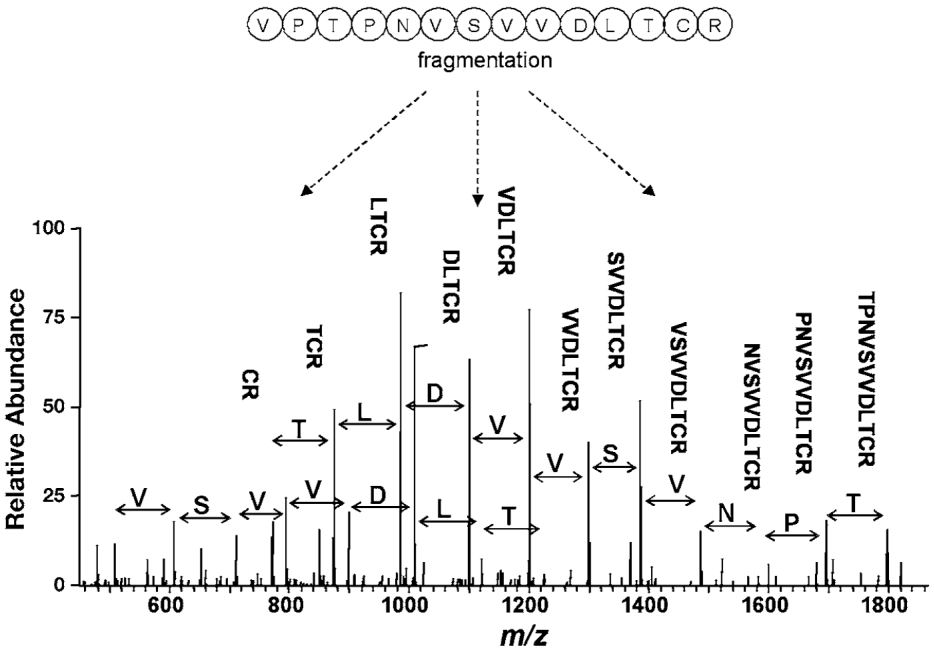

Figure 2.31: An example of a tandem mass spectrometry spectrum [83].

The computational analysis typically starts with the identification of the peptides that give

rise to the acquired MS/MS spectra. The peptide identification strategies can be roughly classified into several categories, which is shown in Figure 2.32. In order to identify peptide, one can correlate acquired experimental MS/MS spectra with theoretical spectra predicted for each peptide contained in a protein sequences database (database search approach), or against spectra from a spectral library (spectral library searching). Alternatively, peptide sequences can be extracted directly from the spectra, i.e., without referring to a sequence database for help (*de novo* sequencing approach). There are also hybrid approaches, such as those based on the extraction of short sequence tags (35 residues long) followed by database searching [95].

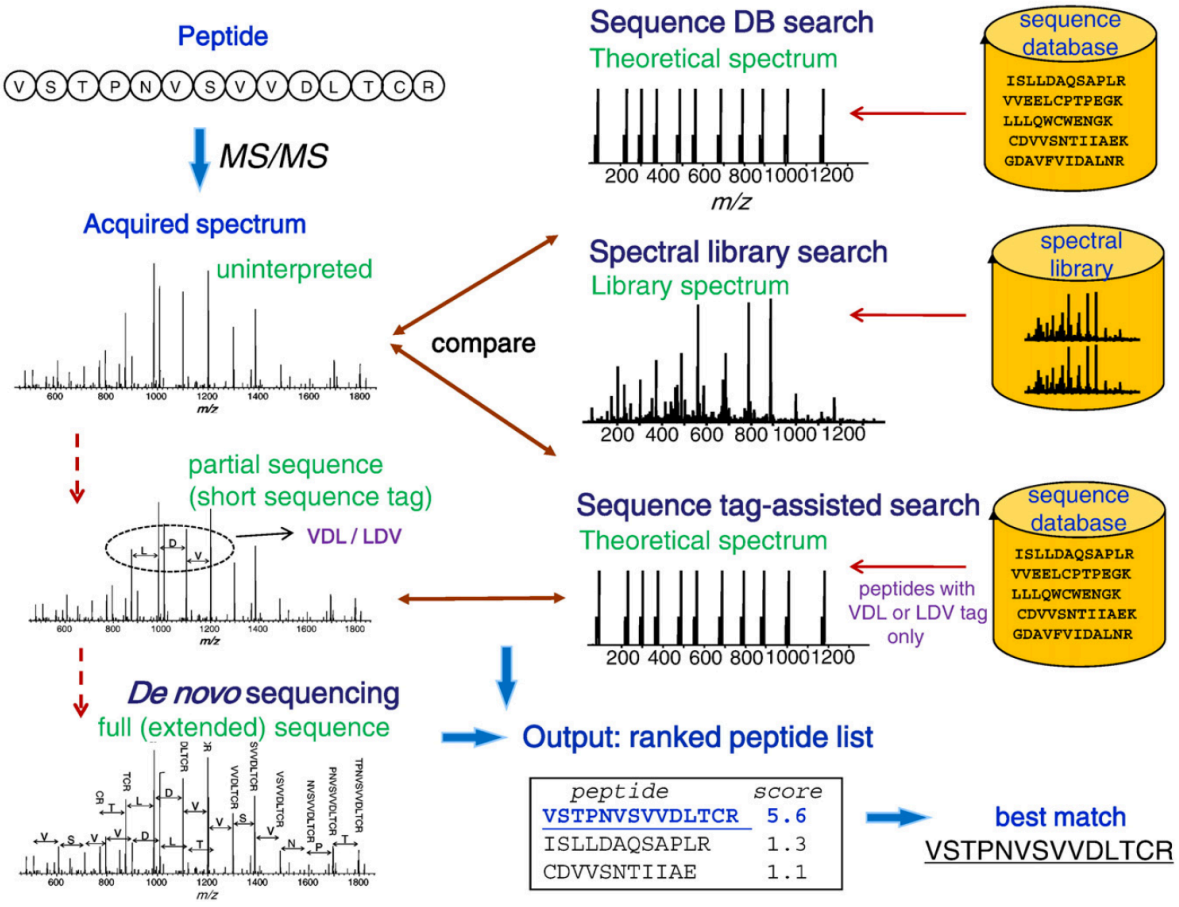

Figure 2.32: Peptide identification strategies [95]. Peptide identification can be performed by database search approach, spectral library search, *de novo* sequencing, and hybrid approaches.

Sequence Database Searching

In high-throughput studies, searching MS/MS spectra against protein sequence databases is the most efficient peptide identification method. The most popular software packages for this task include Mascot [103, 14], PEAKS [82], Tandem [26], Sequest [37, 4], Ommsa [44], and Phenyx [25]. Slightly modified procedure is used for different packages in the dataset analyses. But all of them include two major steps: first, each MS/MS spectrum is used to identify a peptide sequence from the database; secondly, the peptides are grouped together to identify the proteins from the database.

In the first step of peptide identification, a scoring function is used to measure the quality of the matching between a given peptide and the MS/MS spectrum. All peptides in the database with proper mass values are scored using the spectrum and the scoring function, and the highest scoring peptide is output as the answer. A good scoring function is of primary importance for the accuracy of peptide identification. Given a spectrum and a peptide, most software computes the theoretical *m/z* values of the fragment ions of the peptides, and matches the peaks of the spectrum with the *m/z* values. In the scoring function, the intensities and the numbers of the matched peaks, as well as the mass errors of the matches are usually taken into account. Because a certain type of mass spectrometer usually produces higher intensity peaks for certain ion types, then the fragment ion types are also important [80].

One cannot directly compare the peptide identification scores of different search engines. A method was suggested to "normalize" different scores by using the significance of the matching [38]. During the database searching, a "survival function" is trained by the suboptimal matches to the input, which is then used to convert the matching score to a significance value.

Even with a good scoring function, false discoveries still exist. One wants to know the false discovery rate (FDR) at certain score threshold, then he can determine which analysis results are trustworthy and the others discarded. Currently, the so-called decoy database method is commonly used in this result validation step [35]. In such a method, one can generate a random database (the decoy) with similar statistical properties as the target database. Moreover,

the peptide identification algorithm is done both on the target and decoy database. Consequently, one can estimate the FDR at a given score threshold by the number of matches in the decoy database with scores above this threshold. Sometimes decoy database will be used incorrectly, Zhang and Xin et al. pointed out that there were three general pitfalls in FDR and gave corresponding solutions in their PEAKS DB [134].

After all peptides are identified, it is still a challenging problem for the protein identification. One reason is that not all peptides of a protein can be identified. A protein is usually true, if two or more peptides of it are identified with high confidence. However, for a protein with only one identified peptide, it is difficult to judge whether it is a false discovery. In protein identification, this is commonly known as the "one hit wonders". Consequently, the combination of the MS and MS/MS spectra was proposed to improve this situation [78]. Another reason is that each identified peptide may be shared by a few proteins in the database. This is often caused by the homologous proteins existing in the database. Therefore, it is difficult to determine which of the proteins sharing the same peptides are real ones. The relationship between the identified peptides and the proteins in the database can be imagined as a bipartite graph. Figure 2.33 illustrates a simplified example of a protein summary list, where shared peptides are marked with an asterisk. In this figure, the proteins that cannot be conclusively identified are shown at the end of the list [96]. It is a difficult problem to infer the correct proteins from this bipartite graph and researchers were aiming to deal with this situation [96].

In proteomics, protein identification is the most mature application of MS. However, the current software is still not perfect for the aforementioned reasons. Additionally, for some instruments only $5 \sim 50\%$ of the MS/MS spectra could be confidently mapped to the peptides in the database [59]. There are a few reasons for this low utilization of the data. The largest reason is perhaps due to noise spectra, which may be caused by poor fragmentations of the peptide ions. The containing of these noise spectra not only increases the computational complexities, but also increases the FDRs of the results. The PTMs in the peptides is another reason for the low utilization of the dataset. In practice, a significant portion of peptides in the digested

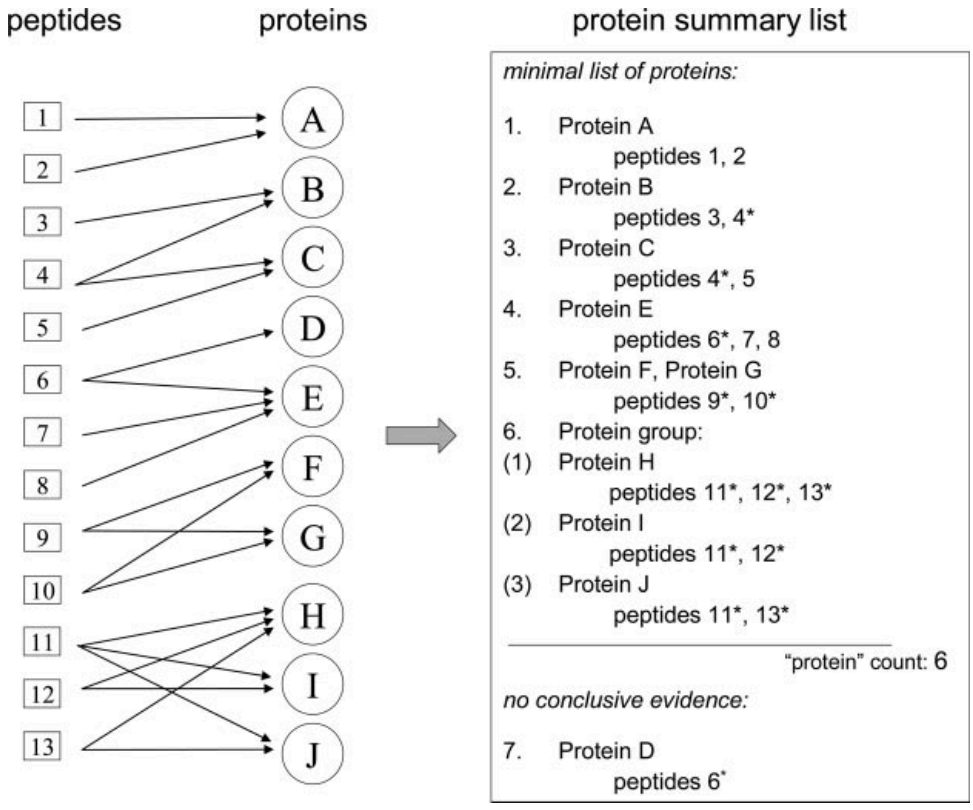

Figure 2.33: A simplified example of a protein summary list [96]. Peptides are apportioned among all their corresponding proteins, and the minimal list of proteins is derived that can explain all observed peptides. Proteins that are impossible to differentiate on the basis of identified peptides are collapsed into a single entry (F and G) or presented as a group (H, I, and J).

samples are modified, which could change mass for some residues. Usually, both fixed and variable PTMs are allowed in a protein identification software package. Consequently, PTMs increase the complexity of the protein identification significantly [80].

Peptide sequence database searching can also be used to identify mixture spectra (the concurrent fragmentation of multiple precursors), which are observed quite frequently in mass spectrometry experiment [76]. LC-MS/MS is also the primary approach for disulfide bond analysis. A new computational method termed DISC was proposed, by matching an input MS/MS spectrum against the putative disulfide linkage structures, which are hypothetically generated from a protein database [77].

Spectral Library Searching

Instead of searching acquired MS/MS spectra against theoretically predicted spectra, one can assign MS/MS spectra to peptides by matching against a spectral library [132, 27, 42, 63]. The spectral library is compiled from a large collection of experimentally observed MS/MS spectra identified in previous experiments. By using a certain mass tolerance window to restrict the set of candidate spectra, a newly acquired MS/MS spectrum can be compared to library spectra to determine the best match [119]. Existing spectral library search tools include Bibliospec [42], X! Hunter [27], and SpectraST [63]. The National Institute of Standards and Technology (NIST) spectral libraries are available for multiple organisms, and contain dataset from a variety of MS instrument types [95].

The spectral library matching approach performs better than traditional sequence database searching in terms of speed, error rates, and sensitivity of peptide identification [63]. Another advantage is that statistical models developed for assessing the validity of the peptide identifications by database searching are adaptable to this method [63]. As a drawback, only those peptides can be identified whose spectra were previously identified and entered into the library. Consequently, the existing libraries are incomplete, especially regarding peptides from low abundance proteins and peptides containing PTMs. At present, the spectral library matching

tools remain underutilized and mostly used as an additional step in multi-stage strategies [95].

De Novo Sequencing

The spectral library and sequence database searching approaches require the target spectra, peptides, and proteins to be in the database. However, this prerequisite is often not satisfied due to many reasons, including inferior gene prediction from the genome, incomplete genome sequencing, and alternatively spliced genes, etc. Under those situations, *de novo* sequencing is the only choice to identify the peptides. A *de novo* sequencing algorithm inputs a MS/MS spectrum, and outputs a peptide sequence that best explains the spectrum. Additionally, the peptide sequence is only constructed by the algorithm from the MS/MS spectrum, without any protein database [80].

When database searching, the algorithm for a scoring function is simple, which can enumerate each peptide in the database with proper mass values. While this is not the case any more for *de novo* sequencing. It is exponential time for enumerating every possible amino acid combinations with a given total mass value. Therefore, designing an efficient algorithm to construct the optimal peptide sequence is very important for *de novo* sequencing. Consequently, some software packages were developed for *de novo* sequencing include PEAKS [82], PepNovo [40] and Lutefisk [122].

One can interpret the spectrum manually by examining the ion ladders. A series of high intensity peaks are named ladders, if the *m/z* difference between every adjacent pair of peaks approximately equals the mass of an amino acid residue. In a CID MS/MS spectrum, one can observe all the *y*-ions (or *b*-ions), if the peptide fragmentation is ideal. Additionally, their peaks should form a complete series of *y*-ions (or *b*-ions) ladders. Consequently, the amino acid sequence of the peptide can be derived from the mass differences between adjacent peaks in the ladders.

The imperfect dataset is the largest reason for the difficulties of *de novo* sequencing. Firstly, only partial sequence information can be derived if the ion ladders are incomplete. Both the

N-terminal ion (e.g., *y*-ion) and C-terminal ion (e.g., *b*-ion) ladders are examined by most algorithms to improve the sequencing accuracy and coverage. Additionally, some of the internal fragment ions are utilized by the PEAKS algorithm to further improve the accuracy [82]. Secondly, there are much more peaks than just the N-terminal and C-terminal ion ladders. In fact, many of these peaks are from other fragmentations of the peptide. Consequently, the algorithm can misinterpret some of the other peaks as the peaks in the N-terminal or C-terminal ion ladders, which would cause errors in the result [80].

Peptide *de novo* sequencing is significantly harder than peptide identification with a database. In order to derive the complete peptide sequence, it requires much higher quality dataset. It is desirable to derive a partial sequence tag, when the complete peptide sequencing is not possible. Many *de novo* sequencing packages such as Lutefisk [122] outputs partial sequence tags when they are unsure about some amino acids. Additionally, PEAKS software computes a "local confidence score" for each amino acid in its *de novo* sequencing result [82]. Consequently, a sequence tag is formed from the remaining amino acids, by removing the amino acids below a confidence threshold. Recently, many new *de novo* sequencing methods have been developed, with the improvements in the accuracy of MS/MS and development of alternative fragmentation modes of MS/MS (compared to traditional CID spectra) [129].

Glycosylation is one of the most important and prevalent PTMs in proteomics, and a heuristic algorithm for glycan *de novo* sequencing from HCD MS/MS spectra of N-linked glycopeptides was proposed to identify the glycopeptides [120]. In a typical wet-lab mass spectrometry experiment, mixture spectra occur quite frequently, and *de novo* sequencing can also be used in the peptide identification for mixture spectra [75].

Hybrid Approaches

The combination of the elements from *de novo* sequencing and database searching is another approach. One common way is to start by extracting, for every acquired MS/MS spectrum, a set of short "sequence tags" [85] which are likely to be a part of the true peptide

sequence. A tag is a short amino acid sequence with a prefix mass and a suffix mass values that designate its position within the peptide sequence. Additionally, the database search for each MS/MS spectrum is only performed against those candidate database peptides which contain one of the sequence tags extracted from that spectrum, thus reducing the number of comparisons and the search time [95]. Hybrid approaches are particularly useful for the identification of PTM peptides [30]. Additionally, the combination of different approaches can improve the performance of each other. For example, with the use of spectra pairs from the same peptide under different fragmentation modes (HCD and ETD), the performance of *de novo* sequencing can be significantly improved [130].

Chapter 3

State-of-the-Art Peptide Mass Fingerprinting

Peptide mass fingerprinting (PMF) is an analytical technique for protein identification, where the unknown protein of interest is first cleaved into smaller peptides, whose absolute masses can be accurately measured with a mass spectrometer.

In this chapter, we will introduce the development of PMF. Additionally, we formulate PMF problem mathematically, and these definitions and notations will be used in this thesis. Consequently, we will review some state-of-the-art PMF approaches, especially for MOWSE, optimization-based PMF, and PMF by the order of retention time.

3.1 Peptide Mass Fingerprinting (PMF)

Peptide mass fingerprinting (PMF) was the first available method to identify proteins using MS, and is still widely used [51]. This approach uses theoretical spectra each comprising the list of masses expected by an enzymatic digestion of each protein sequence in the reference database [86]. The difference between PMF and peptide fragment fingerprinting (PFF) is illustrated in Figure 3.1. PMF only uses MS spectra to identify proteins. In PFF, the set of MS/MS spectra, along with information such as the parent (precursor) mass of these fragmented pep-

tides, are all used in the database search [137]. Traditional PMF was first introduced in 1993, and the representative was MOWSE [100]. Until now, the most popular PMF methods are Mascot (1999) [103], MS-Fit (1999) [24], ProFound (2000) [137], and Aldente (2003) [125].

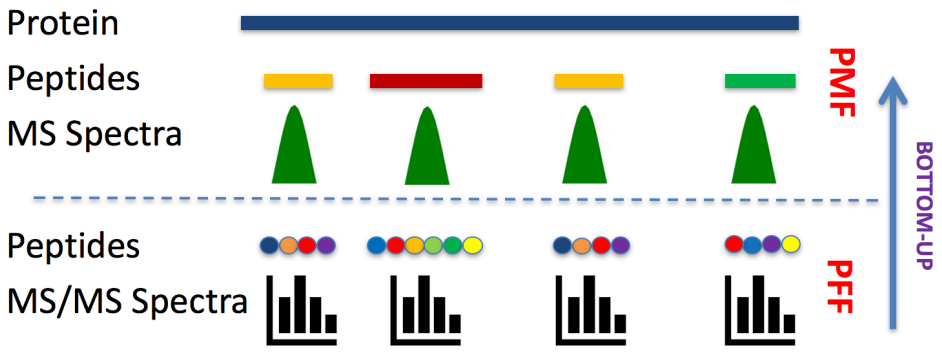

Figure 3.1: The difference between PMF and PFF.

The scoring system is the heart of all protein identification methods. Mass spectrometry dataset derived from an unidentified protein are compared with theoretical dataset from known proteins, and a score is assigned based on how well these two datasets compare. For a unknown protein, a "hit" is termed as any score above an arbitrary confidence threshold. The unknown protein is expected to be the top such hit. If there are no scores above this threshold ("no hits"), then the protein remains unidentified. The score distributions for protein identifications using a scoring scheme is shown in Figure 3.2, where there is no score threshold including all correct identifications while simultaneously excluding all false positives. From this figure, we know that the separation of true from false protein identifications based on a score is never perfect [86].

PMF is popular and works well in practice, since it is relatively fast to compute PMF scores against a database. For good quality samples belonging to well-characterized model organisms, PMF can identify proteins with high confidence in many cases, especially in organisms with smaller genomes. Unfortunately, sometimes a sample spectrum does not resemble any theoretical spectra in the protein database closely enough to make a confident identification. This can happen for many reasons, such as unexpected PTM, splice variants, single nucleotide

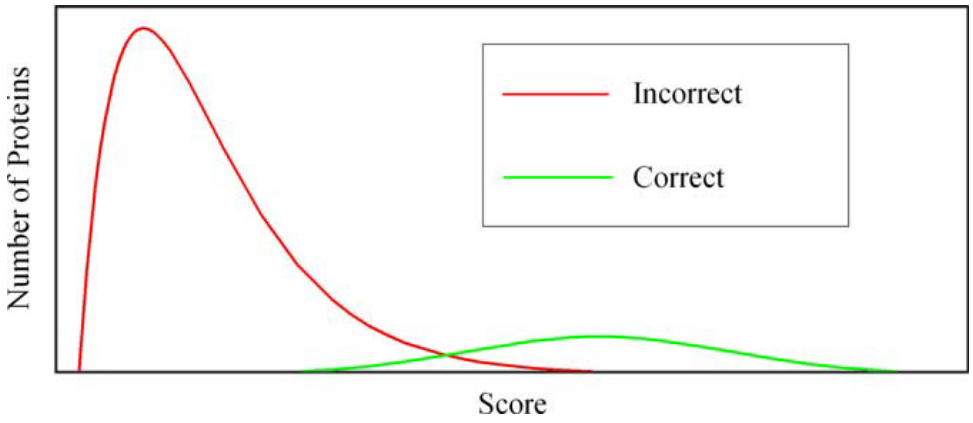

Figure 3.2: The correct and incorrect score distributions [86]. Score distributions for correct (green) and incorrect (red) protein identifications using a scoring scheme.

polymorphisms (SNPs), etc. More proteins can now be identified with confidence, as more sophisticated methods for scoring PMF have been developed. This corresponds to a better separation of true from false positives using the scoring system. The next step involves deciding on a threshold: the identified proteins are those scores above this threshold. It is difficult to define a threshold because setting the threshold too high may result in the correct protein not being identified (false negative), while setting a threshold too low will incorrectly identify a number of proteins (false positives) [86].

One of the limitations of PMF is its sensitivity to database size. As the search database grows, there is a direct effect on the statistical confidence a PMF algorithm can ascribe to protein identification. A larger database has an elevated chance of the experimental masses randomly matching theoretical peptide masses in these databases, thereby decreasing the confidence of protein identifications using PMF [86]. Therefore, a larger database should not allow for better protein identification. Many experimentalists use PMF as a "first pass" to identify a protein, and if the identification fails, move on to methods such as PFF [137]. Some of the more popular PMF packages are listed below in Table 3.1.

Aldente [125] is hosted on the ExPASy Proteomics Server as one of a suite of bioinformatics tools. Released in 2004, Aldente uses a robust Hough transform (a feature extraction technique used in image analysis) to speed searches and find straight lines hidden in the data

Table 3.1: A short list of popular PMF packages

PMF Package	*URL*
Aldente	http://www.expasy.org/tools/aldente
Mascot	http://www.matrixscience.com/search_form_select.html
MS-Fit	http://prospector.ucsf.edu/prospector/cgi-bin/msform.cgi?form=msfitstandard
ProFound	http://prowl.rockefeller.edu/prowl-cgi/profound.exe

(the classical Hough transform was concerned with the identification of lines in the image), making this tool more robust to noise than other PMF packages.

Mascot [103] uses a proprietary scoring algorithm but is known to be based on the MOWSE algorithm [100], which is first described in 1993. By calculating the distribution of tryptic peptide lengths across the entire search database, a probability can be calculated for every observed peak, assuming this match is purely random. Here, the probability represents that whether the observed match is a random event. In Mascot, the searches in smaller protein databases, such as bacterial databases, generally has lower threshold scores for confidence than those conducted in larger databases for higher organisms. We can also infer that for noisy experimental spectra, these extra peaks contribute to the possibility of a random match, and thus raise the confidence score threshold for a given probability [86].

MS-Fit [24] is also a probabilistic algorithm, again based on MOWSE, but runs over FASTA format [148] databases. MS-Fit first bins proteins according to the mass weight. Within each of them, a series of bins are created according to the tryptic peptide masses. When calculating the probability of a random tryptic peptide match, it is calculated specifically for the distribution of these peptide masses for a given precursor mass. Therefore, it can reduce the size of the search database effectively [86].

ProFound [137] uses a Bayesian probability scoring system for the hits, with additional information. For example, provisions for the knowledge that particular amino acids are present (or absent) in the sample protein. Each piece of information functions as an additional constraint upon the search space of protein database, thus reducing the effective size of the database

against which the search is conducted [86].

A study by Chamrad et al. [20] in 2004 using MALDI-TOF mass spectrometry dataset from a project mapping genes onto mouse chromosomes used expert interpretation of the spectra to identify 70% of the proteins, thus forming a reference set for PMF algorithm comparison. This study found the performance of Mascot and ProFound to be similar, correctly identifying around 53% of proteins from the reference set at a 5% significance level (the probability of rejecting the null hypothesis when it is true in hypothesis tests), with MS-Fit identifying only 32% using the same input parameters.

Another study in yeast using 266 spectra gathered on MALDI-TOF instruments from three different manufacturers in 2004 found Mascot to outperform ProFound, with Mascot identifying 45% of proteins while ProFound identified 33% of the proteins [109].

Among traditional PMF, Mascot performed the best from above studies. More recently, PMF was tackled as an optimization problem by He et al. [50] assuming each peak can only be assigned to one protein (2010). In addition to the peptide mass, the RT was also used to facilitate protein identification (2011) by Bochet [17].

In the following sections, we will introduce MOWSE, optimization-based PMF, and PMF by the order of retention time.

3.2 Preliminaries for PMF

We shall use the following definitions and notations in this thesis.

Assuming we have an input set of experimental peaks $P = \{l_i(m_i, r_i) \mid i = 1, 2, \ldots, n\}$, where l_i is a peak, m_i is the mass of l_i, r_i is the corresponding RT, and n is the total number of peaks. We also assume there is a database of proteins $D = \{p_j \mid j = 1, 2, \ldots, s\}$, where p_j denotes a protein, and s is the size of this database. We use Σ to denote the alphabet of 20 different types of amino acids. For an amino acid $a \in \Sigma$, we use $\|a\|$ to symbolize the mass of its residue. Let $S = a_1 a_2 \ldots a_d$ be a string of amino acids, we define the residue mass

of this peptide as $||S|| = \Sigma_{1\leq k\leq d}||a_k||$ and the actual mass as $||S|| + ||H_2O||$, where H_2O is the chemical formula of water. A matching between one theoretical peptide S and an experimental peak l_i means that $|||S|| + ||H_2O|| - m_i| \leq \delta$, where δ is the mass tolerance threshold. We use a scoring function $F(P, p_j, \delta)$ to measure the quality of matching between peaks set P and a protein p_j in the database D. When δ is fixed, we can simplify $F(P, p_j, \delta)$ as $F(P, p_j)$. A protein consists of a set of peptides, usually from proteolytic digestion. We denote $u_q \in p_j$ as a peptide u_q in a protein p_j. For $u_q \in p_j$, if it matches with $l_g \in P$, then we denote such match as $< u_q, l_g >$, and denote such u_q and l_g as $u_q \in_P p_j$ and $l_g \in_{p_j} P$. The set of matched peptides of p_j is $U(p_j) = \{u_q \in_P p_j | q = 1, 2, \ldots, t\}$, where t is the total number of matched peptides in p_j. For each matched peptide u_q, there will be a real number matching score, denoted as $f(u_q)$, whose exact definition will be decided later. The set of matched peaks for p_j in P is $L(p_j) = \{l_g \in_{p_j} P | g = 1, 2, \ldots, o\}$, where o is the total number of matched peaks in P for p_j. We assume that a peptide in a protein can match to more than one peak, since there could be peaks with exactly the same, or very similar masses but different retention time. At the same time, a peak can match to more than one peptide, since there are different peptides with exactly the same or very similar masses. When considering the matching score between a protein and a set of peaks, a peak can only match to one peptide and a peptide can only match to one peak, since multiple use of peaks will generate bias in the scoring scheme (for example, a peak matches to two peptides with different scores). We now define a matching, $M(p_j, P)$, between protein p_j and P as a set of pairs (u_q, l_g), where $u_q \in U(p_j)$ and $l_g \in L(p_j)$, satisfying the following conditions: (1) $< u_q, l_g >$: (u_q matches to l_g); (2) for any two pairs (u_q, l_g) and (u_r, l_h) in M, $q = r$ if and only if $g = h$: (one to one condition). The matching score of $M(p_j, P)$ is the summation of each matching peptide in M: $S(M(p_j, P)) = \sum_{(u_q,l_g)\in M} f(u_q)$. The score of a protein p_j with respect to P is defined as maximum matching score: $F(P, p_j, \delta) = \max_M S(M(p_j, P))$.

For PMF, our objective is to find a set of proteins $Z \in D$ that contains all the proteins whose scores are greater than a threshold score Δ, i.e. $Z = \{p_j \mid p_j \in D, F(P, p_j, \delta) \geq \Delta\}$.

3.3 MOWSE

PFM is a challenging problem since we only have the masses of peptides to identify proteins. The technical details of Mascot [103] are not published, but is known to be based on the MOWSE [100] algorithm. In MOWSE, a frequency matrix X is created, where each row represents an interval of 100 Da in peptide mass, and each column represents an interval of 10 kDa in intact protein mass. Consequently, we scan each sequence entry in entire protein database D. For a protein p_j, after tryptic digestion, for each peptide $u_v \in p_j$, its corresponding matrix element $x_{de} \in X$ would be incremented, where d is the interval number of peptide u_v and e is the interval number for protein p_j. Here, we count the total number of peptides for each cell x_{de}. The tryptic peptide frequency matrix X in Human database is shown in Figure 3.3. Consequently, the observed distribution frequency of all tryptic peptides in Human database is shown in Figure 3.4, where protein intervals range from 1 to 400, and peptide intervals range from 1 to 2,376. When zooming in Figure 3.4 for the partial tryptic peptides, we get Figure 3.5.

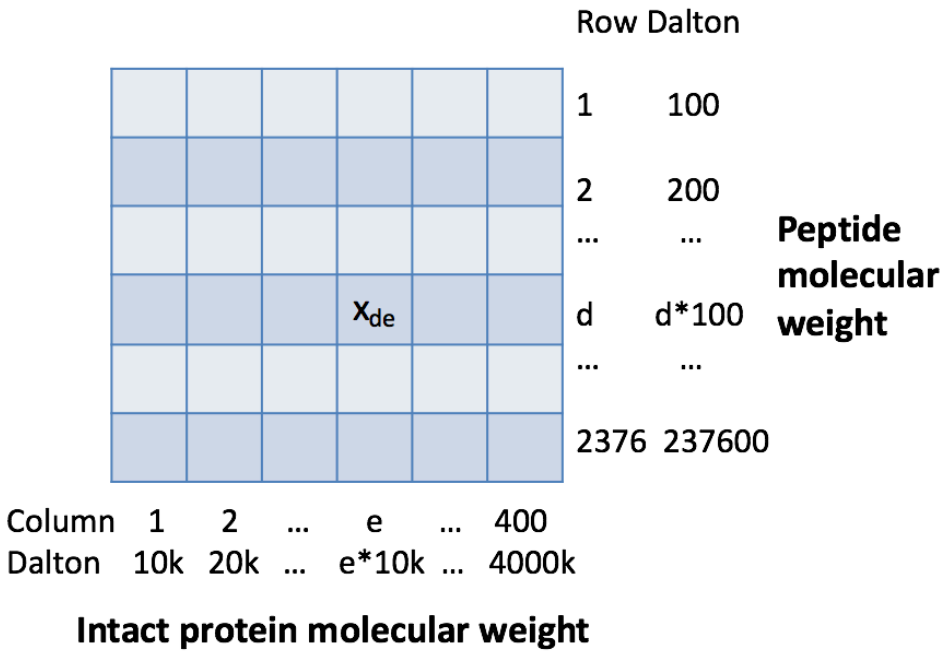

Figure 3.3: Tryptic peptide frequency matrix X in Human database.

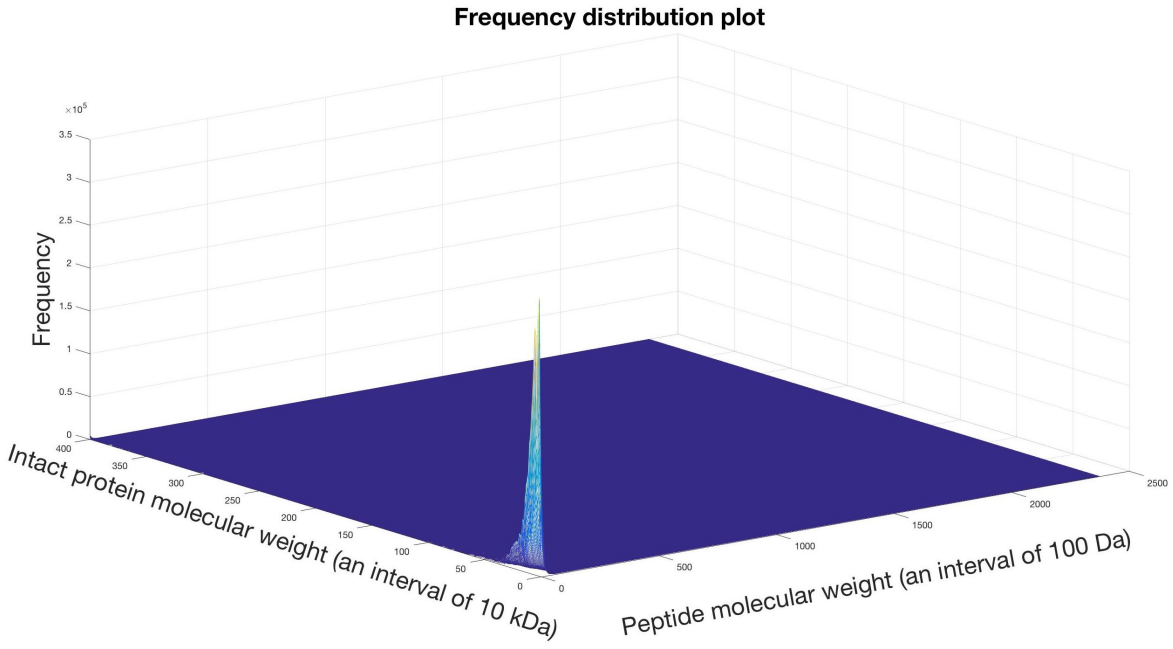

Figure 3.4: A frequency distribution plot for the tryptic peptides in Human database. Protein columns range from 1 to 400, and peptide rows range from 1 to 2,376.

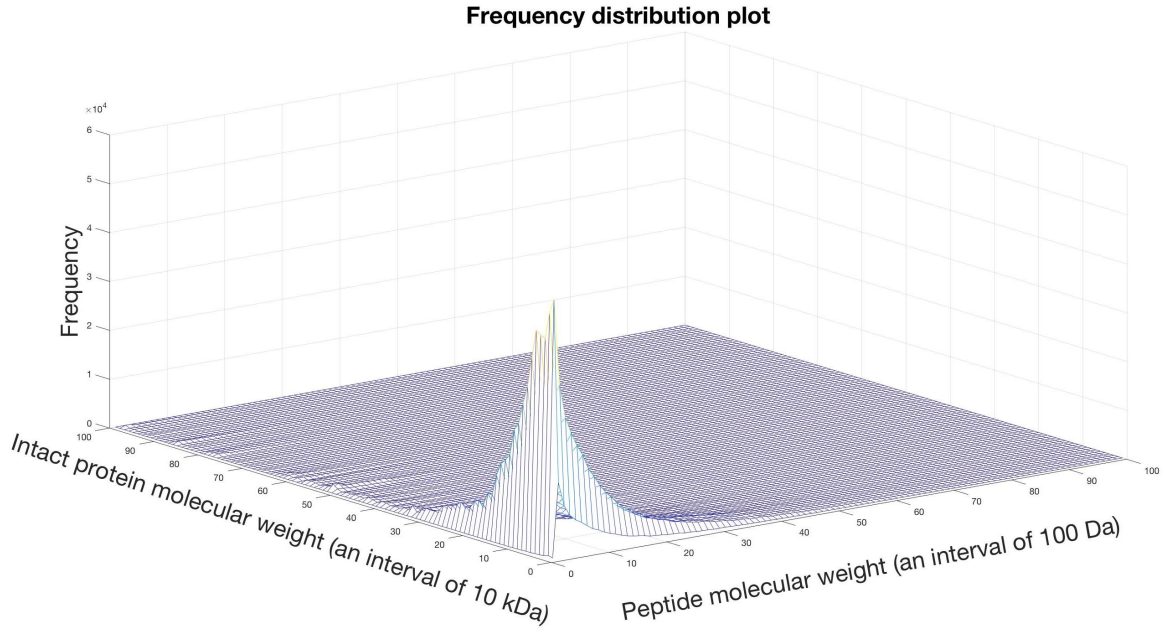

Figure 3.5: A zoomed frequency distribution plot for the tryptic peptides in Human database. Protein columns range from 1 to 100, and peptide rows range from 1 to 100.

The elements of X are normalized by dividing the cells of each 10 kDa column with the largest value in that column. An example for column 1 is shown in Figure 3.6, where the largest value is in row 2, which is 15,666. In Human database, if we combine several consecutive columns together, we get Figure 3.7, where the normalized distributions among them are very similar. Consequently, if we combine several different columns together, we will get Figure 3.8, where the normalized distributions among them are still similar, while it is inconsistent for the large column, since there is not enough protein database for the large columns. We also check the distribution for proteins and peptides from one dimension in Human database. Figure 3.9 illustrates the normalized frequency distribution plot for all protein columns, and Figure 3.10 shows the normalized frequency distribution plot for all peptide rows.

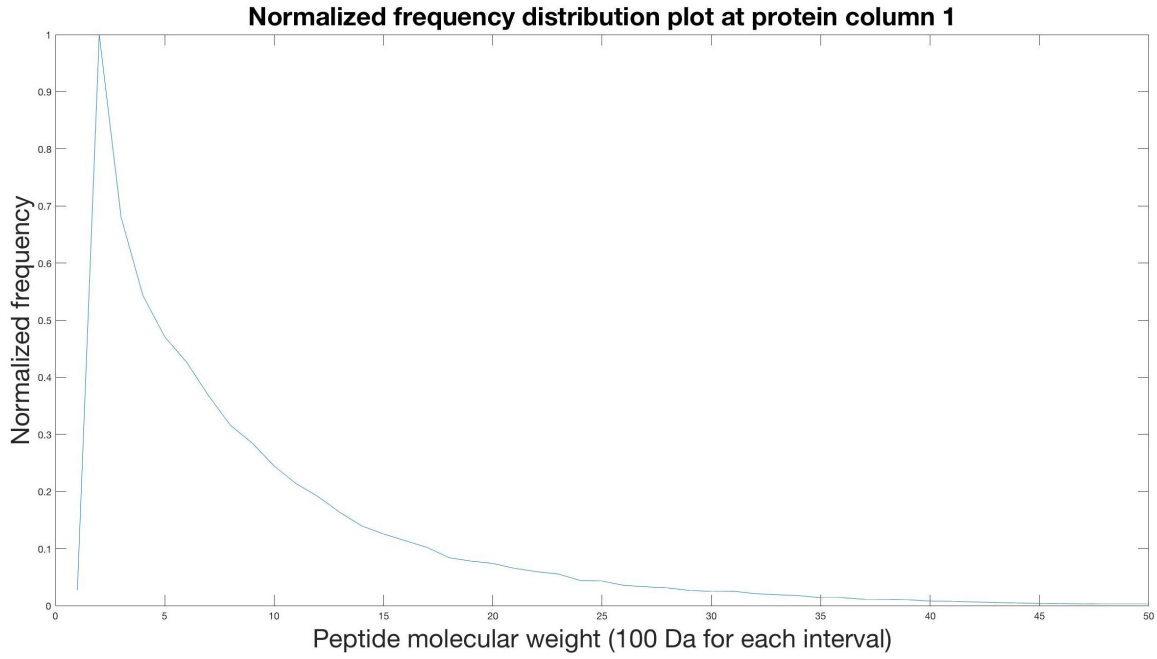

Figure 3.6: A normalized frequency distribution plot in Human database at protein column 1.

With this mass distribution of tryptic peptides, the final score for each protein is calculated as: $F_M(P, p_j, \delta) = 50,000/(|p_j| * \prod_{1 \leq q \leq t} f'_m(u_q))$, where 50,000 is a constant, $|p_j|$ is the molecular weight of the protein, $u_q \in_P p_j$ and u_q is a matching peptide of p_j, $f'_m(u_q)$ is the normalized score for x_{de} (in the frequency matrix X), and t is the total number of matches for p_j. From

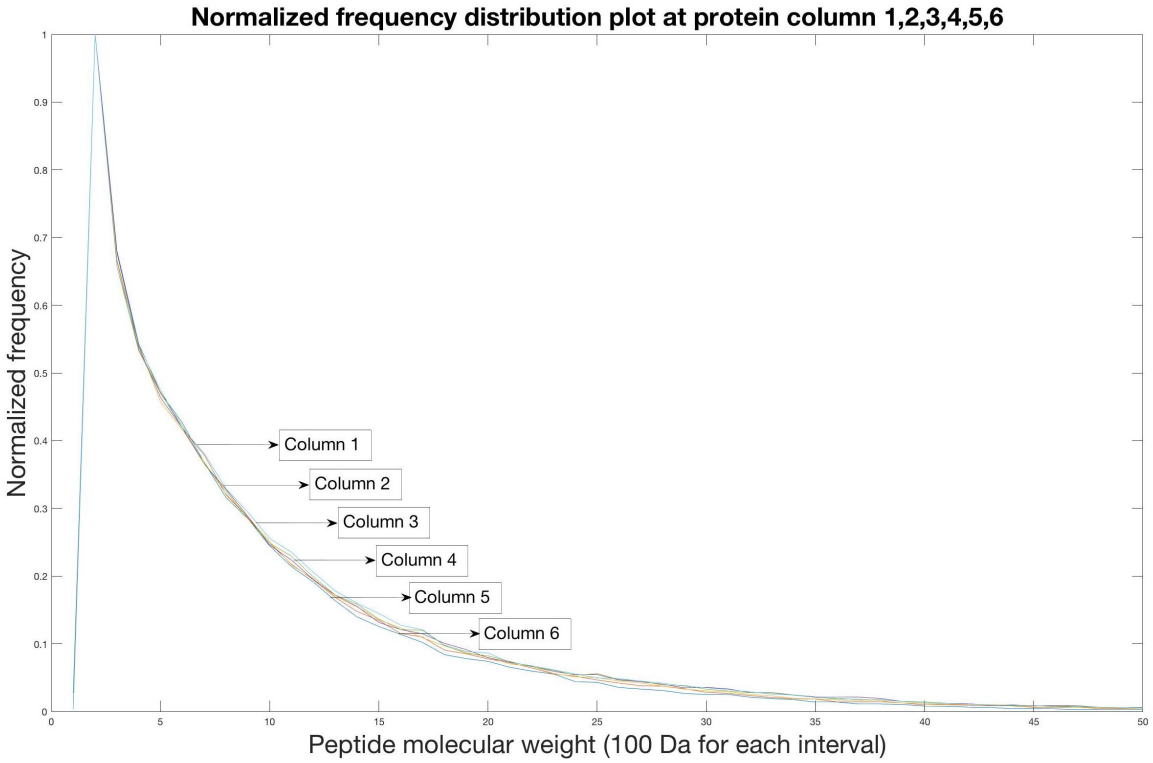

Figure 3.7: Normalized frequency distribution plots for protein columns 1,2,3,4,5,6.

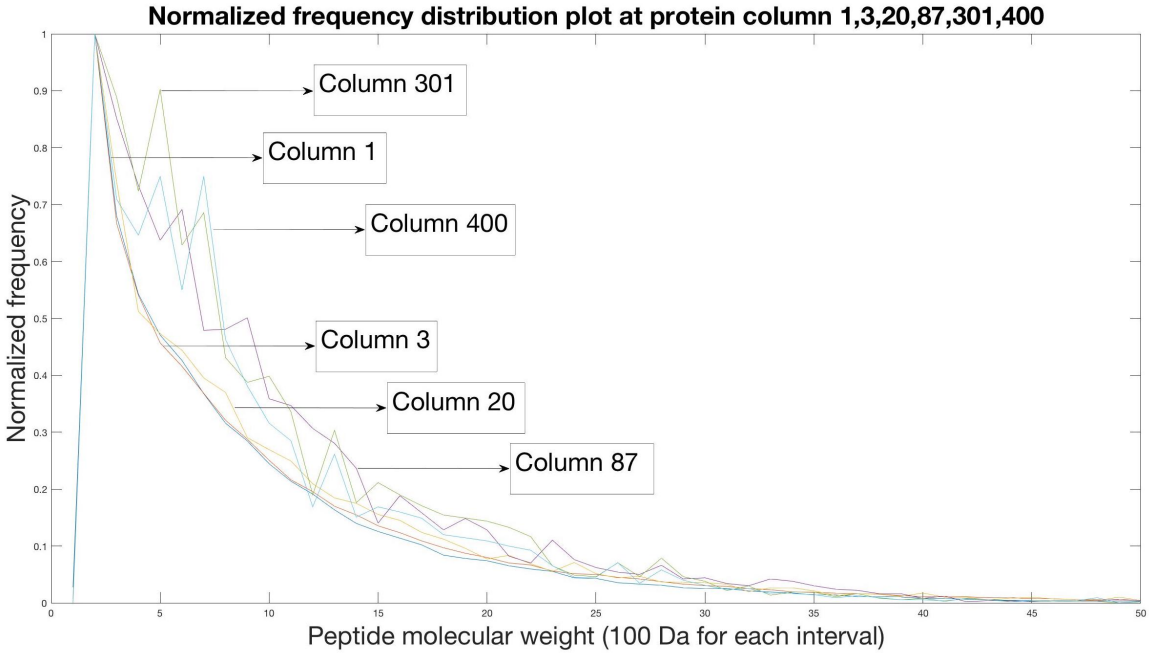

Figure 3.8: Normalized frequency distribution plots for protein columns 11,3,20,87,301,400.

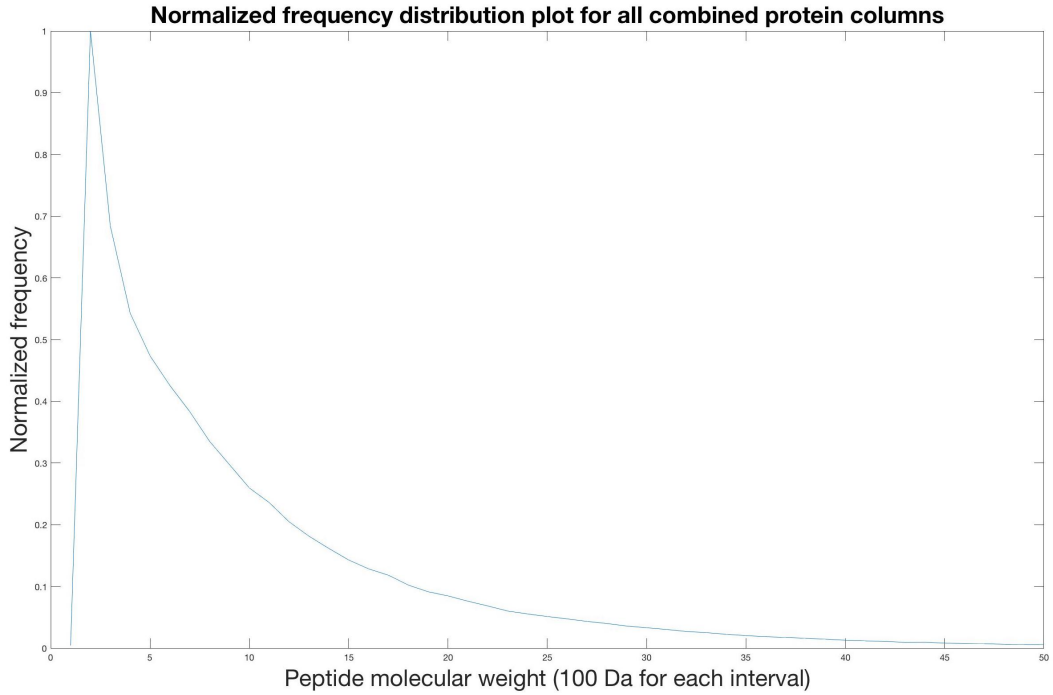

Figure 3.9: The normalized frequency distribution plot for all combined protein columns.

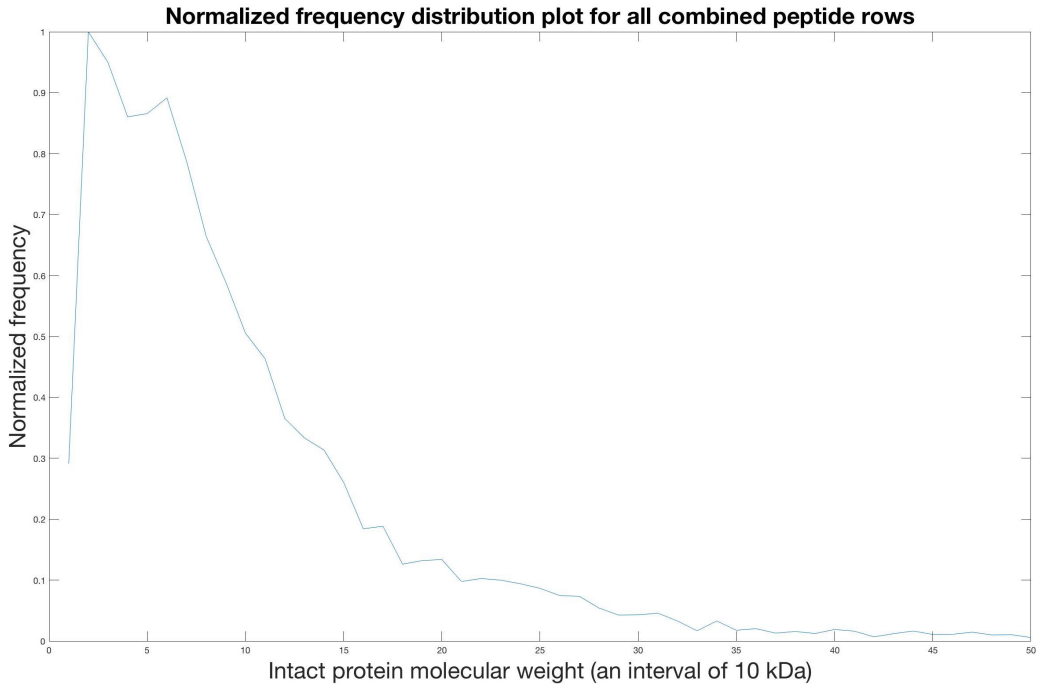

Figure 3.10: The normalized frequency distribution plot for all combined peptide rows.

this score function, the final score for each protein depends on the scores of matching peptides, they also use a constant (50,000) divided by the protein mass to adjust the raw score. For a matching peptide in a protein, if the normalized score for this peptide is high, it contributes less for the protein score, otherwise it gives more contribution to the final score. Another factor is the mass of protein, for large protein it will get smaller score with the same matches.

The MOWSE framework is shown in Algorithm 1.

Algorithm 1: MOWSE: finding a set of proteins to best explain experimental peaks.

Input: D: a database of s proteins;
P: an observed peak list;
δ: a mass tolerance threshold;
$U(p_j)$: a set of matched peptides for protein p_j;
$f'_m(u_q)$: a normalized score for a peptide $u_q \in_P p_j$;
$|p_j|$: the molecular weight of the protein p_j;
Δ: a threshold score ;
Output: Z: a set of proteins $Z \in D$ that best explain P;

```
for j := 1 to |D| do
    F_M(P, p_j, δ) ← 0
    for q := 1 to |U(p_j)| do
        F_M(P, p_j, δ) ← F_M(P, p_j) + lg(1/f'_m(u_q))
    end for
    if |U(p_j)| > 0 then
        F_M(P, p_j, δ) ← F_M(P, p_j, δ) + lg(1/|p_j|) + lg(50,000)
    end if
end for
Z ← {p_j | p_j ∈ D, F_M(P, p_j, δ) ≥ Δ}
return Z
```

3.4 Optimization-Based PMF

In 2010, He et al. [50] tackled PMF as an optimization problem for peaks partition, assuming each peak can only be assigned to one protein. They used their methods for protein mixture identification, and they could identify part of 49 proteins from real data, which obtained much better identification results than traditional approaches. But they only considered the assignment of peaks to proteins, and ignored the matched peptides between the peaks and proteins,

which would also lose some important information at the same time. The relationship between peaks, peptides, and proteins for Optimization-Based PMF are shown in Figure 3.11, where the authors only assigned the peaks to proteins.

Figure 3.11: Optimization-based PMF.

The main objective of their paper was to study the potentials and pitfalls of PMF in protein mixture identification. They first demonstrated that protein mixture identification was an optimization problem. While obtaining an optimal solution was difficult, they could employ some heuristic searching methods to find the local optima. As a result, the identification accuracy was significantly better than that of traditional PMF and subtraction strategy (will be introduced later). They also demonstrated that the performance of PMF was mainly affected by the mass accuracy of mass spectrometer, the number of component proteins in the mixture, the sequence coverage of each protein and the noise level in a MS dataset [50].

If there is only one protein in the sample (one ground-truth protein), assuming that this ground-truth protein has the highest score, the problem of single protein identification is an optimization problem:

$$Z = \arg\max_{p_j \in D} F^{(L)}(P, p_j),$$

where Z is the protein with the highest score and $F^{(L)}(P, p_j)$ is a scoring function. In other words, Z best "explains" P.

In the context of protein mixture identification (multiple ground-truth proteins), the objective is to find a set of proteins Z that best "explains" P:

$$Z = \arg\max_{Y \subseteq D} F^{(M)}(P, Y),$$

where Y denotes a subset of proteins and $F^{(M)}(P, Y)$ is a scoring function. Note that $F^{(M)}(\cdot,\cdot)$ is different from $F^{(L)}(\cdot,\cdot)$ as Y may have multiple proteins.

Obviously, single protein identification is a special case of protein mixture identification when an additional constraint $|Y| = 1$ is provided.

To define $F^{(M)}(P, Y)$, there are two choices:

1. Virtual single protein approach: considering Y as a "virtual" single protein V, yielding the following relationship:

$$F^{(M)}(P, Y) = F^{(L)}(P, V).$$

2. Peak partition approach: distributing peaks in P to different proteins in Y explicitly, assuming that each peak can only be assigned to one protein (In general, such assumption is invalid since one peak may belong to multiple proteins). One needs to partition the peaks of P into disjoint subsets. Assuming that Y consists of k proteins $p_{g_1}, p_{g_2}, \ldots, p_{g_k} (1 \leq k \leq s)$ and divide P into k disjoint subsets $P_1, P_2, \ldots, P_k$, then the score is calculated as:

$$F^{(M)}(P, Y) = \sum_{j=1}^{k} F^{(L)}(P_j, p_{g_j}).$$

The *subtraction* strategy by Jensen et al. [58] in 1997 can be regarded as a special instance of peak partition approach in which peaks are divided in a greedy manner. Suppose the peak subsets $P_0, P_1, P_2, \ldots, P_k$ are generated by the subtraction strategy sequentially (P_0 is an empty set), the score at each step $j (1 \leq j \leq k)$ is calculated as:

$$F^{(L)}(P - \bigcup_{t=0}^{j-1} P_t, p_{g_j}).$$

The peak partition approach is very complicated since one needs to explicitly assign peaks to proteins, and it is a NP hard problem. Therefore, the authors use the virtual single protein approach to define the scoring function.

After defining the scoring function, one needs to find a subset of proteins that maximizes it. Even when k is given, an exhaustive search is prohibitive since there are totally C_s^k possible solutions, where s is the size of protein sequence database. This combination is a selection of k proteins from a database with the size of s. When s is large, computationally expensive approaches become unattractive.

The authors first assume that the true number of ground-truth proteins is known in advance, i.e., k is an input parameter. Then, they relax this requirement by introducing an adaptive algorithm that can determine the number of target proteins automatically.

Local search algorithm with known k (Losak) is a local search algorithm with known k for protein mixture identification. The details is shown in Algorithm 2. Losak takes the number of target proteins as input and iteratively improves the value of objective function. Initially, randomly selected k proteins are labeled as "target". In the iteration process, for each "non-target" protein, its label is exchanged with each of the k target proteins and the objective value is re-evaluated. If the objective value increases, the "non-target" protein is exchanged with the "target" one. When all "non-target" proteins have been checked, a full iteration is completed. The algorithm terminates when a full iteration does not change any labels, thereby indicating that a local optimum is reached.

Local search algorithm with unknown k (Losau) is an extension of Losak with details described in Algorithm 3. To determine the number of target proteins automatically, an "insertion and a "deletion" are introduced into the local optimization process. If protein p_i is contained in Z, it will be deleted from Z if such an operation increases the score. Similarly, if protein p_i is not contained in Z, it will be either inserted into Z or exchanged with another protein in Z if such an operation increases the score. To achieve a trade-off, a penalty ω on the insertion is introduced to reflect the intention of "explaining" Z using as few proteins as possible. In the first iteration, the number of proteins in Z is used to update ω value so that Z can be expanded to a reasonable size. In the subsequent iterations, the parameter $df(0 < df < 1)$ is used as the decay factor to decrease ω at each iteration: $\omega \leftarrow df \cdot \omega$.

Algorithm 2: *Losak*: local search algorithm with known k

Input: D: a database of s proteins;
P: observed peak list;
δ: mass tolerance threshold;
k: the number of target proteins;
Output: Z: a set of k proteins, $Z \in D$ best explain P;

```
Randomly select k proteins into Z as "target" proteins;
Initialize hasSwap ← True;
while hasSwap = True do
    hasSwap ← False;
    for i := 1 to s do
        if p_i does not belong to Z then
            h ← arg max_j F^(M)(P, Z + {p_i} − {p_{g_j}}); /* p_{g_j} belongs to Z; */
            if F^(M)(P, Z + {p_i} − {p_{g_h}}) > F^(M)(P, Z) then
                Z ← Z + {p_i} − {p_{g_h}};
                hasSwap ← True;
            end if
        end if
    end for
end while
return Z
```

Many false positives may be introduced into the final protein list because of the adaptive nature of Losau. It is desirable to filter out these incorrect proteins. Meanwhile, it is also necessary to provide a procedure to evaluate the confidence of each single protein. Algorithm 4 describes the filtering procedure. The peak subset $M_P(p_{g_j})$ is used to evaluate each protein p_{g_j}. Since all the peaks in $M_P(p_{g_j})$ match p_{g_j}, the probability that other proteins achieve better single protein identification score than p_{g_j} on $M_P(p_{g_j})$ is very low if p_{g_j} is the ground-truth protein. The number of "winning proteins" is used to measure the rank uncertainty and θ as the threshold. If more than θ proteins outperform p_{g_j} on $M_P(p_{g_j})$ in terms of single protein identification score, it is removed from the result set.

Both the simulated and real dataset are used to demonstrate the improvement of their algorithms over others. The real MS dataset is obtained from a mixture of 49 standard human proteins in the ABRF sPRG2006 study. The dataset is generated using a linear ion trap-orbitrap (LTQ-Orbitrap) mass spectrometer. In evaluation, they compare Losak and Losau with the fol-

Algorithm 3: *Losau*: local search algorithm with unknown k

Input: D: a database of s proteins;
P: observed peak list;
δ: mass tolerance threshold;
df: decay factor;
θ: rank threshold in filtering;
Output: Z: a set of k proteins, k is determined automatically.

Randomly select 2 proteins into Z as "target" proteins;
Initialize $\omega \leftarrow 1$ and $q \leftarrow 1$; /*ω: penalty value; q: iteration number */
Initialize $hasOperation \leftarrow True$;
while $hasOperation = True$ **do**
  $hasOperation \leftarrow False$;
  if $q = 1$ **then**
    $\omega \leftarrow df \cdot \omega$;
  end if
  for $i := 1$ **to** s **do**
    $\zeta_{noop} \leftarrow F^{(M)}(P, Z)$;
    if $q > 1$ **then**
      $\omega \leftarrow |Z|$;
    end if
    if $p_i \in Z$ **then**
      if $F^{(M)}(P, Z - \{p_i\}) > \zeta_{noop}$ **then**
        $Z \leftarrow Z - \{p_i\}$;
        $hasOperation \leftarrow True$; /* Deletion */
      end if
    end if
    else
      $h \leftarrow \arg\max_j F^{(M)}(P, Z + \{p_i\} - \{p_{g_j}\})$;
      $\zeta_{swap} \leftarrow F^{(M)}(P, Z + \{p_i\} - \{p_{g_h}\})$;
      $\zeta_{inst} \leftarrow F^{(M)}(P, Z + \{p_i\}) - \omega$;
      if $\zeta_{swap} > \zeta_{inst}$ *and* $\zeta_{swap} > \zeta_{noop}$ **then**
        $Z \leftarrow Z + \{p_i\} - \{p_{g_h}\}$;
        $hasOperation \leftarrow True$; /* Swap */
      end if
      if $\zeta_{inst} > \zeta_{swap}$ *and* $\zeta_{inst} > \zeta_{noop}$ **then**
        $Z \leftarrow Z + \{p_i\}$;
        $hasOperation \leftarrow True$; /* Insertion */
      end if
    end if
  end for
  $q \leftarrow q + 1$
end while
$Z \leftarrow ProteinFilter(D, P, \delta, \theta, Z)$; /* Filtering (See Algorithm 4) */
return Z

Algorithm 4: PoteinFilter Algorithm

Input: D, P, δ, θ, and Z is a set of unfiltered proteins.
Output: F: a refined set of proteins, $F \subseteq Z$.

```
Initialize F ← ∅;
for j := 1 to |Z| do
    Initialize Winner ← 0;
    for i := 1 to s do
        if F^(L)(M_P(p_{g_j}), p_i) > F^(L)(M_P(p_{g_j}), p_{g_j}) then
            Winner + +;
        end if
    end for
    if Winner < θ then
        F ← F + {p_{g_j}};
    end if
end for
return F
```

lowing algorithms:

- SPA: single protein identification algorithm with the scoring function in Samuelsson et al. [109] in 2004. The reasons are its scoring function has comparable performance to Mascot and ProFound, and Its score can be calculated incrementally. Since SPA ranks each protein in the database separately, k top-ranked proteins are reported, where k is a user specified parameter.

- Subtraction: an implementation of the subtraction strategy also with the scoring function in Samuelsson et al. [109]. The algorithm repeats k rounds to identify k proteins as output, where k is specified by the user. In each round, the masses matching the protein identified in the previous round are removed prior to the next round of searching.

The mass tolerance threshold is 1 ppm, and the results in Table 13 show that Losak and Losau achieve significant higher protein identification rate than previous PMF methods. Here the number of reported proteins for SPA, Subtraction, and Losak is 49, i.e., the number of ground-truth proteins. The decay factor df is set to 0.95 in Losau. As observed in the experiment, there are only 34 ground-truth proteins that match more than five peaks in candidate

selection process. On one hand, it indicates that the sequence coverage is insufficient during dataset generation (less than five peptides were identified for some proteins); on the other hand, it means that some mono-isotopic masses of ground-truth proteins are not assigned correctly.

Table 3.2: Identification performance of different algorithms on the real MS data

Algorithms	*Precision*	*Recall*	*F1-measure*	*Running time(s)*
SPA	24%	24%	24%	7.9
Subtraction	43%	43%	43%	24.0
Losak	67%	67%	67%	21.2
Losau	61%	71%	66%	19.6

3.5 PMF by the Order of Retention Time

One of the most common approaches for large-scale protein identification is High-Performance Liquid Chromatography (HPLC), followed by MS. The amount of time between the injection of a sample and its elution from the column (the elution is then identified by MS) is known as the *Retention Time* (RT).

RTs were seldom used for protein identification, while recently Bochet et al. [17] considered the order of peptide RT and applied quantile regression to identify proteins without experimental fragmentation spectra. Using a combination of highly accurate and precise mass measurements, modern retention time prediction, and a robust scoring algorithm, they were able to identify 257 proteins of Francisella tularensis from a single LC-MS experiment in a fragmentation-free approach. This number amounts to 59% of that of proteins identified in a standard fragmentation-based approach, when executed with same false discovery rate. Independent evidence supports at least 27 of a set of 31 proteins that were identified only in the fragmentation-free approach. Their results suggest in the future PMF methods may be an interesting complement or an alternative to fragmentation-based approaches.

HPLC peptide RTs can be used to identify peptides by comparing experimentally obtained RTs with those that have been predicted, or previously measured. In practice, it is hard to

predict the accurate experimental RT from the algorithm for wet laboratories (where chemicals, drugs, or other materials are handled in liquid solutions or volatile phases). Therefore, the method relies on the retention order of the peptides rather than their RT. By substituting the RTs by the retention order, a more robust characterization of the retention process is extracted. Furthermore, one does not seek to identify isolated peptides, but rather groups of peptides coming from the digest of the same protein. A peptide will be identified from the experimental data, only if other predicted peptides from the same candidate protein are also found in their predicted order. The proposed method identifies proteins in one single step and not by first finding peptides and then proteins.

The principle of the new method is illustrated in Figure 3.12, where intensities of the peaks are not considered. In this figure, the solid lines represent matching peptides, and the dashed lines represent non-matching peptides. For each protein in the relevant sequence database, one first predicts the peptides resulting from its enzymatic digest. Secondly, the HPLC RT predictions for the peptides are obtained and the peptides are sorted by predicted RT. Thirdly, one attempts to assign each predicted peptide for a given protein to a peak in a spectrum in the predicted order. The solution for this alignment problem is dynamic programming.

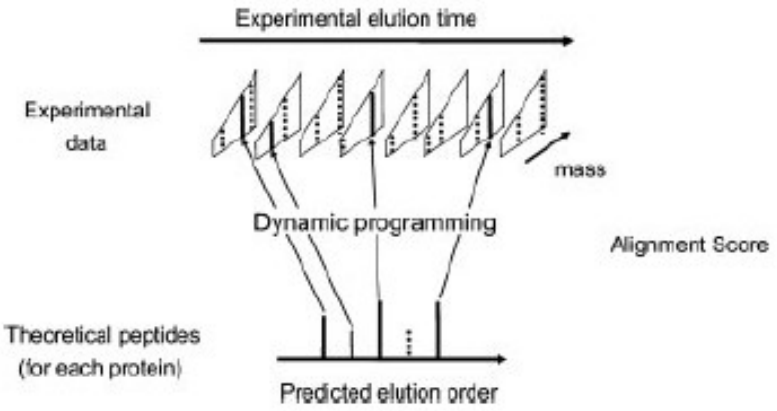

Figure 3.12: The principle of the identification method [17]. For each protein the predicted tryptic peptides are sorted according to their predicted retention time (bottom). They are then matched to ions from the spectra in the same order. Each trapezoid represents one experimental spectrum, and the masses of the peptides are symbolized by the height of the lines.

The completion in silico digest was performed on all proteins predicted by the database. Only peptides at least four amino acids long were kept. For each protein, these peptides were sorted according to their predicted order of retention from the HPLC column. Here the hydrophobic predictions in trifluoroacetic acid of the SSRCalc software [60] were used. For each protein, the algorithm described in Algorithm 5 computes the optimal alignment of the predicted peptides with the ions found in the experimental spectra. This maximizes the ordered peptide match (OPM) score. The score optimized in the alignment is defined as follows: the mass of each predicted peptide is compared with the closest mass in each spectrum after deisotoping and decharging. If Δ, the difference between the two masses is less than a given threshold, the contribution of the match to the score is +1, otherwise a penalty is counted (-1). This dynamic programming algorithm iteratively computes the score $s_{i,j}$ of the best alignments between a prefix $(p_i, \cdots, p_j)$ of the list of peptide masses and a prefix $(S_1, \cdots, S_i)$ of the list of spectra. Here the values of the no-peptide score and the no-spectrum score are 0 and 1, respectively.

To evaluate the statistical significance of the OPM scores, they were compared with the decoy scores whose proteins obtained by random permutation of the real sequences. Each of these decoy proteins had the same length, number of tryptic peptides, and amino acid composition as its naturally occurring counterpart. In the database, the number of proteins with the largest content of digest peptides was low. To obtain a sufficiently accurate estimate of the quantile limits for these large proteins (with more than 56 peptides), several permutations were performed for each of them.

Because the alignment scores depend on the number of tryptic peptides, and in the absence of a substantial theoretical justification for the use of a particular parametric distribution model, quantile regression were applied to describe the empirical distribution of scores for the decoy proteins. Here, quantile regression is a way to estimate the conditional quantiles of a response variable distribution in the linear model. Briefly, for each value of the variable quantile regression uses a weighting function to construct a local score distribution for the randomized

Algorithm 5: The order of peptide matching score.

Input: A protein p_g;
A list $(l_1, \cdots, l_m)$ of decharged, deisotoped spectra;
A list of masses $(u_1, \cdots, u_n)$ of tryptic peptides of p_g, ordered by predicted elution time;
A peak match threshold δ;
no-peptide score nps, no-spectrum score nss, and a peptide-spectrum match score ps
Output: maximum score for protein p_g;

```
ps ← 0;
for i := 1 to m do
    for j := 1 to n do
        if l_i has a peak in the interval (u_j − δ, u_j + δ) then
            set ps ← +1;
        end if
        else
            set ps ← −1;
        end if
        Compute s_{i,j} ← max(s_{i−1,j−1} + ps, s_{i−1,j} + nps, s_{i,j−1} + nss) under the convention
         that s_{i,j} = 0 if i = 0 or j = 0
    end for
end for
return s_{m,n}
```

proteins. The score of each real protein was compared with that local distribution for the same number of tryptic peptides, and the corresponding quantile was reported as the significance of the match. Figure 3.13 depicts the distribution number of matched peptides, where abscissa is the number of peptides longer than four amino acids per protein, and ordinate is the number of matched peptides.

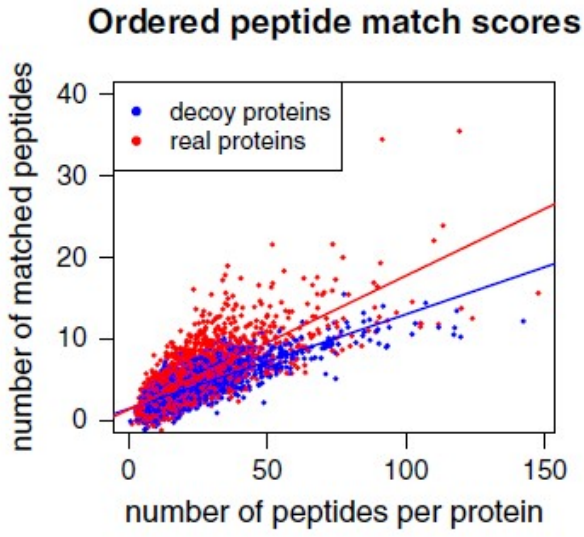

Figure 3.13: The distribution of matched peptide number with a HPLC-MS experiment [17]. Each point represents a protein from the sequence database. For visualization only, the positions of the points are shifted by a small gaussian random amount (σ=0.5). The lines represent the best linear fits for the data.

Chapter 4

Network-Based Inference Methods

When imagining the relationship between peptides and proteins as a bipartite graph, we may design new peptide protein matching score functions, based on network inference. Here we borrow the idea from recommender systems, hoping to discover important peptide-protein information from the projection of peptides.

In this chapter, we will present the concept of recommender systems, and formulate network-based inference (NBI) methods mathematically. We design two novel NBI methods for PMF: probabilistic spreading (ProbS), which is based on a probability framework; and heat spreading (HeatS), for a dynamic analysis. The experiments on simulated, authentic, and simulated authentic dataset demonstrate that our NBI approaches can improve the performance.

4.1 Recommender Systems

The rapid expansion of the Internet greatly increases the necessity of effective recommender systems to filter abundant information [79]. A recommender or recommendation system is a subclass of information filtering system that turns dataset on past user preferences into predict possible future likes and interests [139, 140]. Tremendous activity to the understanding of complex networks can facilitate researchers to design good recommender systems [141]. Bipartite networks are a particular class of networks, whose nodes are divided into two sets X and Y,

and only the connection between two nodes in different sets is allowed. Bipartite networks are illustrated in Figure 4.2 (a). Additionally, bipartite network projection can extract the hidden information of networks and do a personal recommendation, since people with similar hobbies would like the same items in the future [142]. Figure 4.1 illustrates a recommender system consisted of five users and four books, where some additional information frequently exploited in the design of recommendation algorithms, including user profiles, object attributes and object content. This network-based inference method can be used for mining potential information in other applications, such as speeding up drug repositioning [21], discovering undetectable proteins [99], and identifying cancer genes [131].

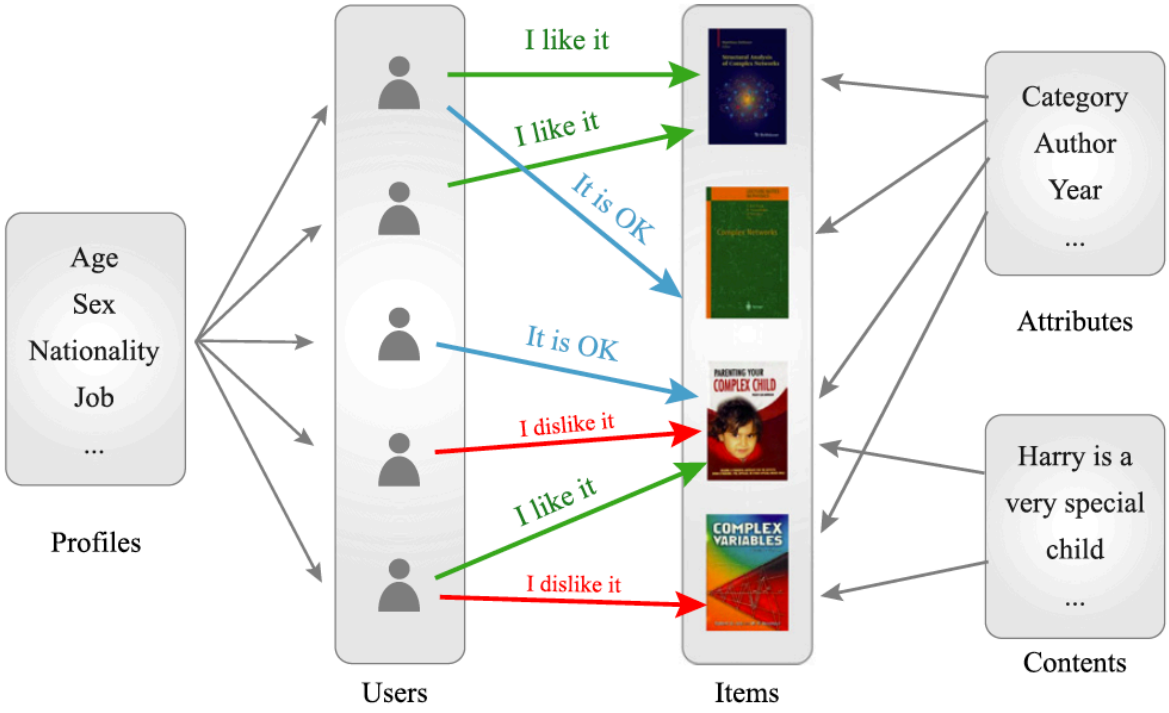

Figure 4.1: Illustration of a recommender system consisted of five users and four books [79]. The relations between users and objects that can be represented by a bipartite graph is the basic information contained by every recommender system.

The huge potential of recommender system was first noticed by web entrepreneurs in the forefront of the information revolution. While being originally a field dominated by computer scientists, it is now a topic of interest also for mathematicians, physicists, and psychologists.

Recommender system can be used for the Internet and electronic commerce, such as online shopping. A recommender system may have significant impact on a company's revenues: for example, 60% of DVDs rented by Netflix are selected based on personalized recommendations [79].

The input dataset is used by a recommender system to predict potential further likes and interests of its users. Users' past evaluations are typically an important part of the input data. Let M be the number of users and N be the number of all objects that can be evaluated and recommended. Note that object is simply as a generic term that can represent books, movies, etc. The goal of a recommender system is to deliver lists of personalized "recommended" objects to its users. Objects with the highest predicted ratings constitute the recommendation list that is presented to the target user. The usual classifications of recommender systems is as follows [79]:

- Content-based recommendations: recommended objects are those with content similar to the content of previously preferred objects of a target user.
- Collaborative recommendations: recommended objects are selected on the basis of past evaluations of a large group of users. They can be divided into:
  - Memory-based collaborative filtering: recommended objects are those that were preferred by users who share similar preferences as the target user, or, those that are similar to the other objects preferred by the target user.
  - Model-based collaborative filtering: recommended objects are selected on models that are trained to identify patterns in the input dataset.
- Hybrid approaches: these methods combine collaborative with content-based methods or with different variants of other collaborative methods.

There are some basic metrics used to measure the quality of recommendations, including: accuracy metrics, rank-weighted indexes, diversity and novelty, and coverage.

Here we will introduce two model-based collaborative filtering recommender systems, both are diffusion-based methods: probabilistic spreading and heat spreading. Before introducing these two approaches, we shall first present the basic knowledge for the one-mode projection of bipartite network, and object-object networks as well.

When analyzing and modeling bipartite networks, one-mode projecting can be extensively used to compress bipartite networks. One-mode projection has the convenience of directly showing the relations among a particular set of nodes. Since one-mode projection is always less informative than the bipartite representation, a proper weighting method is required to better retain the original information. The one-mode projection onto X means a network containing only X nodes, where two X nodes are connected when they have at least one common neighboring Y node. Figures 4.2 (b) and (c) show the resulting networks of X and Y projection, respectively.

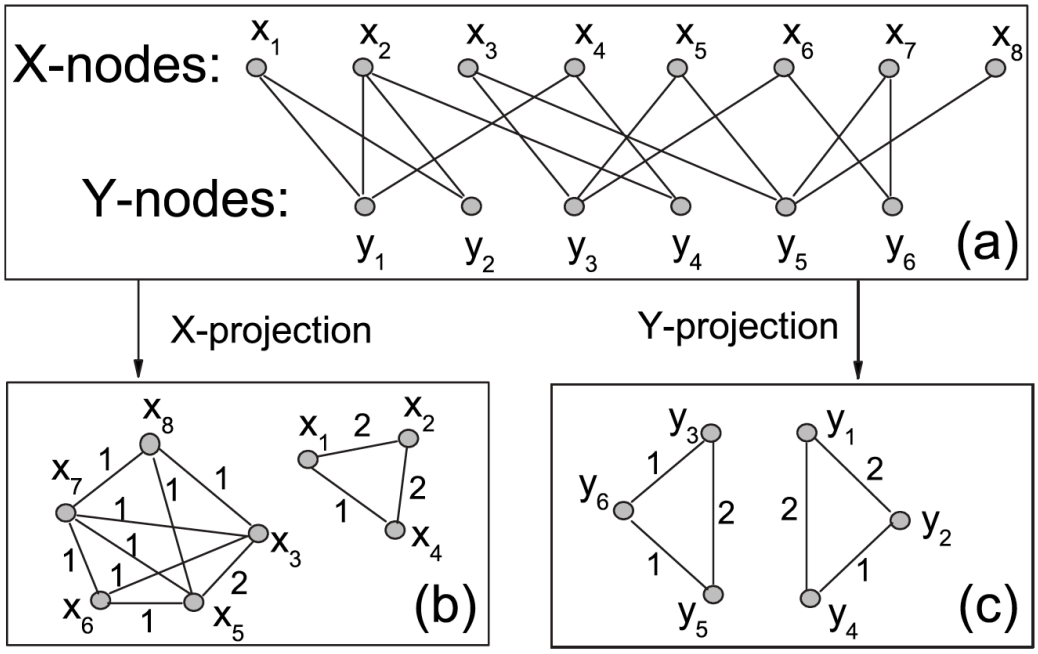

Figure 4.2: The illustration of a bipartite network (a), its X projection (b), and Y projection (c) [141]. The edge weight in (b) and (c) is set as the number of common neighbors in Y and X, respectively.

Basically, a recommendation system consists of users and objects, where each user has

collected some objects. In the web searching, the classical PageRank algorithm [15] could measure the importance of a website page by counting the number and quality of links to this page. In the recommender system, similarly as PageRank, one can obtain recommendations using a network representation of the input dataset with user preferences. Here we present two algorithms based on specific transformations of the input dataset to object-object networks. Personalized recommendations for an individual user are then obtained by using this user's past preferences as "sources" in a given network, and propagating them to yet unevaluated objects.

The first algorithm is probabilistic spreading (ProbS), which is suitable for dataset without explicit ratings, i.e. only the sets of object visited by each user is known. The spreading recommendation algorithm proposed in [141] is based on a projection of the input dataset (an unweighted user-object network) to an object-object network. In this projection, we would like to determine the edge weight in X projection. The weight $W_{\alpha\beta}$ can be considered as the importance of node α with respect to node β, and it differs from $W_{\beta\alpha}$ generally. For example, a book projection can be produced from a customer-book opinion network. Additionally, the weight $W_{\alpha\beta}$ between two books α and β, demonstrates the strength of book α recommendation to a customer who has bought book β. A suitable form of $W_{\alpha\beta}$ can be initialized by studying the original bipartite network, where a certain amount of a resource is assigned to each object node. For example, for α, the value of '1' represents a person has bought book α, and '0' represents this person have not bought book α yet. Since it is an unweighted network, the unbiased allocation of the initial resource is split equally among all its neighboring customer-nodes. Consequently, resources collected by customer-nodes are equally redistributed back to their neighboring book-nodes. This is equivalent to a random walk, from the initial source nodes to a distance of two in the customer-book bipartite graph. An illustration of this resource-allocation process for a simple bipartite network is shown in Figure 4.3. In plots (a)-(c), the upper three are X-nodes, and the lower four are Y-nodes. After two steps of flow, a weighted projection of a bipartite network will be produced. Here the weight located on the directed

edge $A \rightarrow B$ means the fraction of resource node A would transfer to node B. The weights of self-connections are also labeled besides the corresponding nodes [142].

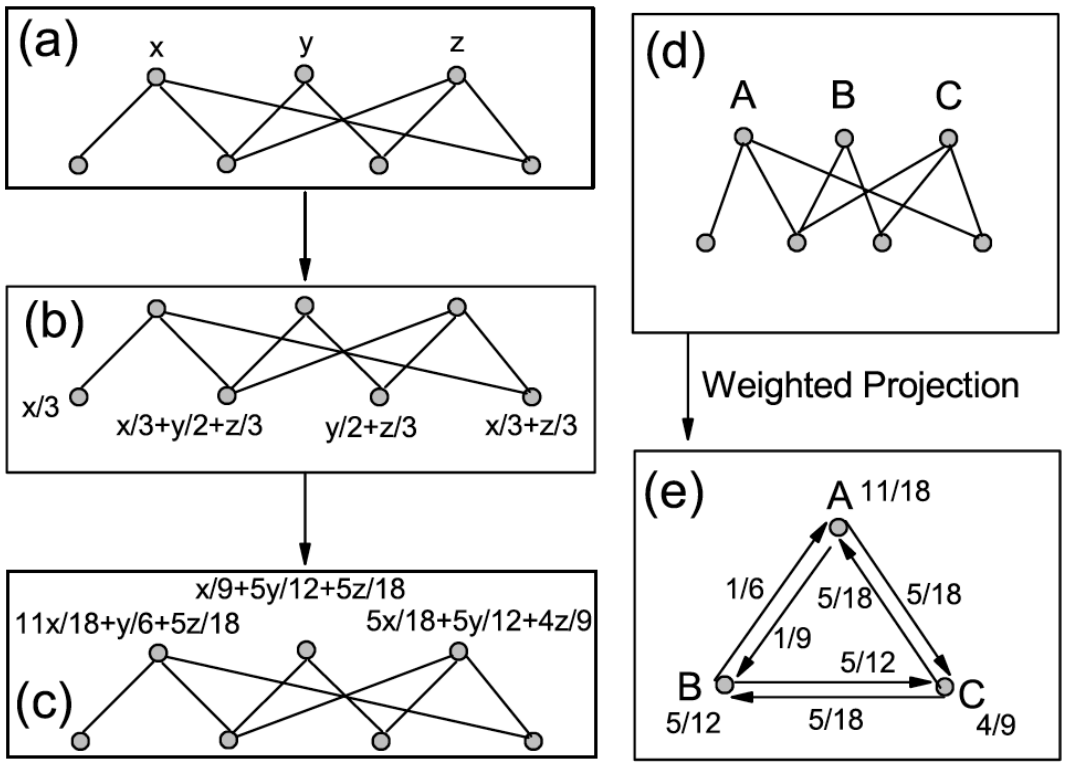

Figure 4.3: Illustration of the resource-allocation process in a bipartite network [142]. The resource first flows from X to Y ($a \rightarrow b$), and then returns to X ($b \rightarrow c$). The process from (a) to (c) can be considered as a weighted projection of a bipartite network, shown as ($d \rightarrow e$).

The second algorithm is heat spreading (HeatS), which can answer the need of diversity in algorithm-based recommendation. A hybrid algorithm was proposed in [140] which combines accuracy-focused ProbS with diversity favoring heat spreading. As in probabilistic spreading, heat spreading works by assigning objects an initial level of "resource" denoted by a vector, and then redistributing it via the transformation. In contrast to ProbS, HeatS is row-normalized and corresponds to a heat diffusion process on the given user-object network.

Figure 4.4 illustrates the procedures of ProbS and HeatS. The objects with initial resource '1' means they are collected by the target user. For example, the books bought by a person are '1s' (including first and fourth), and those '0s' are books not bought yet (including second,

third, and fifth). Consequently, we would like to recommend a book for this person among the '0s'. According to the final scores, ProbS will recommend the third object to the target user (third has the maximum score among the initial resources '0s'), while HeatS will recommend the second one (second has the maximum score among the initial resources '0s'). Generally speaking, ProbS tends to recommend popular objects and thus lacks diversity and novelty, while HeatS is able to find out unpopular (i.e., of low degree) objects. On the other hand, recommendations obtained by HeatS are too peculiar to be useful.

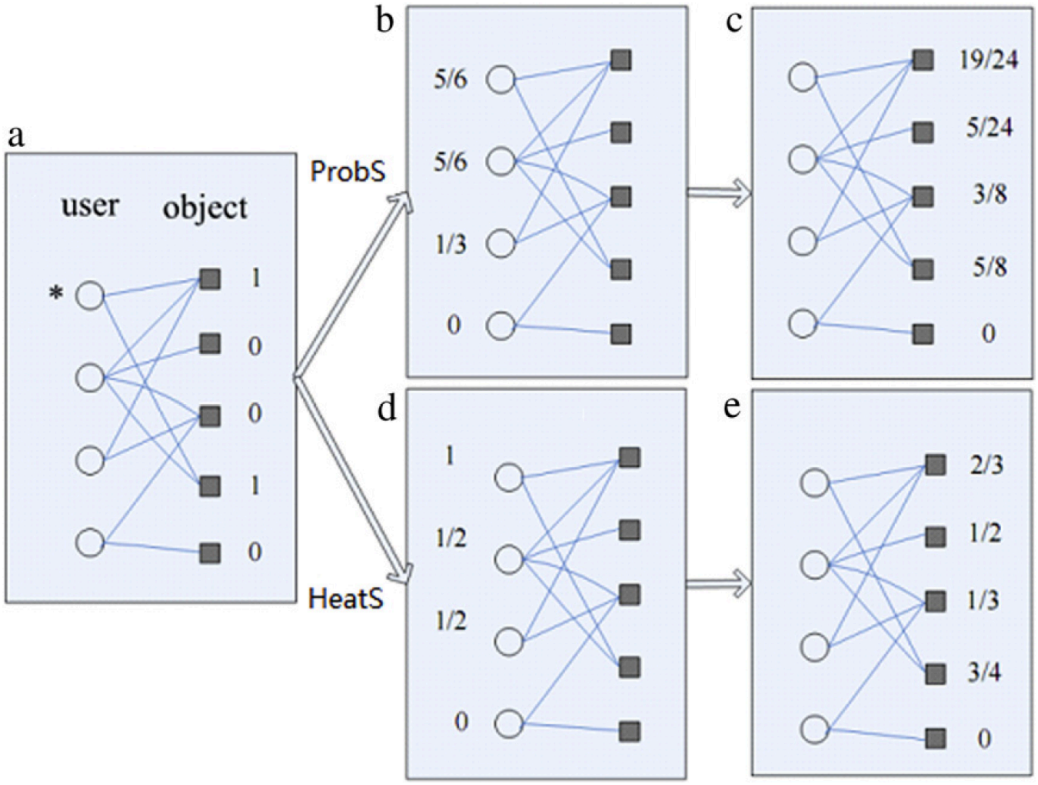

Figure 4.4: The comparison of ProbS and HeatS [79]. The target user is marked by a star and the collected objects are of initial resource 1. The final scores after ProbS and HeatS are listed in the right sides of plots (*c*) and (*e*).

4.2 Preliminaries for Network-Based Inference (NBI)

We shall use the following definitions and notations in this thesis.

With the protein database D, we can digest it with trypsin, and obtain a peptide database denoted as $T = \{u_v \mid v = 1, 2, \ldots, w\}$, where u_v denotes a peptide, and w is the size of this database. Additionally, we can describe the protein-peptide associations as a bipartite DT graph $G(D, T, E)$, where $E = \{e_{jv} \mid p_j \in D, u_v \in p_j, u_v \in T\}$. There is an edge if $u_v \in p_j$. Therefore, the DT bipartite network can be presented as a $s * w$ adjacent matrix $\{b_{jv}\}$, where $b_{jv} = 1$ if p_j and u_v are linked, otherwise it is 0 if there is no relationship between them.

The network-based inference (NBI) method we use here was first developed by Zhou et al [141] for personal recommendation, and there are mainly two types: ProbS [141] and HeatS[140].

Given above protein-peptide bipartite network, ProbS works by assigning peptides an initial resource, such as those selected objects. We also denote $f_p(u_v)$ as the initial resource allocated to the vth peptide in database T, the superscript p stands for "probabilistic." Here we give all of $f_p(u_v)$ as 1, then ProbS can be illustrated as a resource-allocation process. In the first step, all resources in peptide set T flow to protein set D according to

$$f_p(p_j) = \sum_{v=1}^{w} \frac{b_{jv} f_p(u_v)}{k(u_v)} = \sum_{v=1}^{w} \frac{b_{jv}}{k(u_v)},$$

where $k(u_v)$ is the degree of node u_v. Subsequently, all resources in protein set D return similarly back to peptide set T. Consequently, the final resource allocates to u_t is calculated as

$$f'_p(u_t) = \sum_{j=1}^{s} \frac{b_{jt} f_p(p_j)}{k(p_j)}.$$

Here, the intuitive meaning of this score is the calculation of the popularity for each peptide.

If a peptide only belongs to one protein, it is a unique peptide. When the degree of a peptide is large, and if the degrees of its connected proteins are also large, it may be a popular peptide. After the calculation, the final score for each peptide would be different from the initial score of 1, but the summation of them still equals to $|T|$. By redistribution of the scores, for unique peptides, they contribute more to others ($f'_p(u_t) < 1$); while for those popular ones, they absorb

more from others ($f'_p(u_t) > 1$).

Similar to ProbS, we can define HeatS score as the follow:

$$f_h(p_j) = \sum_{v=1}^{w} \frac{b_{jv} f_h(u_v)}{k(p_j)},$$

$$f'_h(u_t) = \sum_{j=1}^{s} \frac{b_{jt} f_h(p_j)}{k(u_t)}.$$

In contrast to $f'_p(u_t)$, $f'_h(u_t)$ is row-normalized (ipsative scaling, rescaling the score by dividing the total degree of this node) and corresponds to a heat diffusion process (or power iteration algorithm) on the given protein-peptide network [140]. Here $f_h(u_v)$ can be 1 or 0, depending on the input data, so it is a dynamic model. After calculations, for the nodes with $f_h(u_v) = 1$, their $f'_h(u_t)$ would be in (0,1]. But unlike ProbS, usually the HeatS scores for unique peptides should be greater than those popular ones.

4.3 Probabilistic Spreading Algorithm (ProbS)

In contrast to MOWSE, we do not use probability for each peptide. Instead, we would like to design a new scoring system. In the protein inference problem [96, 113, 53, 138], we can imagine that the relationship between identified peptides and proteins in a database is a bipartite graph [80]. Figure 4.5 demonstrates this relationship. Additionally, we may infer correct proteins from this bipartite graph [133], although it is still a difficult problem [2]. Now considering PMF, we can depict the relationship between peptides from proteins based on network technology.

The main pipeline of ProbS is illustrated in Figure 4.6. We initialize all of peptides with 1, and the total number of resources is $|T|$. After two flows, all the values for each peptide will be modified depending on its uniqueness or popularity. A hypothesis is that the contributions of different peptides should not be uniform for protein identification. Those unique peptides

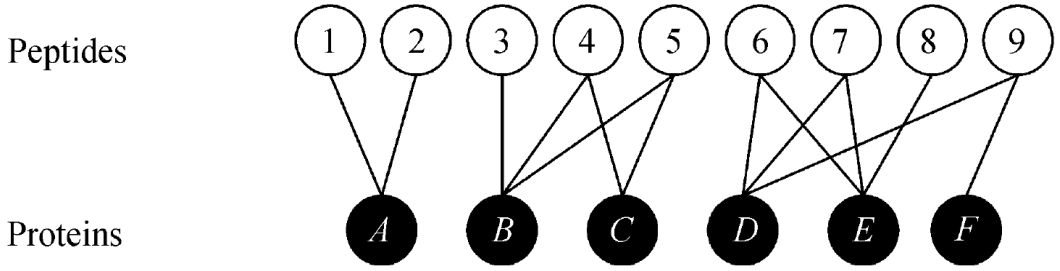

Figure 4.5: Each peptide may be contained by multiple proteins, resulting in a bipartite graph that is hard to resolve [80].

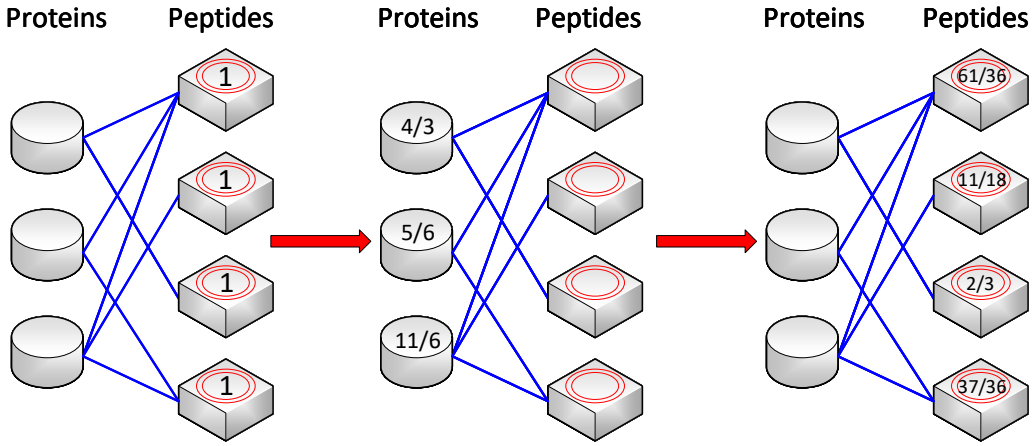

Figure 4.6: The flowchart of the method ProbS.

could contribute more than the popular ones. Otherwise, a naive scoring function could be only counting the matched numbers for each protein.

We denote K as the total degrees of all the nodes in T. Consequently, $K = \sum_{v=1}^{w} k(u_v)$ and $k(u_v)/K$ can represent the probability of the degree of node u_v among total degrees. Therefore, the scoring function for ProbS can be

$$F_P(P, p_j, \delta) = \sum_{q=1}^{t} \frac{f'_p(u_q)}{k(u_q)/K} = K \sum_{q=1}^{t} \frac{f'_p(u_q)}{k(u_q)},$$

where $f'_p(u_q)$ is the ProbS score of matched peptide u_q, and t is the total number of matches for

p_j. Since K is a constant, we can use

$$F_P(P, p_j, \delta) = \sum_{q=1}^{t} \frac{f'_p(u_q)}{k(u_q)} = \sum_{q=1}^{t} \frac{f'_p(u_q)}{k(u_q)},$$

to calculate the score in practice. Additionally, since we initialize all the peptides with 1, then $f'_p(u_q)$ does not depend on the experimental peaks P, and we can calculate it before processing the input dataset. But the total number of matches (t) still depends on the experimental peaks P. Here, $f'_p(u_q)/k(u_q)$ is from 0 to 1, and those larger scores are important to identify proteins. From this function, we know that ProbS is similar to MOWSE, which is also based on probability. The ProbS framework for PMF is shown in Algorithm 6.

Algorithm 6: ProbS for PMF: a set of proteins best explaining experimental peaks.

Input: D: a database of s proteins;
P: an observed peak list;
δ: a mass tolerance threshold;
$U(p_j)$: a set of matched peptides for protein p_j;
$f'_p(u_q)$: a matching ProbS score for a peptide $u_q \in p_j$;
$k(u_v)$: the degree of node u_v;
Δ: a threshold score ;
Output: Z: a set of proteins $Z \in D$ that best explaining P;

```
for j := 1 to |D| do
    F_P(P, p_j, δ) ← 0
    for q := 1 to |U(p_j)| do
        F_P(P, p_j, δ) ← F_P(P, p_j, δ) + f'_p(u_q)/k(u_q)
    end for
end for
Z ← {p_j | p_j ∈ D, F_P(P, p_j, δ) ≥ Δ}
return Z
```

4.4 Heat Spreading Algorithm (HeatS)

Another NBI method is HeatS. In recommender systems, ProbS is popularity-focused while HeatS favors uniqueness, as we mentioned before. As in probabilistic spreading, HeatS works by assigning peptides an initial level of resource, then redistributing it via the transformation.

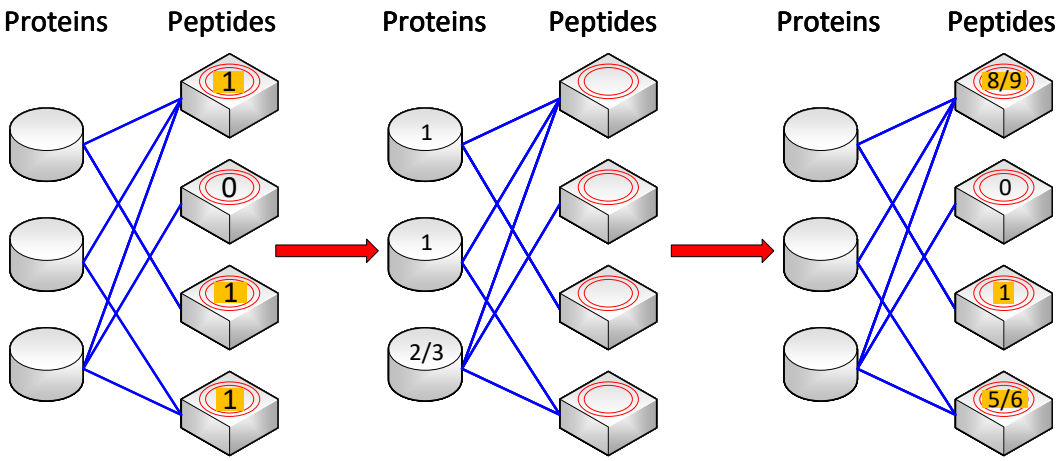

Figure 4.7: The flowchart of the method HeatS.

The main pipeline of HeatS is illustrated in Figure 4.7. In our method, only the peptides that matched with the peaks will be assigned to 1, and unmatched to be 0. Consequently, the score for a protein is the matched peptides divided by the degree of this protein, which can be considered as the coverage for this protein. For example, in Figure 4.7, the score for the third protein is $2/3$, which means there are two peptides matched among the total three peptides. Unlike ProbS, HeatS should compute simultaneously because the assignment of the peptides depends on the input dataset of experimental peaks P. After calculations, the scores for matched peptides will be changed, usually the scores of popular peptides would be smaller than that of unique ones. Additionally, we ignore the scores for those peptides that not matching.

HeatS performs better than other methods in our testings. The reason for that is after the first heat spreading, proteins with higher scores tend to be the real ones in the experiment. In the second flow, they give high scores to their corresponding peptides. Because peptides from the same protein are dependent on each other, they usually obtain high scores from peptides co-operation. By decreasing the scores for other random matches, the ground-truth proteins have more chances to be on the top. Our experimental results also demonstrate this phenomenon.

The scoring function for HeatS can be

$$F_H(P, p_j, \delta) = \sum_{q=1}^{t} f'_h(u_q),$$

where $f'_h(u_q)$ is HeatS score of matched peptide u_q, t is the total number of matches for p_j. The HeatS framework for PMF is shown in Algorithm 7.

Algorithm 7: HeatS for PMF: a set of proteins best explaining experimental peaks.

Input: D: a database of s proteins;
P: an observed peak list;
δ: a mass tolerance threshold;
$U(p_j)$: a set of matched peptides for protein p_j;
$f'_h(u_q)$: a matching HeatS score for a peptide $u_q \in p_j$;
Δ: a threshold score ;
Output: Z: a set of proteins $Z \in D$ that best explaining P;

```
for j := 1 to |D| do
    F_H(P, p_j, δ) ← 0
    for q := 1 to |U(p_j)| do
        F_H(P, p_j, δ) ← F_H(P, p_j, δ) + f'_h(u_q)
    end for
end for
Z ← {p_j | p_j ∈ D, F_H(P, p_j, δ) ≥ Δ}
return Z
```

4.5 Experiments

Recall that we have an input set of experimental peaks $P = \{l_i(m_i, r_i) \mid i = 1, 2, \ldots, n\}$, where l_i is a peak, m_i is the mass of l_i, r_i is the corresponding RT, and n is the total number of peaks. We also have a database of proteins $D = \{p_j \mid j = 1, 2, \ldots, s\}$, where p_j denotes a protein, and s is the size of this database. For PMF, our objective is to find a set of proteins $Z \in D$ that contains all the proteins whose scores are greater than a threshold Δ, i.e. $Z = \{p_j \mid p_j \in D, F(P, p_j, \delta) \geq \Delta\}$.

We use three datasets to show the superiority of our approaches compared to others. We also consider factors that affect the performance. The source code of the program consists of

approximate 45,000 lines of Java code, and all the answers for different approaches can be produced within several seconds. Here we choose Java since it has some advantages, such as it is object-oriented, and platform-independent. Because of Java's robustness, ease of use, cross-platform capabilities and security features, it has the ability to move easily from one computer system to another. We choose Eclipse as our integrated development environment (IDE) for Java programming, since Eclipse is the most widely used Java IDE.

In order to evaluate different algorithms, we generate three datasets: simulated dataset, authentic dataset, and simulated authentic dataset. We also test these datasets in the following chapters.

4.5.1 Evaluation Criteria

A protein in output Z is counted as a true positive (TP) if it belongs to ground-truth proteins, otherwise as a false positive (FP). Here, we could use standard performance metrics in information retrieval, including

precision, *recall*, and *F1-measure*, to evaluate identification performance. Their definitions are as follows:

- n_{TP}: the number of true positives.
- n_{FP}: the number of false positives.
- n_P: the number of all ground-truth proteins.
- $precision = n_{TP}/(n_{TP} + n_{FP})$: the proportion of identified ground-truth proteins to all identified proteins.
- $recall = n_{TP}/n_P$: the proportion of identified ground-truth proteins to all ground-truth proteins.
- $F1-measure = (2*precision*recall)/(precision+recall)$: the harmonic mean of recall and precision.

Usually *precision* and *recall* are not the same for different thresholds. In practice it is difficult to find a perfect Δ for reporting all identified proteins. In order to compare different methods, in the experiments we choose Δ to be n_P, since we know the n_P before the searching. Therefore, n_P and $(n_{TP} + n_{FP})$ would be the same under this situation. Consequently, *recall*, *precision*, and *F1-measure* would be the same. Another solution is choosing different Δ to count n_{TP} and n_{FP}, then *recall*, *precision*, and *F1-measure* will be different. Here, in order to compare with other state-of-the-art methods, we simply use n_P as our Δ.

4.5.2 Processing of Database

We use the newest *Swiss-Prot* database in *UniProt*. Swiss-Prot is the manually annotated and reviewed section of the UniProt Knowledgebase (UniProtKB). Since 2002, it is a high quality annotated and non-redundant protein sequence database, which brings together experimental results, computed features and scientific conclusions [154]. For Human, we have 20,161 canonical entries and 21,956 additional isoform proteins. After theoretical tryptic digestion without missed cleavages, there are total 699,457 different theoretical peptides generated from this Human protein database. The cleavage rule for trypsin is: after *R* or *K*, but not before *P*. That is, the trypsin cleaves the protein sequence after each *K* or *R*, unless *K* or *R* is followed by a *P*. For example, cleavage of the sequence "MVPPSKGGAKPLGRSLGPLLLRPEEP" should result in these 3 peptides: "MVPPSK", "GGAKPLGR", and "SLGPLLLRPEEP".

In order to compare different methods fairly, we put a canonical protein and its isoforms into one group. If any protein in a group is identified, we say this group is identified. Additionally, we always select the protein with highest score in this group as its representative [39]. Also for NBI methods, we treat a canonical protein and its isoforms as one group, and use groups as the nodes in the network. Consequently, the size of the bipartite graph will be decreased, and the impact of isoforms would be avoided, since we do not calculate the identical peptides from the isoforms repetitively.

In order to count the identified proteins precisely, we also combine the proteins with dif-

ferent names but same sequences to be one protein. For example, P0DMV8 and P0DMV9, they have identical protein sequences, so we combine them as one group. When reporting the result, we treat the proteins with high percentage of identical peptides as a protein group. Here we choose the threshold of 0.8, which means for two proteins, if 80% of their matches are the same, we report them as a protein group.

4.5.3 Testing Datasets

In order to evaluate the performances of different methods, we use three datasets to test them. These datasets are generated from different methods, including: simulated dataset, authentic dataset, and simulated authentic dataset. Since it is not easy to get the suitable MS dataset from online and wet-lab, besides the authentic dataset, we also design algorithms to generate the simulated dataset, and the simulated authentic dataset, from a liquid chromatography and tandem mass spectrometry (LC-MS/MS) dataset.

Here, simulated dataset has the information for identified peptides (mass and RT), while simulated authentic dataset has the raw dataset which contains peaks information for the identified peptides. In this thesis, all the results of the approaches are based on these three datasets, which are introduced in the following subsections. After processing, there are a certain number of peaks in each dataset. Additionally, each peak includes two floating point numbers: mass and RT of a peptide.

Dataset One

Dataset one is a simulated dataset, which is generated from a LC-MS/MS dataset. In the wet-lab, we perform experiments with a Thermo Scientific Orbitrap Elite mass spectrometer, which has a dual-pressure linear ion trap and a high-field Orbitrap mass analyzer. This ion trap contributes enhanced ion optics that increase sensitivity and reliability, a greater dynamic range for better precursor detection, and the power of MSn identification. The high-field Orbitrap mass analyzer geometry and advanced signal processing technologies enable resolution

of >240,000. The availability of multiple fragmentation techniques (CID, HCD and optional ETD) offers versatility for challenging research applications [145].

In the wet-lab with LC-MS/MS, we generated a sample of protein complex for Human, which includes protein *telomeric repeat-binding factor 1* (TERF1). TERF1 is a protein that encoded by the TERF1 gene, which is located in the chromosome 8, from 73,921,097 to 73,960,357 base pair. TERF1 gene encodes a telomere specific protein which is a component of the telomere nucleoprotein complex. This protein is present at telomeres throughout the cell cycle and functions as an inhibitor of telomerase, acting in cis to limit the elongation of individual chromosome ends [156].

We searched this HCD MS/MS spectra with Mascot, by using the following parameters: trypsin digestion, no PTM, monoisotopic peaks, any charge state, and unrestricted protein mass. There are altogether 248 identified proteins in Human database. From this protein candidate list, we selected the protein that at least has four matched peptides. Consequently, we had 89 proteins that satisfied this rule. For each of these 89 proteins, we obtained the information for the matched peptides from Mascot, including the mass and RT. From these 89 proteins, we selected the masses and RTs from the matched peptides as the simulated MS spectra, and the total number of the peaks was 770. Finally, we set our simulated MS dataset to contain these 770 matched peptides, and these 89 proteins are considered as our ground-truth proteins. For each line of the simulated dataset, there are two numbers that represent a peptide: mass and RT.

Dataset Two

Dataset two is from an authentic dataset. In 2005, a proteomics standard composed of 49 highly purified human proteins in an equimolar mixture was created by the ABRF Proteomics Standards Research Group (sPRG). The sPRG conducted a blind study to assess the proteomics communitys ability to determine the identities of the constituent proteins using their proteomics platforms of choice. 120 laboratories from across the world volunteered to participate. Each

	Peptide Identification	Protein Inference	Decoy DB Assessment	Final Validation
Kowalak	Mascot	MassSieve	Run but not applied as filter	Manual
Lane	SEQUEST	Proteomics Browser Suite	spectral level, <1% FDR	Required 3 distinct peptides
Nesvizhskii	SEQUEST	Protein Prophet	Run but not applied as filter	Protein prob>0.95
Searle	Mascot	Scaffold	Run but not applied as filter	Protein prob>0.95, 2 pept. prob>0.95
Seymour	Paragon	Pro Group	protein level, <5% FDR	Pro Group conf>99%
Tabb	MyriMatch	IDPicker	spectral level, <5% FDR	Required 3 distinct peptides

Figure 4.8: A diversity of analysis tools were used by six BIC members.

laboratory received a complex mixture of 49 unknown proteins and were asked to identify as many of these proteins as possible using their best analytical strategies. Consequently, 78 laboratories returned their results. The study revealed the value of multiple independent analyses of this proteomics standard, and contributors were asked to voluntarily contribute their datasets for future public distribution. Finally, 24 raw datasets were submitted representing a wide variety of proteomics strategies and mass spectrometry platforms. Format conversion utilities were used to create peak lists to .mgf, mzData, and mzXML, allowing a total of 19 of the 24 datasets to be examined ultimately by the sPRG Bioinformatics Committee (BIC). These selected datasets included: dataset from all major vendors' instruments; ESI and MALDI, 1D-LC, 2D-LC, and gel workflows. Only the MS/MS dataset were considered. Consequently, all six BIC members searched the datasets using different software tools for peptide identification and protein inference, which is shown in Figure 4.8.

After initial explorations, a FASTA database was assembled for final searches as the human component of Swiss-Prot, plus 39 non-human and 5 non-Swiss-Prot human candidates (15,681 total proteins). For final searches, each BIC reported a count of confident peptide sequences

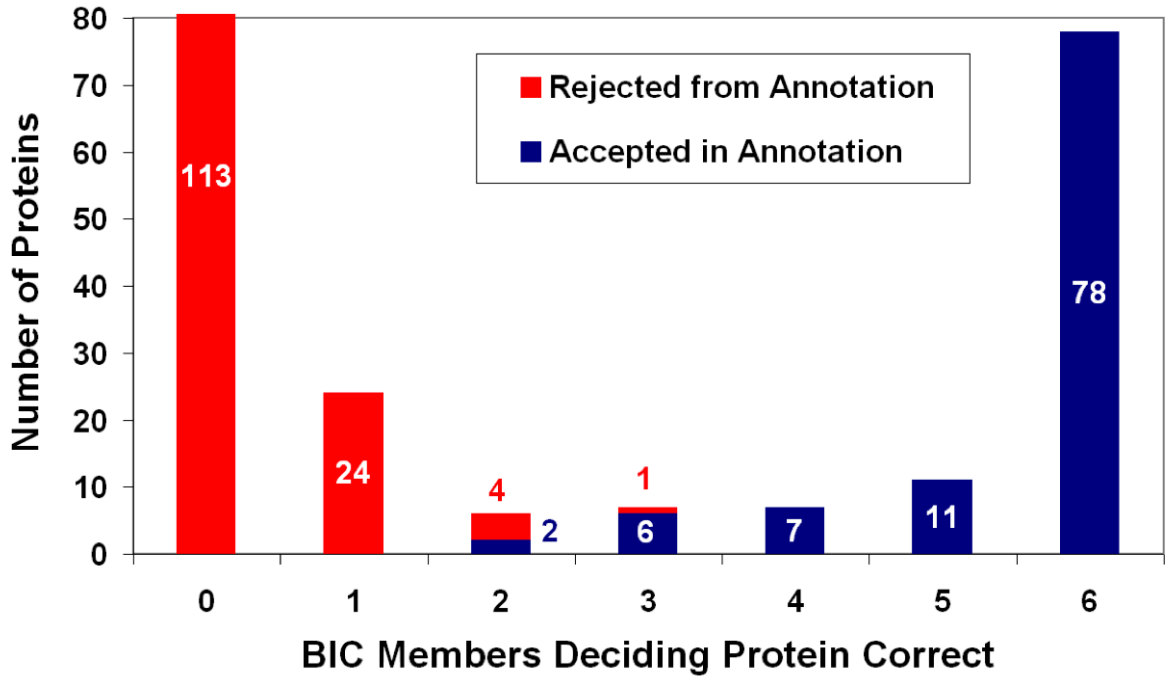

Figure 4.9: BIC consensus on protein identifications [64].

and protein probability or binary decision. There were 84 final searches on datasets from 19 laboratories by six BCI members, and Figure 4.9 demonstrates a histogram of the degree of consensus among each BIC members' final decisions about which proteins were and were not detected. From this figure, 78 proteins were declared by all six BIC members and 89 were declared by at least five or more. Only 15 proteins ultimately considered correct had less consensus than this – in some cases because not all BIC members searched the lab where a protein was implicated. The summary of BIC consensus protein annotation is shown in Table 4.1.

For these 19 datasets, not everyone counts peptides the same way, so the dataset from each BIC member were normalized to the mean across the group. Using the annotation, BIC re-grade sPRG2006 protein lists, and the picture changes slightly as some of 2006's putative bonus proteins were confirmed, Figure 4.10 shows this situation.

From above steps, we know that although the ABRF protein mix was originally intended to contain 49 proteins only, ABRF sPRG BIC have confirmed the presence of further 20 'bonus'

Table 4.1: The summary of BIC consensus protein annotation [64].

Total proteins detected in any lab's data	104
Expected: Intended 49 proteins	49
Expected: Additional human proteins	20
Expected: E.coli	3
Total desirable protein detections (originally in sample)	72
Expected: Digestion enzyme	2
Variable: Human keratins	12
Variable: Sheep proteins (from wool)	5
Variable: Trypanosome proteins	9
Variable: E.coli proteins	4
Total other detections (avoidable artifacts)	30

proteins in the mix [64].

In this thesis, we use an authentic MS dataset obtained from a mixture of 49 standard human proteins in the ABRFsPRG2006 study, which was also tested by He et al [50]. The dataset was generated by a linear ion trap-orbitrap (LTQ-Orbitrap) mass spectrometer. Additionally, we consider identifications of all expected and 'bonus' proteins to be ground-truth proteins [10].

The raw dataset for ABRFsPRG2006 study can be processed by third party software, such as MZmine 2 [105]. MZmine 2 is an open-source software for mass-spectrometry dataset processing, with the main focus on LC-MS dataset. Figure 4.11 shows the result of the raw dataset processed by MZmine 2, there are tree views in the left window: the blue ones are MS, while the red ones are MS/MS. In the right window, a MS scan #8,024 is displayed with some peaks. We first employed Decon2LS [56] to find peaks in individual mass spectra from raw LC-MS data. DeconTools is a software package used to de-isotope mass spectra and detect features in MS dataset by using the isotopic signatures of expected components. Consequently, VIPER [92] is used to identify LC-MS peaks from MS peaks across elution time. Figure 4.12 shows the result of the raw dataset processed by VIPER, vertical axis represents the mass, and the horizontal axis is the retention time. Here we used the default parameter settings for Decon2LS and VIPER, except for the threshold of intensity for peaks peaks (only consider the peaks whose intensities are greater than 1E4). The mass range is from 800 to 4,100 Da. In

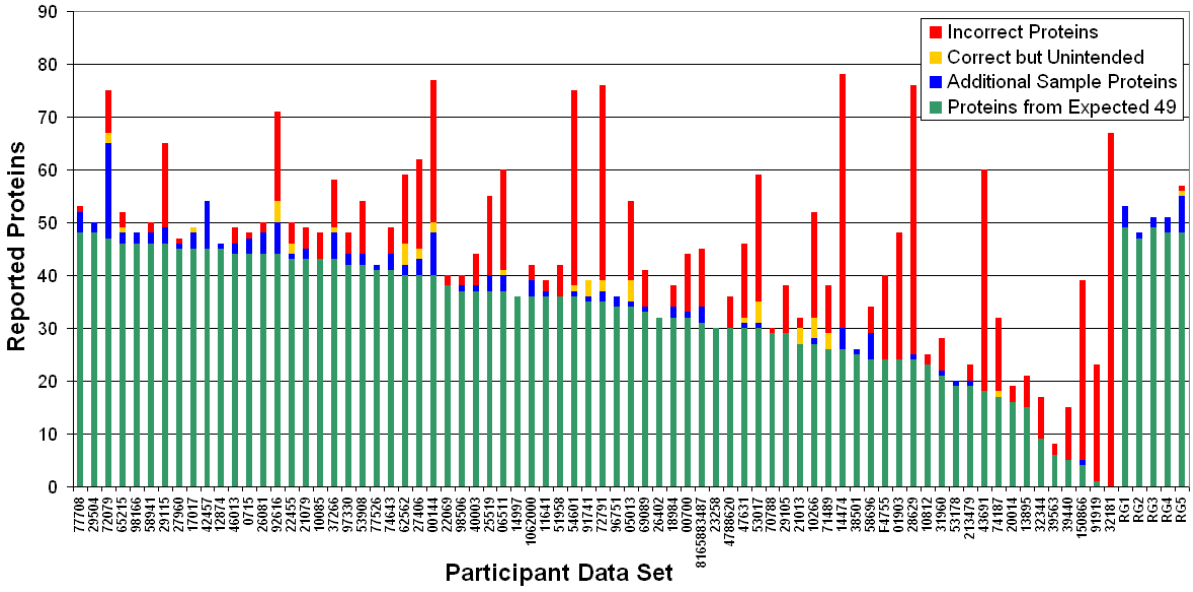

Figure 4.10: The re-grading of sPRG2006 protein lists [64].

total, we obtained 9,976 de-convoluted peaks as the input dataset.

We also tested other input datasets with different intensities, such as 5E4 and 1E5, then the number of peaks are 6,812 and 3,996 respectively. For 1E5, mass ranges from 800 to 4,200, and charge less than 3, we have Figure 4.13 that processed by VIPER, which is much spare than Figure 4.12.

Dataset Three

Dataset three is from a simulated authentic dataset. Now, we only have datasets with one hundred proteins, but we would like to know what is the maximum protein number PMF could identify with satisfying precision. In wet-lab experiments, the number of proteins in complex samples is usually several thousands and is identified by MS/MS spectra with PFF. Therefore, we set our goal to identify several hundred proteins with PMF. Unfortunately, it is difficult to find a suitable sample with only several hundred proteins from online and wet-lab resources [32]. Consequently, we designed a new method to generate MS spectra partially from the raw dataset for thousands of proteins, assuming this MS dataset is from hundreds of proteins.

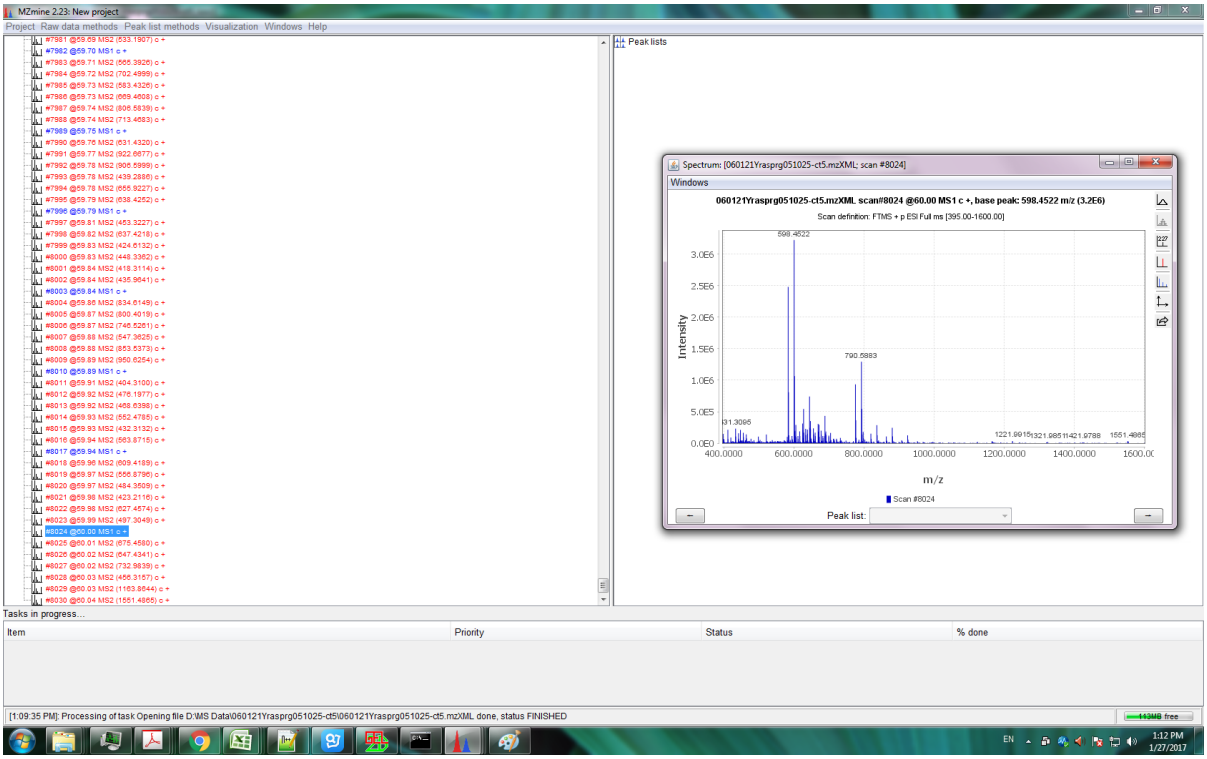

Figure 4.11: The display of the raw ABRFsPRG2006 dataset processed by MZmine 2. In the left window, the blue ones are MS spectra, while the red ones are MS/MS spectra. In the right window, #8,024 is a MS scan and there are some peaks within it.

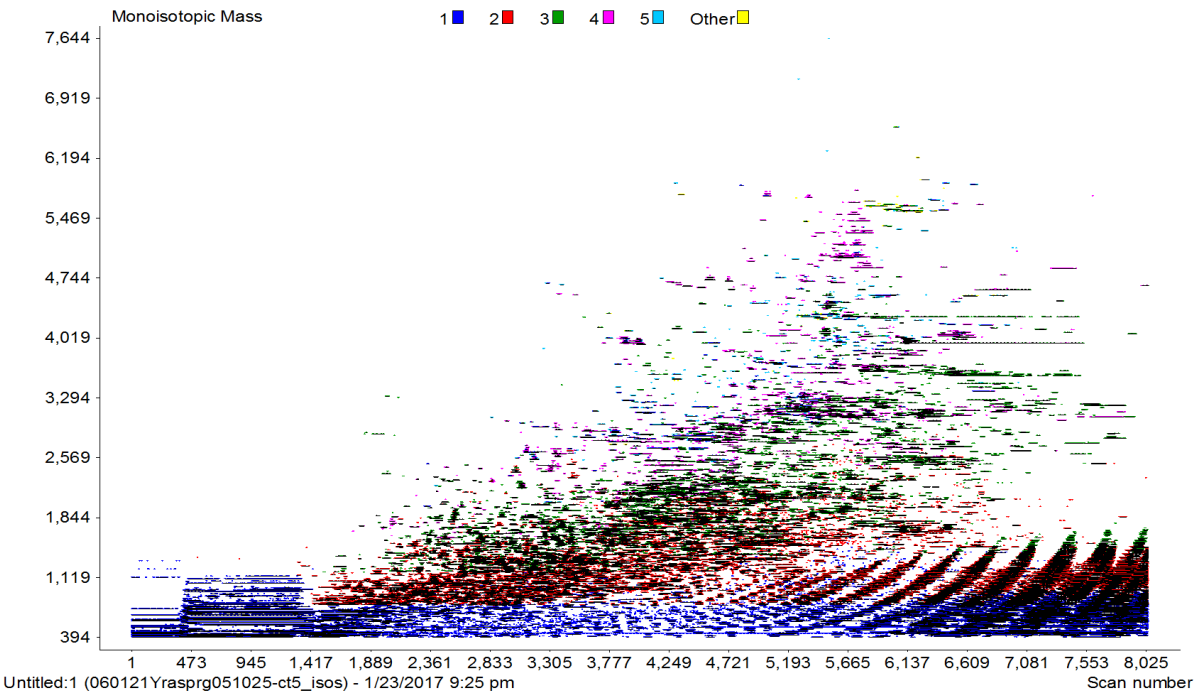

Figure 4.12: The display of the raw ABRFsPRG2006 dataset processed by VIPER. The horizontal axis represents the retention time, and the vertical axis is the mass.

We performed a LC-MS/MS experiment for 2,000 Human proteins by Orbitrap Elite. Now we would like to select partial MS spectra which could represent five hundred proteins from this MS/MS dataset. We first used PEAKS software (Ma et al., 2003) to identify the proteins, by the following parameters: parent mass error tolerance mass tolerance (2 ppm), fragment mass search type (0.02 Da), precursor mass search type (monoisotopic), enzyme (trypsin), max missed cleavage (0), non-specific cleavage (none), max variable PTM per peptide (0), searched entry (42,164), and database (UniProt Human), etc. After searching, we created a protein candidate list, which contains 1,606 identified proteins. We selected the top 519 proteins, since there is a distinct difference between the scores at this position. From those proteins, we selected their corresponding 5,686 identified peptides. After removing those same peptides that matched different proteins, 5,158 ones left. Consequently, we employed Decon2LS (Jaitly, 2009) to find peaks in individual mass spectra from raw LC-MS/MS data. There were two files generated after the processing of Decon2LS, one contains the information for all 23,380 MS

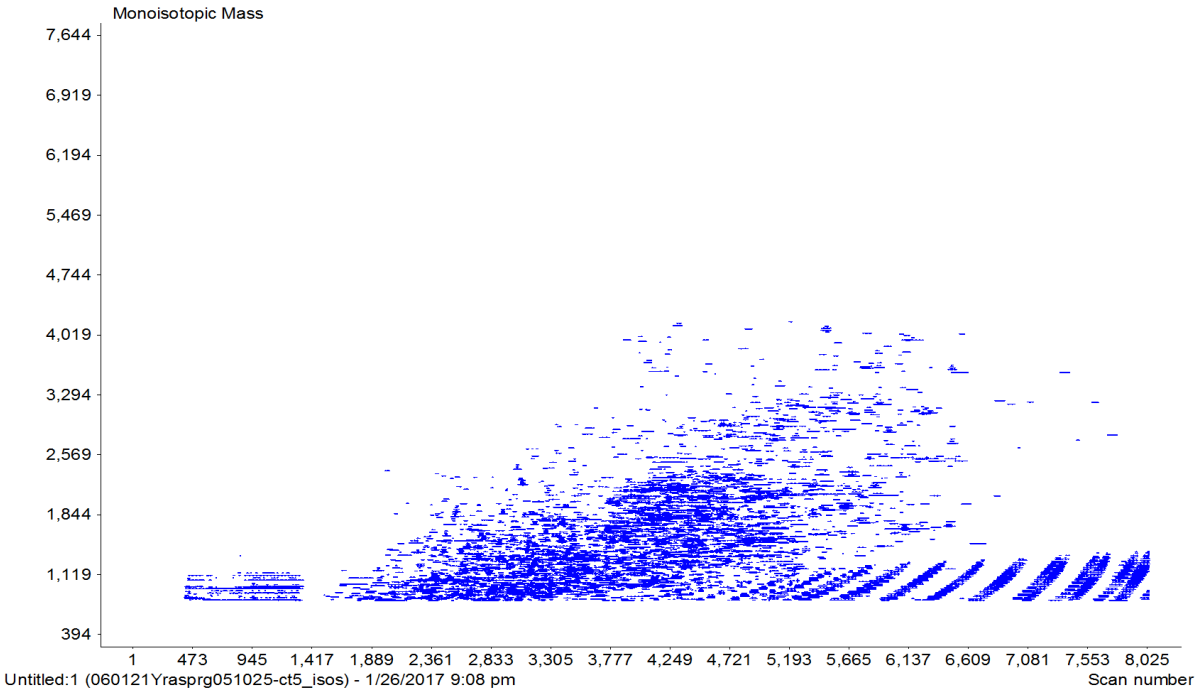

Figure 4.13: ABRFsPRG2006 processed by VIPER with intensity 1E5, charge less than 3. The horizontal axis represents the scan number, and the maximum is 8,025. The vertical axis is the mass, where the range is from 800 to 2,000.

survey scans, another has the peaks for all MS scans. We used those selected peptides from PEAKS to find their corresponding MS survey scans, with the facilitation of MS scans from Decon2LS. As a result, the number of selected MS scans was 4,148. After that, we searched the peaks for MS from Decon2LS, for all selected MS survey scans. Recall that from PEAKS we have selected 5,158 identified peptides, we used their masses to filter the peaks, with the mass tolerance of 2 ppm. After selecting peaks from MS spectra with these peptides, we generated a new file with smaller number of peaks for MS. Finally, we used VIPER to identify LC-MS peaks from this new file of MS peaks across elution time, and we obtained 9,974 de-convoluted peaks as the input. The framework for selecting peaks from the raw dataset is shown in Figure Algorithm 8. Additionally, the details of these steps are introduced in the follows.

Algorithm 8: Filtering peaks: generating a raw simulated authentic dataset.

Input: CPL: a protein candidate list;
SPL: a selected proteins' identified peptides list;
MSS:all the MS survey scans;
S: a survey MS scan;
P: a peak in a survey MS scan;
SMS: all the MS survey scans that contain selected peptides;
PMS: all the peaks in a survey MS scan;
δ: a mass tolerance threshold;
Output: PRW: a partially raw simulated authentic dataset;

```
for i := 1 to |MSS| do
    if S_i ∈ SMS then
        for j := 1 to |PMS| do
            for k := 1 to |SPL| do
                if P_j is a peak in the interval of (P_k − δ, P_k + δ) then
                    put P_j into PRW;
                    break;
                end if
            end for
        end for
    end if
end for
return PRW
```

In the first step, we used PEAKS software [82] to identify the proteins. PEAKS studio performs LC-MS/MS dataset analysis and statistics according to the experimental design. Raw

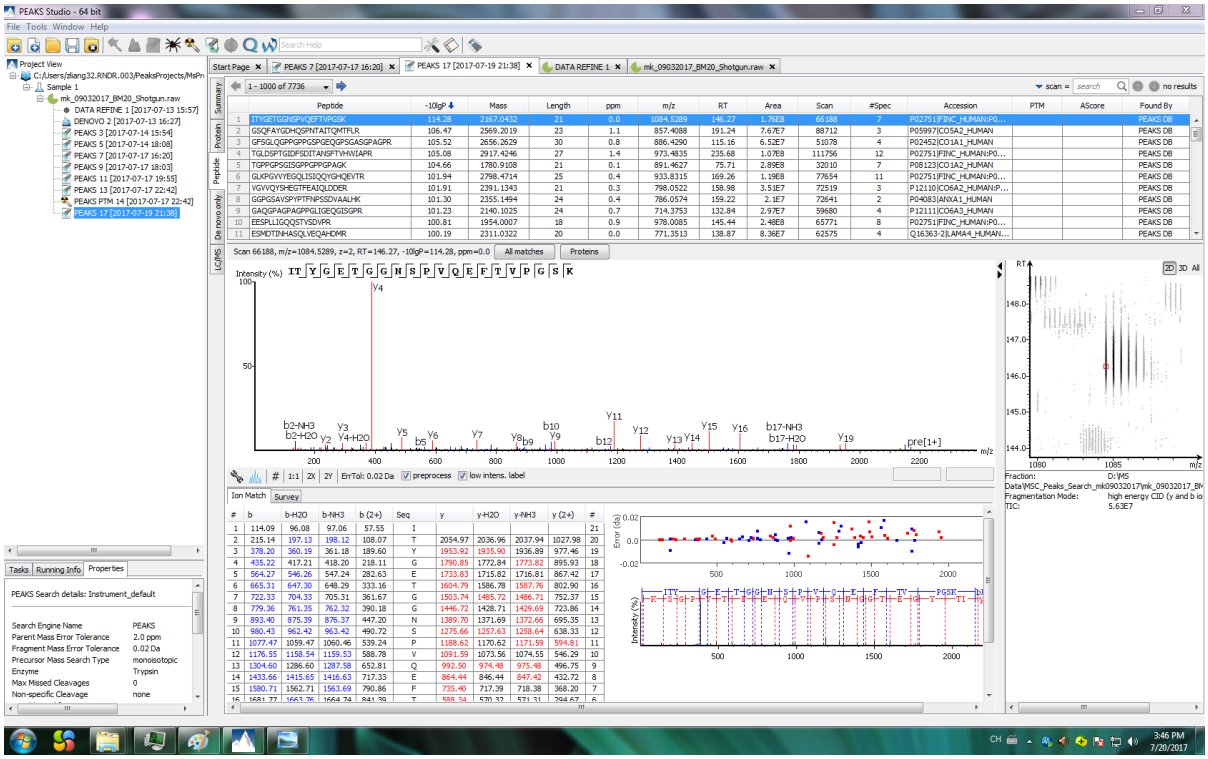

Figure 4.14: The identified peptides for 2,000 proteins by PEAKS DB. The right windows have the information for identified peptides: the top is a score for a peptide, the middle is a MS/MS spectrum for this peptide (scan 66,188), and the bottom is a score for the amino acid residue.

LC-MS/MS dataset often contains noise, and PEAKS will do dataset refinement to improve the overall quality of the data, such as precursor *m/z* correction. The next processing is peptide *de novo* sequencing, which derives the peptide sequence from the tandem mass spectrum without the need of a sequence database. Following the identification of peptides with MS/MS spectra, the resulting peptide sequences are used to determine the original protein components of the samples with high sensitivity and accuracy. Figure 4.14 shows the identified peptides by PEAKS DB, the right windows have the information for these identified peptides: the top has a score for a peptide, the middle has a MS/MS spectrum for a peptide (scan 66,188), and the bottom has a score for the peptide's amino acid residue. Figure 4.15 displays the identified proteins by PEAKS DB, where the right windows have the information for these identified proteins: the top has all scores for the identified proteins, and the bottom has the details for an identified protein and all its matched peptides, which are marked by blue lines.

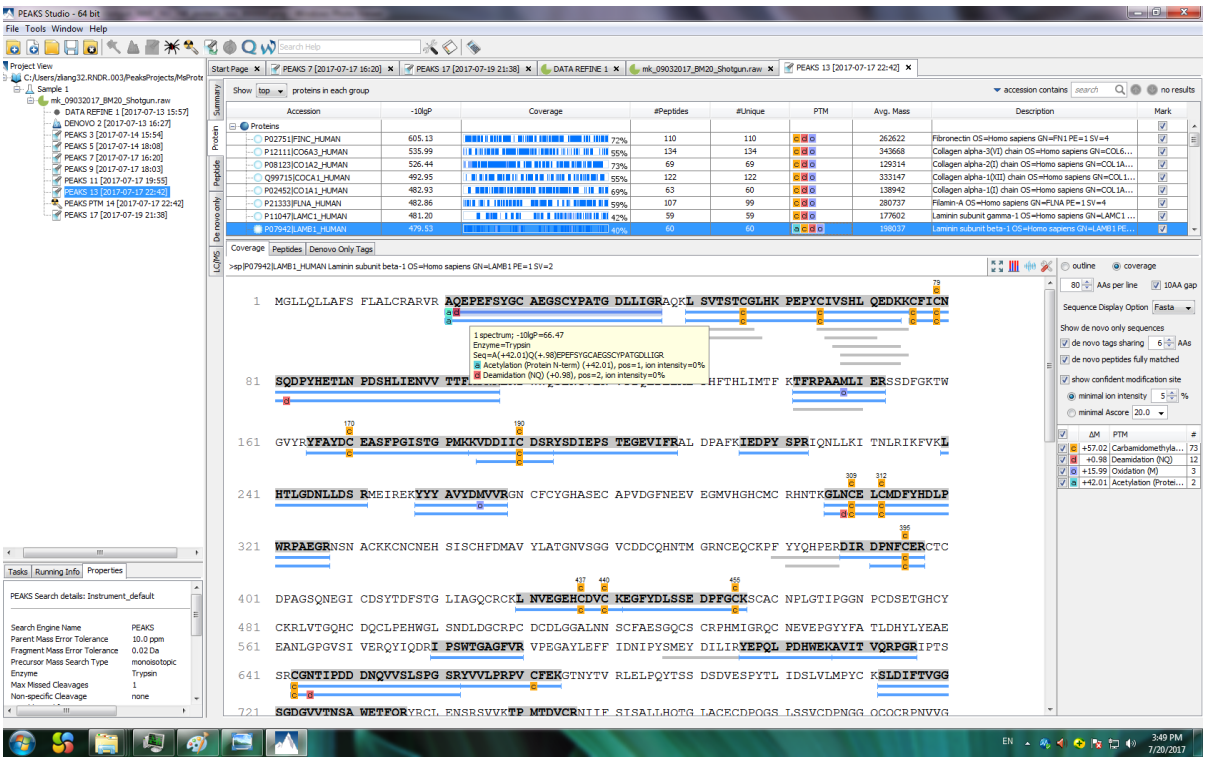

Figure 4.15: The identified proteins for 2,000 proteins by PEAKS DB. The right windows have the information for identified proteins: the top is all identified proteins' information, and the bottom is the details for an identified protein with all its matched peptides, which are marked by blue lines.

The next step is the database searching in PEAKS. The relationship of the original MS and scans from the raw dataset is shown in Figure 4.16, the maximum mass is approximate 9,000, and the maximum scan is 127,129. Among all the scans, there are 23,380 MS scans. After searching by PEAKS, we have the protein candidate list, which contains 1,606 identified proteins altogether. We selected top 519 proteins from them, and obtained corresponding 5,686 identified peptides. After removing those same peptides matched to different proteins, 5,158 ones left. We used these peptides to find their MS survey scans, and the number was 4,148. Again, we searched from the raw dataset for all these selected MS survey scans. We filtered the peaks with these identified peptides' masses, by the mass tolerance of 2 ppm. After selecting the peaks from the raw dataset to generate a new raw data, we used VIPER to identify LC-MS peaks from MS peaks across elution time. The two dimensional relationship of mass and scan number for the selected 519 proteins from the partially raw dataset is shown in Figure 4.17, where the maximum mass is approximate 6,000, and the maximum scan is 127,129. From this figure, we know that distribution of the partially raw dataset is much sparse than the original one in Figure 4.16. The framework for selecting peaks from the raw dataset is shown in Algorithm 8.

4.5.4 Experimental Results

Dataset One

We used 770 simulated peaks from 89 proteins to our algorithms, with a tolerance δ=2ppm. Here the parameter for trypsin digestion is "without missed cleavage" (not allowing internal trypsin cleavage site). Therefore, for the peptides with 1 or 2 missed cleavages, they cannot match the peptides from ground-truth proteins. The results are shown in Table 4.2. From this table, we know that our approach is better than MOWSE.

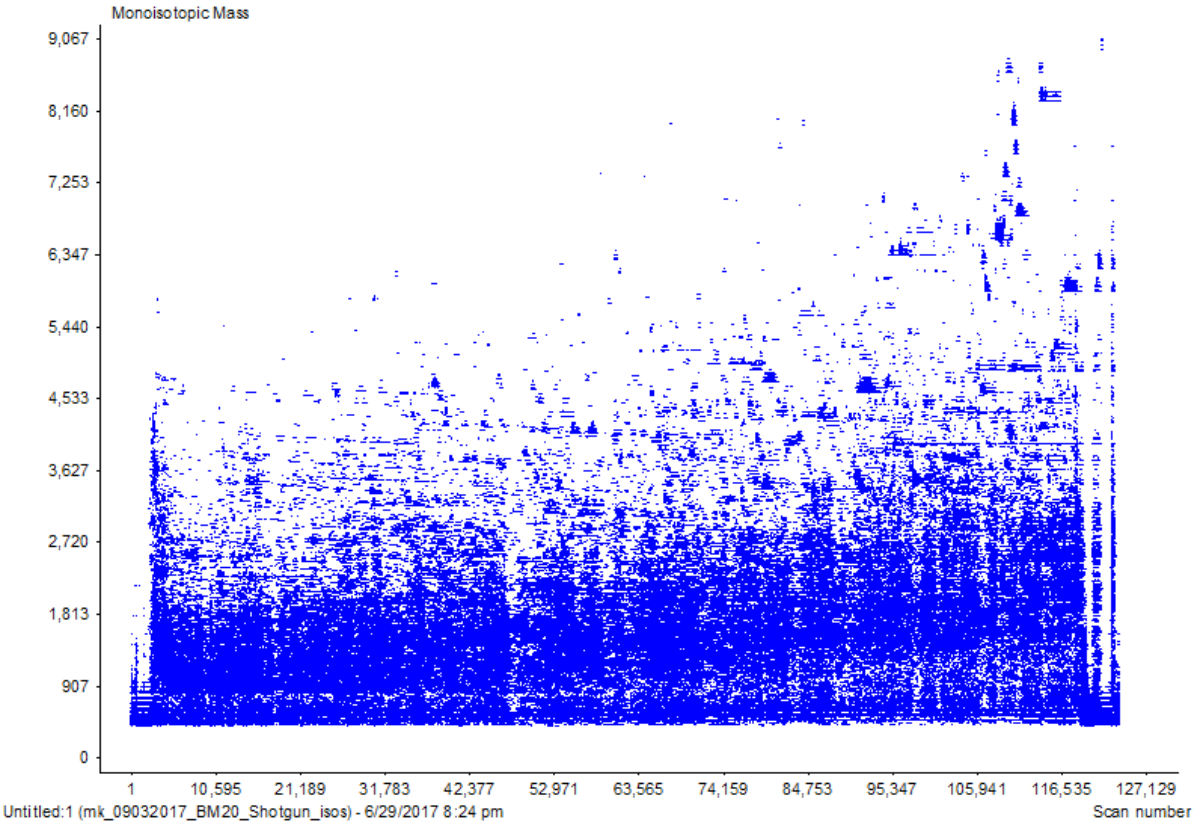

Figure 4.16: The display of the raw dataset for 2,000 proteins processed by VIPER. The horizontal axis represents the retention time, and the maximum scan is 127,129. The vertical axis is the mass, and the maximum mass is approximate 9,000.

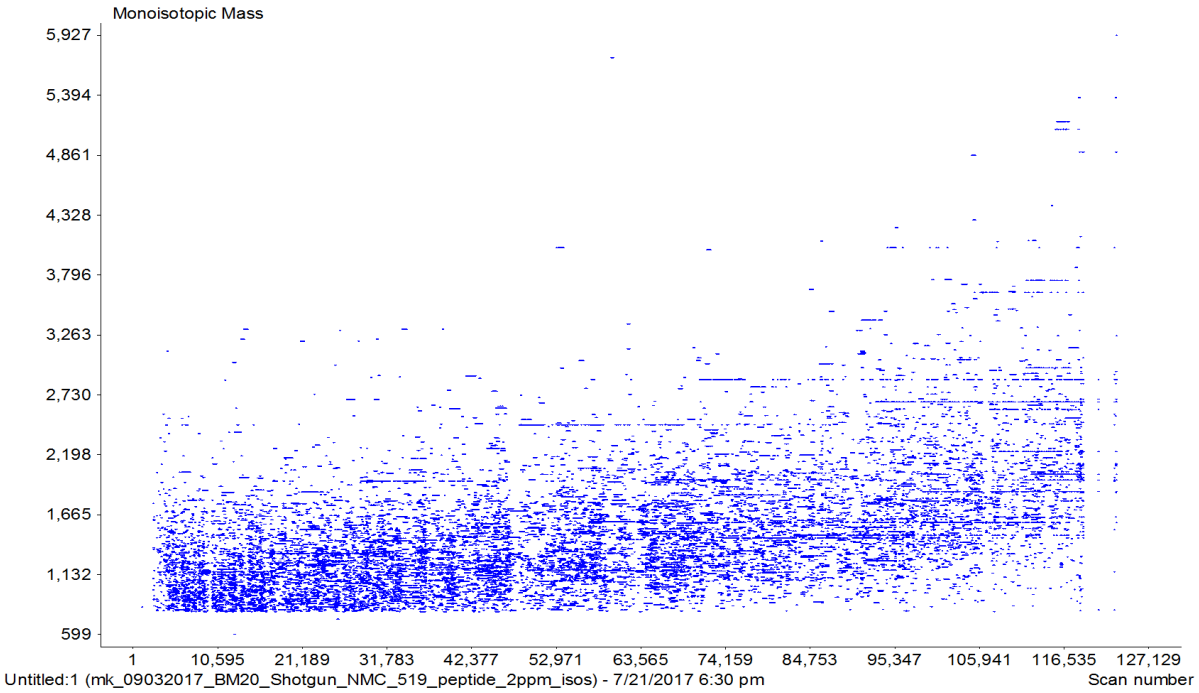

Figure 4.17: The display of the partially raw dataset for top 519 proteins processed by VIPER. The vertical axis is the mass, and the maximum one is approximate 6,000. The horizontal axis represents the retention time, and the maximum scan is 127,129. The distribution is much sparse than the original one.

Table 4.2: The performance of different algorithms on dataset one

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	63%	63%	63%
ProbS	65%	65%	65%
HeatS	67%	67%	67%

Dataset Two

When database searching, we set the mass tolerance threshold as 1 ppm. The results are shown in Table 4.3. For the intensities of 5E4 and 1E, we saw similar results, which shows our method is not sensitive to the intensity threshold. In order to demonstrate the performance of our approach, we compare other state-of-the-art methods with this dataset. Here SPA, Subtraction, Losak, and Losau are from He et al [50], where SPA and Subtraction are traditional PMF algorithms, Losak and Losau are developed by He et al [50]. Here the performance of ProbS is similar to traditional PMF such as SPA, while HeatS performs better than other methods.

Table 4.3: The performance of different algorithms on dataset two

Algorithms	*Precision*	*Recall*	*F1-measure*
SPA	24%	24%	24%
Subtraction	43%	43%	43%
Losak	67%	67%	67%
Losau	61%	71%	66%
MOWSE	33%	33%	33%
ProbS	29%	29%	29%
HeatS	69%	69%	69%

Dataset Three

From the report of PEAKS, we selected the top 519 proteins. Based on them, we filtered the raw dataset with 2 ppm, and a partially raw dataset could be produced. Consequently, we used Decon2LS [56] and VIPER [92] again, obtained 9,974 de-convoluted peaks as the input. When database searching, we set the mass tolerance threshold to 2 ppm. In order to compare different methods fairly, we used the same human database from UniProt. For the

519 proteins identified by PEAKS, we grouped them if they were isoforms. After that, 510 different proteins were identified from PEAKS, and we used them as the ground-truth proteins. For each algorithm, the top 510 proteins were compared with the ground-truth proteins. The results are shown in Table 4.4.

We have demonstrated that our algorithms have significant improvement compared to other methods, and the purpose of dataset three is to find the maximum proteins our algorithms can identify. Therefore, here we only compare with MOWSE. The results show that our algorithms are still better than MOWSE, although they are not perfect for hundreds of proteins. We also checked the results for different threshold number from the candidate proteins list, in order to know the distribution of ground-truth proteins. By using the same dataset, we counted the number of true positive proteins from top 255 proteins, and 127 ones as well, which are shown in Table 4.5 and Table 4.6. Notice that, here *Recall* and *F1-measure* are different to *Precision*, since n_P and $(n_{TP}+n_{FP})$ are different by choosing different thresholds. For example, in Table 4.5, the *Precision* for HeatS is 82%, which means 209 proteins are TPs in the reported 255 proteins. Consequently, the *Recall* is $209/510 = 41\%$, and the *F1-measure* is $0.82*0.41*2/1.23 = 55\%$.

Table 4.4: The performance of different algorithms on dataset three for top 510 proteins

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	48%	48%	48%
ProbS	29%	29%	29%
HeatS	60%	60%	60%

Table 4.5: The performance of different algorithms on dataset three for top 255 proteins

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	59%	30%	40%
ProbS	49%	25%	33%
HeatS	82%	41%	55%

Table 4.6: The performance of different algorithms on dataset three for top 127 proteins

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	72%	18%	29%
ProbS	55%	14%	23%
HeatS	89%	22%	35%

Chapter 5

Adjustment of Scores

After the above steps (network based inference methods), we could obtain a set of proteins Z from the scoring function of ProbS or HeatS. For the purpose of getting better results, we could also consider other factors, such as the mass of a protein, and the RT of a peptide.

The raw score from NBI has a bias with larger proteins compared to smaller proteins. Because heavier proteins have more peptides than smaller ones, their matches and scores will be higher than the latter. Therefore, we need to adjust the raw score. Here we choose linear regression, trying to tackle this bias.

So far, in ProbS and HeatS, we only analyze the information for the mass of a peptide, however there is also retention time (RT) associated with this mass. In addition, we use the order of RTs from the matched peptides for further adjustment of the protein score. Here, a dynamic programming will be applied.

In this chapter, we will show how to adjust the raw scores, with linear regression and ordered retention time.

5.1 Linear Regression (LR)

When database searching, a larger protein has a larger set of peptides than a small protein resulting, often resulting in a number of matched peptides. Taking the simulated 89 proteins as

example, Figure 5.1 shows the average matched number in 400 intervals. In the database, with the increase of the protein intervals, the average matched number of peptides for the protein interval is also increasing. For example, if a protein p_1 has a matched peptide, and the average matches for the interval of p_1 is one; while another protein p_2 has two matched peptides, but the average matches for the interval of p_2 is two. Under this situation, the score of p_2 should not be double than that of p_1, they could have the same score. Accordingly, we need to adjust our scoring function to correct this bias towards large proteins. In traditional PMF methods, the molecular weight of each protein is restricted, such as in the range of 30,000 - 100,000 Da. Additionally, they used a simple method, a constant (50,000) divided by the protein mass, to adjust the raw score. However, this phenomenon increases dramatically with the increase of the size of input proteins. Therefore, we need a better method to deal with the bias with larger proteins.

Here we can use a simple linear regression technology to deal with this problem. For this purpose, we divide protein mass by 10,000 Da, resulting in the 400 protein intervals, which is the same with MOWSE. For the proteins in each interval, we can calculate the number of average matched peptides. Since there are not enough proteins in the protein interval with heavier proteins in the Human database, we use the first 25 intervals to estimate a linear regression line for all 400. Finally, we could get the estimated matching number from an equation of $c = a + b * |p_j|$, where b is the slope of the line, a is the intercept, a and b are estimated from first 25 intervals, $|p_j|$ is the explanatory variable, and c is the dependent variable. Here, $|p_j|$ is the mass of protein p_j, and we use c to normalize the score for p_j, in order to compare the proteins with different masses fairly. Therefore, the formula is as follows:

$$F_L(P, p_j, \delta) = \frac{\Sigma_{q=1}^{t} f(u_q)}{c(|p_j|)},$$

where $f(u_q)$ is the raw score for a matched peptide u_q, t is the total number of matches for p_j, and $c(|p_j|)$ is the estimated matching number for p_j. By using linear regression for each

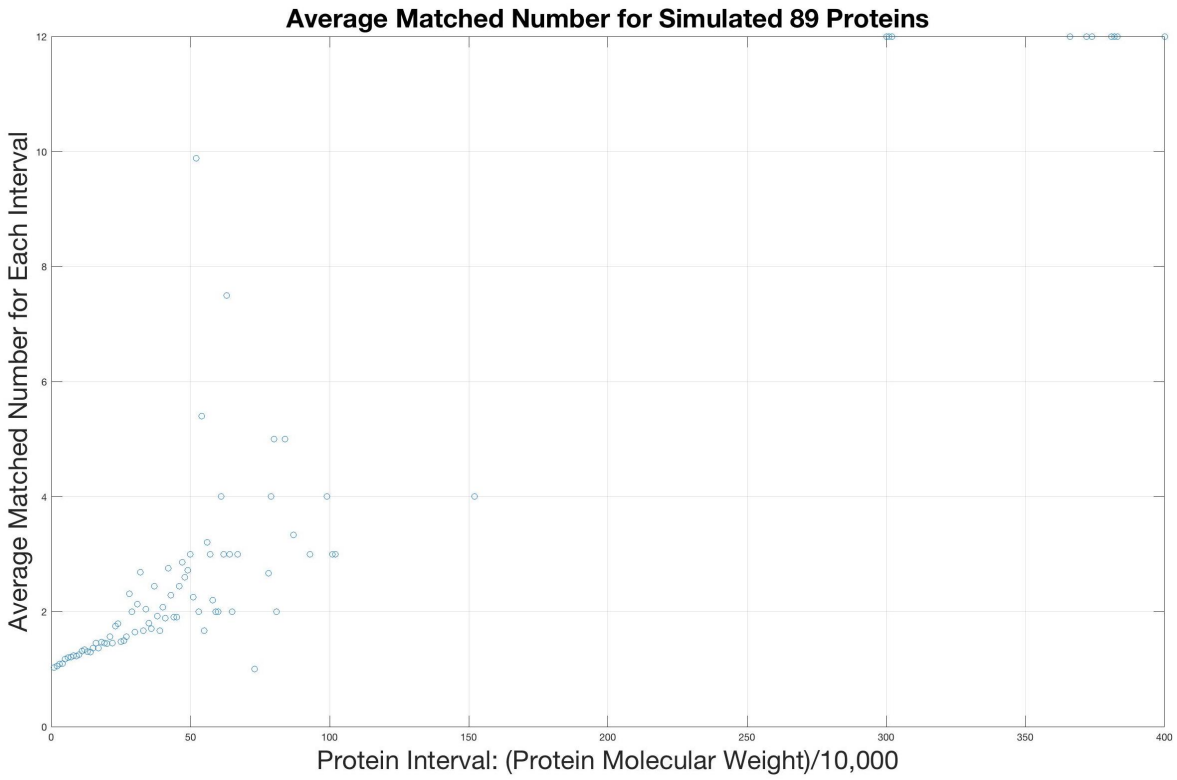

Figure 5.1: The average matched number of peptides in 400 protein intervals for the simulated 89 proteins.

interval, we adjust the raw scores with the estimated average matched number, hoping to keep the balance between different protein masses.

In ProbS, we multiply the scores of matched peptides for a protein. Now we use the root of $c(|p_j|)$ to adjust the raw score. Therefore, we have

$$F_{PL}(P, p_j, \delta) = \sqrt[c(|p_j|)]{\prod_{1 \le q \le t} (\frac{1}{f''_p(u_q)})}$$

to calculate the geometric mean, and get

$$F'_{PL}(P, p_j, \delta) = \frac{\sum_{q=1}^{t} \lg(\frac{1}{f''_p(u_q)})}{c(|p_j|)}$$

by using log. While in HeatS, we add the scores of matched peptides for a protein. Now we adjust raw score by dividing by the average matched number $c(|p_j|)$. Consequently, we have

$$F_{HL}(P, p_j, \delta) = \frac{\sum_{q=1}^{t} f'_h(u_q)}{c(|p_j|)},$$

which is an arithmetic mean. The regression line based on average matched number for each interval is shown in Figure 5.2. From the above formulas, if two proteins have the same number of matches, the protein in the small interval has more chance to get a high score, compared to the protein in the large interval. Consequently, the score for the huge protein in the interval of 400, will be decreased dramatically to avoid false positives. The linear regression framework for PMF is shown in Algorithm 9.

5.2 Ordered Retention Time (ORT)

Together with highly accurate peptide mass measurements, the additional RT information is also provided by preliminary HPLC analysis [5, 11, 12, 45]. RT was not used for protein identification. However, in 2011 Bochet et al [17] considered the order of peptide RT and applied

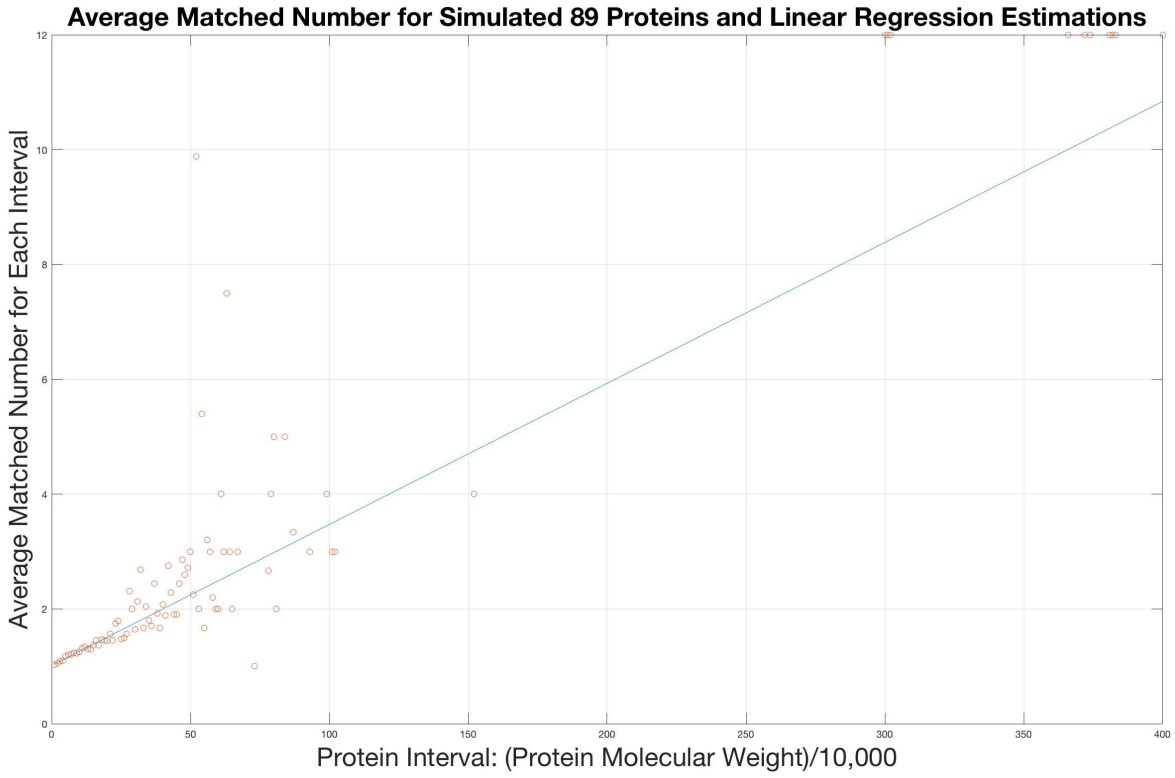

Figure 5.2: The average matched number in 400 intervals and their linear regression for the simulated 89 proteins.

Algorithm 9: LR for PMF: finding a set of proteins to best explain experimental peaks.

Input: D: a database of s proteins;
P: an observed peak list;
δ: a mass tolerance threshold;
$U(p_j)$: a set of matched peptides for protein p_j;
$f(u_q)$: a matching score for a peptide $u_q \in p_j$;
$c(|p_j|)$: the estimated matched number for the protein p_j;
Δ: a threshold score ;
Output: Z: a set of proteins $Z \in D$ that best explaining P;

```
for j := 1 to |D| do
    F_L(P, p_j, δ) ← 0
    for q := 1 to |U(p_j)| do
        F_L(P, p_j, δ) ← F_L(P, p_j, δ) + f(u_q)
    end for
    F_L(P, p_j, δ) ← F_L(P, p_j, δ)/c(|p_j|)
end for
Z ← {p_j | p_j ∈ D, F_L(P, p_j, δ) ≥ Δ}
return Z
```

dynamic programming with quantile regression to identify proteins. They assumed that for the matched peptides in a protein, the order of peptides' predicted RTs should be consistent with the order of peptides' experimental elution times. Therefore, among these matched peptides in one protein, they would like to find the maximum number of peptides, whose order of predicted RTs should be the same with that of elution times. Additionally, the success of this method relies on accurately predicted peptide RT. In 2004, Krokhin, et al. developed s SSRCalc tool [62], used a linear model with different hydrophobicity coefficients for amino acids close to termini of the peptide. For each peptide, SSRCalc provides a hydrophobicity index (HI) which can be converted into a predicted RT by an affine transformation [61]: $RT = a + b * HI$.

In practice, it is hard to compute the accurate experimental RT from the predicted algorithm, and sometimes there are several different RTs for one peptide from a raw dataset. Therefore, we only keep the order of RTs, which is more robust than using RTs directly. For a protein p_j, the matched peptide set $U(p_j)$ is sorted by predicted RT, and the matched peak set $L(p_j)$ is also sorted according to experimental RTs. By dynamic programming [117], we would like to find a subset in $U(p_j)$, which is consistent with the order of predicted RTs with the maximum score. Here the purpose is to remove potentially random matches. The dynamic programming with HeatS and ORT for protein p_j is shown in Algorithm 10.

5.3 Experiments

Again, we have an input set of experimental peaks $P = \{l_i(m_i, r_i) \mid i = 1, 2, \ldots, n\}$. We also have a database of proteins $D = \{p_j \mid j = 1, 2, \ldots, s\}$. For PMF, our objective is to find a set of proteins $Z \in D$ that contains all the proteins whose scores are greater than a threshold Δ, i.e. $Z = \{p_j \mid p_j \in D, F(P, p_j, \delta) \geq \Delta\}$.

Here we still use three datasets introduced in the former chapter. The testing results demonstrate that our adjustment algorithms (LR and ORT) could improve the performance significantly.

Algorithm 10: ORT: dynamic programming with HeatS and order of RT for protein p_j.

Input: P: an observed peak list with RT;
δ: a mass tolerance threshold;
$U(p_j)$: a set of matched peptides for protein p_j;
$L(p_j)$: a set of matched peaks for protein p_j;
$f'_h(u_q)$: a HeatS score for a peptide $u_q \in p_j$;
$c(|p_j|)$: the estimated matched number for the protein p_j;
Output: the maximum score for protein p_j;

Sort $U(p_j)$ with HI;
Sort $L(p_j)$ with RT;
for $q := 0$ **to** $|U(p_j)|$ **do**
  $n[q, 0] \leftarrow 0$
end for
for $g := 1$ **to** $|L(p_j)|$ **do**
  $n[0, g] \leftarrow 0$
end for
for $q := 1$ **to** $|U(p_j)|$ **do**
  for $g := 1$ **to** $|L(p_j)|$ **do**
    $n[q, g] \leftarrow \max(n[q-1, g-1] + f'_h(u_q), n[q-1, g], n[q, g-1])$
  end for
end for
return $(n[|U(p_j)|][|L(p_j)|])/c(|p_j|)$

5.3.1 Dataset One

Again, we inputted 770 simulated peaks from 89 proteins to our algorithms. The tolerance δ is 2ppm, and the parameter for trypsin digestion is "without missed cleavage". The results are shown in Table 5.1. Here ORT is based on the algorithm of Bochet et al [17]. To ensure a fair comparison, we added LR to MOWSE and ORT, and also improved their performance. From this table, we know that our approaches are better than MOWSE and ORT. Among four different methods, HeatS has the best performance. Generally, the adjustment of a raw score could get better result.

Table 5.1: The performance of different algorithms on dataset one

Algorithms	*Precision*	*Recall*	*F1-measure*
ORT	60%	60%	60%
MOWSE	63%	63%	63%
ProbS	65%	65%	65%
HeatS	67%	67%	67%
MOWSE&LR	69%	69%	69%
ProbS&LR	69%	69%	69%
ProbS&HeatS&LR	71%	71%	71%
HeatS&ORT&LR	74%	74%	74%
ORT&LR	75%	75%	75%

5.3.2 Dataset Two

The authentic MS dataset is still from a mixture of 49 standard human proteins in the ABRFsPRG2006 study. When database searching, we set mass tolerance threshold to 1 ppm. The results are shown in Table 5.2. There are only 39 proteins whose matches are greater than two. We identified all of them. However, the other 10 proteins, we cannot report because they only have zero or one match. We also tested other input dataset with different intensities, such as 5E4 and 1E5, then the number of peaks were 6812 and 3996 respectively. We saw similar results, which shows our method is not sensitive to the intensity threshold. In order to demonstrate the performance of our approach, we compare other state-of-the-art methods with

this data. Here SPA, Subtraction, Losak, and Losau are from He et al [50], where SPA and Subtraction are traditional PMF algorithms, Losak and Losau are developed by He et al. We also test MOWSE and ORT from Bochet et al [17]. To ensure a fair comparison, we add LR for both of them.

The results demonstrate that our algorithms achieve significant improvement compared to other methods. In the top 49 proteins for $HeatS\&ORT$, there are only 2 proteins belong to FP. The others are 39 expected proteins and 8 bonus expected ones. Here expected proteins are intended 49 proteins, and bonus expected proteins are 20 additional human proteins, which were reported by abrf2007. That means all 47 proteins are TP, which proves the correctness and robustness of our approaches.

Table 5.2: The performance of different algorithms on dataset two

Algorithms	*Precision*	*Recall*	*F1-measure*
SPA	24%	24%	24%
Subtraction	43%	43%	43%
Losak	67%	67%	67%
Losau	61%	71%	66%
MOWSE	33%	33%	33%
MOWSE&LR	63%	63%	63%
ORT	37%	37%	37%
ORT&LR	65%	65%	65%
ProbS	29%	29%	29%
ProbS&LR	63%	63%	63%
HeatS	69%	69%	69%
HeatS&LR	73%	73%	73%
HeatS&ORT&LR	80%	80%	80%

5.3.3 Dataset Three

The simulated authentic dataset has 9,974 de-convoluted peaks. When database searching, we set mass tolerance threshold to 2 ppm. In order to compare different methods fairly, we used the same Human database from UniProt. Finally, we obtained 510 different proteins from PEAKS, and used them as the ground-truth proteins. The results are shown in Table 5.3. We

also checked the different threshold numbers from the protein candidate list, such as top 255 proteins, and 127 ones as well, which are shown in Table 5.4 and Table 5.5. From these results, we know that LR and ORT can improve the performance of different algorithms. Notice that, with the increase of input dataset, for the smaller proteins (in the interval 1, 2, etc), we should keep their original scores to avoid underestimating with linear regression.

Table 5.3: The performance of different algorithms on dataset three for top 510 proteins

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	48%	48%	48%
ProbS	29%	29%	29%
ProbS&LR	49%	49%	49%
HeatS	60%	60%	60%
HeatS&LR	65%	65%	65%
HeatS&ORT&LR	66%	66%	66%

Table 5.4: The performance of different algorithms on dataset three for top 255 proteins

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	59%	30%	40%
ProbS	49%	25%	33%
ProbS&LR	73%	37%	49%
HeatS	82%	41%	55%
HeatS&LR	85%	43%	57%
HeatS&ORT&LR	85%	43%	57%

Table 5.5: The performance of different algorithms on dataset three for top 127 proteins

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	72%	18%	29%
ProbS	55%	14%	23%
ProbS&LR	91%	23%	37%
HeatS	89%	22%	35%
HeatS&LR	91%	23%	37%
HeatS&ORT&LR	91%	23%	37%

Chapter 6

Post Processing Algorithms

After the above steps, we obtained a protein candidate list with a scoring function. Given a threshold Δ, we could find a set of proteins $Z = \{p_j \mid p_j \in D, F(P, p_j, \delta) \geq \Delta\}$, which can best explain the peaks. But there may be still room to improve the accuracy of these score functions, by the heuristic ways.

In this chapter, we will design two post processing algorithms to correct the scores: the assignment of peaks, restricting that a peak can only be assigned to one peptide in order to reduce random matches; and the protein filtration, assuming each peak can only be assigned to one protein.

6.1 Assignment of Peaks (AP)

Since we do not use MS/MS, we cannot know the sequence for each matched peptide, and it is hard to distinguish peptides within a tolerance threshold. After searching the database, we have some candidate proteins with descending order of scores. The top ones tend to be ground-truth proteins. Therefore, we start from the top protein to the bottom one. For each matched peak in a protein, if this peak is never assigned to a theoretical peptide, we mark it; otherwise, just ignore it. After searching, each matched peak will be assigned to only one peptide. The reason is that, a peak is actually from one peptide in most wet-lab experiments.

Now we reevaluate the proteins with these updated peaks, hoping to reduce random matches. The detail of the assignment of peaks is shown in Algorithm 11. Here we assume that a peak can only match to one peptide. After the assignment of peaks, we select the best matching from a peak among several peptides, and reduce the other matches which are false positives.

Algorithm 11: Assignment of peaks: assuming a peaks can only match to one peptide.

Input: P: an observed peak list;
PSM: all of the peptide spectrum matches;
$U(p_j)$: a set of matched peptides for protein p_j;
$L(p_j)$: a set of matched peaks for protein p_j;
CPL: a protein candidate list in a descending order;
Output: PSO: a set of peptide spectrum matches;

```
for j := 1 to |CPL| do
    for q := 1 to |U(p_j)| do
        (u_q, l_g) is a matching;
        if l_g never matches to any peptide then
            put (u_q, l_g) into PSO;
            mark l_g;
        end if
    end for
end for
return PSO
```

6.2 Protein Filtration (PF)

In the last step, we wish to filter false positive proteins. We hope that for each reported protein, it at least has one novel peak, which means this peak is never assigned to others. In other words, we restrict that each matched peak will be assigned to only one protein. We also start from the top protein to the last. For each matched peptide in a protein, if it is never used by others, it is a "novel" peptide and we mark it. For each protein, if all of its peptides are not novel, we will remove it from the candidate list. Finally, we may get a smaller Z with the same Δ. The detail of the protein filtration is shown in Algorithm 12, assuming a peak can only match to one protein.

Algorithm 12: Protein filtration: assuming a peaks can only match to one protein.

Input: P: an observed peak list;
PSO: a set of petide spectrum matches where one spectrum matches to one peptide;
$U(p_j)$: a set of matched peptides for protein p_j;
$L(p_j)$: a set of matched peaks for protein p_j;
CPL: a protein candidate list in a descending order;
Output: CPF: a set of protein candidate list after protein flitration;

```
for j := 1 to |CPL| do
    Novel ← false;
    for q := 1 to |U(p_j)| do
        (u_q, l_g) is a matching;
        if l_g never matches to any protein then
            Novel ← true;
            mark l_g;
        end if
    end for
    if Novel==ture then
        put (p_j) into CPF;
    end if
end for
return CPF
```

6.3 Experiments

Again, we have an input set of experimental peaks P. We also have a database of proteins D. For PMF, our objective is to find a set of proteins $Z \in D$ that contains all the proteins whose scores are greater than a threshold Δ.

Here we still use the same three datasets. The testing results demonstrate that our post processing algorithms (AP and PF) could improve the performance.

6.3.1 Dataset One

The simulated dataset is still from a wet-lab experiment with 248 proteins from Human, which identified by MS/MS with Mascot. We selected top 89 proteins and their 770 matched peptides as the input dataset, with a tolerance δ=2ppm. The results are shown in Table 5.1. Additionally, the post processing algorithms could also improve the performance. We can

identify all the proteins whose matches are greater than two, and the top 61 ones are all TP. As a result, AP and PF could improve 8% performance compared to the best one of other methods.

Table 6.1: The performance of different algorithms on dataset one

Algorithms	*Precision*	*Recall*	*F1-measure*
ORT	60%	60%	60%
MOWSE	63%	63%	63%
ProbS	65%	65%	65%
HeatS	67%	67%	67%
MOWSE&LR	69%	69%	69%
ProbS&LR	69%	69%	69%
ProbS&HeatS&LR	71%	71%	71%
HeatS&ORT&LR	74%	74%	74%
ORT&LR	75%	75%	75%
HeatS&ORT&LR&AP	80%	80%	80%
HeatS&ORT&LR&AP&PF	83%	83%	83%

6.3.2 Dataset Two

The authentic MS dataset is from a mixture of 49 standard human proteins in the ABRF-sPRG2006 study. In the last chapter, for $HeatS\&ORT$, we have identified 47 ground-truth proteins, including 39 expected proteins and 8 bonus expected ones. Since there are only 2 proteins belong to FP in the top 49, it is hard to improve the performance. Therefore, we do not use post processing algorithms here.

6.3.3 Dataset Three

The simulated authentic dataset still contains 9,974 de-convoluted peaks. When database searching, we set mass tolerance threshold as 2 ppm. The results are shown in Table 6.2. We also checked the different threshold numbers from top 255 proteins in the candidate list, and 127 ones as well, which are shown in Table 6.3 and Table 6.4. From the results, AP could improve 5% performance compared to the best of other methods for 510 proteins.

Table 6.2: The performance of different algorithms on dataset three for top 510 proteins

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	48%	48%	48%
ProbS	29%	29%	29%
ProbS&LR	49%	49%	49%
HeatS	60%	60%	60%
HeatS&LR	65%	65%	65%
HeatS&ORT&LR	66%	66%	66%
HeatS&ORT&LR&AP	71%	71%	71%

Table 6.3: The performance of different algorithms on dataset three for top 255 proteins

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	59%	30%	40%
ProbS	49%	25%	33%
ProbS&LR	73%	37%	49%
HeatS	82%	41%	55%
HeatS&LR	85%	43%	57%
HeatS&ORT&LR	85%	43%	57%
HeatS&ORT&LR&AP	87%	44%	59%

Table 6.4: The performance of different algorithms on dataset three for top 127 proteins

Algorithms	*Precision*	*Recall*	*F1-measure*
MOWSE	72%	18%	29%
ProbS	55%	14%	23%
ProbS&LR	91%	23%	37%
HeatS	89%	22%	35%
HeatS&LR	91%	23%	37%
HeatS&ORT&LR	91%	23%	37%
HeatS&ORT&LR&AP	92%	23%	37%

Chapter 7

Target-Decoy Search Strategy

After database searching, we can obtain a protein candidate list with a scoring function. Given a threshold Δ, we could find a set of proteins $Z = \{p_j \mid p_j \in D, F(P, p_j, \delta) \geq \Delta\}$, which can best explain the peaks. Now, the problem is how to choose a proper score threshold Δ, for reporting the identified proteins with satisfactory accuracy. This step is also called result validation.

Precision and recall are two competing goals in MS data analysis for protein identification, which should be considered together. Nowadays, the accuracy of the result is often measured by the false discovery rate (FDR), which is defined as the ratio between the number of reported false hits to the total number of reported hits. Moreover, the most widely used approach for FDR estimation is the target-decoy strategy.

In this chapter, we will review target-decoy strategy for tandem mass spectrometry, and introduce the concept of decoy database. Furthermore, we design two new target-decoy strategies for PMF, including random dataset and decoy dataset. The experiments on three datasets demonstrate that our target-decoy methods achieved intriguing performance.

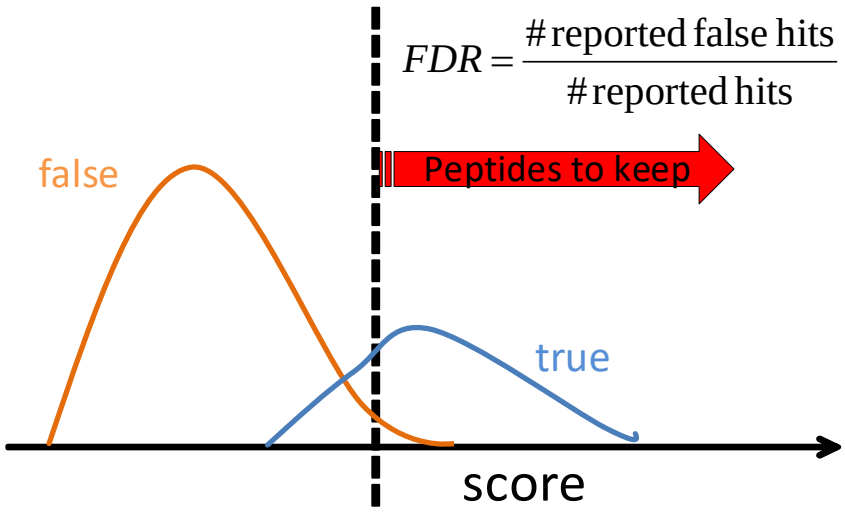

Figure 7.1: A scoring function is used to separate the true and false identifications. False discovery rate (FDR) is the portion of false positives above the user-specified score threshold.

7.1 Target-Decoy for Tandem Mass Spectrometry

In MS/MS data analysis for peptide identification, accuracy and sensitivity are two competing goals that must be talked about together. How to estimate incorrect peptide and protein identifications accurately and precisely is crucial in proteome analyses by MS/MS. Consequently, a scoring function is the core to evaluate the matching quality between a peptide and a MS/MS spectrum. Usually, the software will find a peptide that maximizes the peptide-spectrum matching score. The match between the spectrum and the highest-scoring peptide is called as a peptide-spectrum match (PSM) [35, 36].

A PSM can be incorrect, the reasons include: (1) low quality spectrum; (2) peptides not in the database; and (3) imperfect scoring function. Therefore, invariably have different standards for what constitutes a good match. Recently, the accuracy of the result is often measured by FDR, which is defined as the ratio between the number of reported false hits to the total number of reported hits. In Figure 7.1, the PSMs are sorted by their scores, and the FDR can be estimated after choosing a threshold.

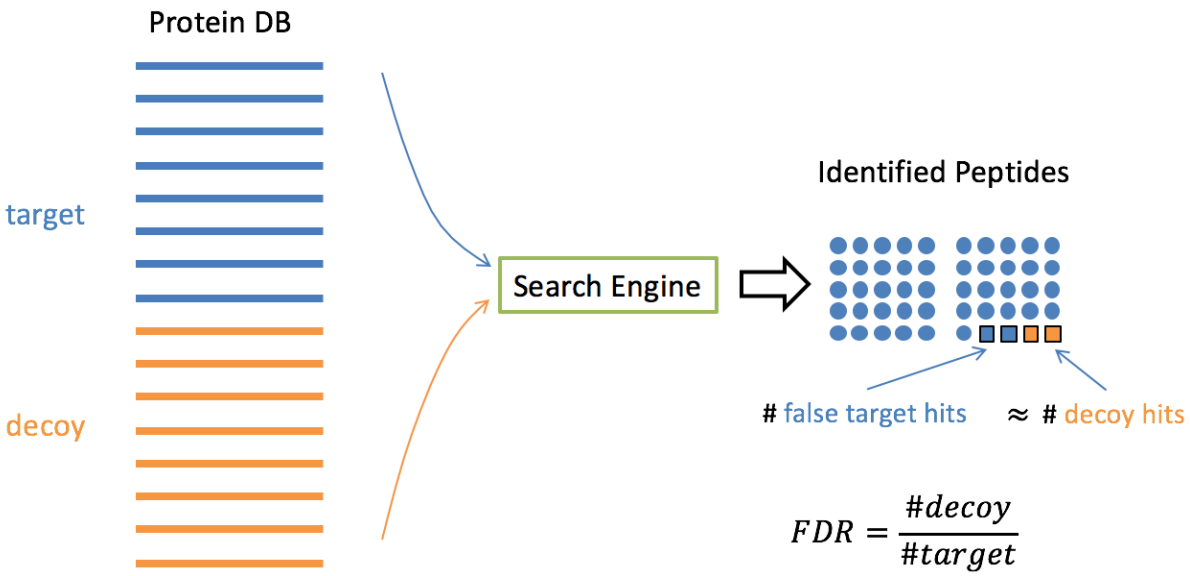

Figure 7.2: FDR estimation [146]. The decoy proteins are randomly generated so that any decoy hit is supposedly a false hit.

In practice, rather than deciding exactly which PSMs are true or false, the *target-decoy* search strategy has emerged as a simple, effective, and most popular way to validate PSMs identified with LCMS/MS since 2007. Nowadays, the target-decoy strategy is the most widely used method for FDR estimation in proteomics [134]. The idea of target-decoy strategy is to extract the false positive information by simulation, with a decoy database, i.e., a database of sequences known to be incorrect. Figure 7.2 illustrates the FDR estimation by the target-decoy strategy, where the blue colors indicate the target hits and the orange colors indicate the decoy hits, the circles are the true hits, and squares are false hits. Furthermore, in the design of sensitive filtering criteria, researchers can be guided by decoy hits to precisely distinguish correct from incorrect PSMs [55].

Target-decoy searching is usually performed in the following steps [36]:

- Constructing a concatenated target-decoy sequence list, marking decoy sequences with a text flag in their annotation.

- Using a MS/MS search engine to interpret input MS/MS spectra, using target-decoy

sequence list.

- Evaluating the relative proportion of target and decoy sequences in the search space, by deriving the multiplicative factor required to estimate false positives, if necessary.
- Estimating false positive-related statistics.
- Using decoy hits to guide the establishment of filtering criteria.
- Reporting statistics for filtered dataset.

7.2 Decoy Database

There are several methods for creating decoy databases. Each has varying advantages and disadvantages, and it must be stressed that no single decoy type is perfect. Here we introduce four of them. Each method produces a decoy database of the same size (i.e., number of amino acids), and also the same number of proteins as the original.

- *Reversed Proteins* — a simple reverse of the amino acid sequence of each protein. This is by far the simplest and most widely used method for creating decoy sequences.
- *Shuffled Proteins* — the order of amino acids in each protein sequence is randomly shuffled, using a uniform distribution random number generator. Protein shuffling is another method used for creating decoy sequences, where the amino acids of each input target protein are randomly rearranged to yield a new decoy protein.
- *Random Proteins* — each amino acid is generated randomly according to their occurrence frequencies in the target database, using a uniform distribution random number generator that can accommodate the generation of billions of random numbers. This is the method internally implemented by some search engines, such as Mascot, for performing target-decoy analyses.

- *Decoy Peptides* — rather than generating entire decoy proteins, one can instead generate decoy peptides directly by altering each peptide sequence derived from the target sequence list. Alterations can take the form of reversals or shuffling. This procedure has the advantage of creating decoy peptides exactly matching the masses of all target peptides considered by the search engine.

In the target-decoy search strategy, assuming no correct peptides are found in both target and decoy portions, then incorrect assignments from target or decoy sequences are equally likely. This means the number of FPs in target and decoy database are identical. Additionally, by doubling the number of selected decoy hits, one can estimate the total number of FPs that meet specific selection criteria [126]. This represents the number of obvious incorrect decoy hits, combined with the hidden incorrect target hits [35]. With FP estimations, it is possible to derive measurements that help evaluate and compare scoring methods and datasets, which is shown in Table 7.1. For example, false discovery rate can be represented as $FP/(TP + FP)$, where FP is "2 * passing decoy assignments", FP is "total passing assignments - number of FP". Here, for the assignments whose scores are above than the score threshold, we call them "passing assignments".

7.3 Random Dataset

Conventionally, a decoy database search is only used for validating searches of a MS/MS dataset. It is hard to get a FDR for PMF, but it can be informative to see the result of repeating a PMF search against a decoy database, especially if the match from target database is close to the significance threshold, or if there is reason to think the experimental values or search parameters may be producing a false positive (FP) [100, 103]. Our experiments on the decoy database also proved that for traditional PMF, it is impossible to use the strategy of FDR. But we also would like to estimate the false positive of our algorithms, instead of a decoy database, we use a random dataset to approximately evaluate it.

Table 7.1: The measurements derived from decoy database search results [35]

Measurement	*Formulation*	*Description of estimate*
False positive (FP)	2 * passing decoy A	# incorrect A above ST
True positive (TP)	Total passing A - # FPs	# correct A above ST
Total correct (TC)	Max TPs for all score criteria	# total correct A in the dataset
Total incorrect (TI)	Total A - TC	# total incorrect A in the dataset
False negative (FN)	TC - TP	# correct A falling below ST
True negative (TN)	TI - FN	# incorrect A falling below ST
Precision	TP / (TP + FP)	% correct A above ST
FP rate or FDR	FP / (TP + FP) or 1 - precision	% incorrect A above ST
Sensitivity	TP / TC	% all correct A above ST
Specificity	TN / (TN + FP)	% all incorrect A below ST
Accuracy	(TP + TN) / total A	% all A correctly classified by ST
Notes:	Abbreviations	# = number of, % = proportion of, A = assignments, ST = score threshold, FDR = false discovery rate.

For each testing dataset, we get the total number of input spectra. For each peak, we add additional noise. This noise can be produced randomly or generated by normal distribution, then the mass and retention time will be changed a little from the original ones. After that, we put the new random dataset to our algorithms, using the same searching parameters. Finally, a protein candidate list will be calculated for random dataset, and we can compare the two lists to estimate FP. Algorithm 13 show the detail of generating a random dataset from a target one.

Now we have two protein candidate lists, target and random results. We can roughly estimate the false discovery rate (FDR) with these two lists. For the target proteins list, we start from the top to the bottom, and get the score and index for each protein; also, at the same time, we start from top to bottom in the random one, and get the protein's score and index. We compare these two scores, and if the target is greater than random, we let the index of target go; otherwise we add the index of the random. By comparing these two indexes, we can evaluate the value of FDR. The detail of the calculation of FDR is shown in Algorithm 14, where we can roughly estimate FDR from these two lists. Actually, this strategy has limitation. Only for the

Algorithm 13: Random dataset: generating a random dataset from the target dataset.

Input: TD: a target dataset;
mt: the mass of a peak in TD;
rt: the retention time of a peak in TD;
MT: a mass tolerance threshold;
RT: a retention time tolerance threshold;
Output: RD: a random dataset with the same size of TD;

```
for i := 1 to |TD| do
    generate a random number mr for mass where |mr| < MT;
    md_i = mt_i + mr;
    generate a random number rr for retention time where |rr| < RT;
    rd_i = rt_i + rr;
    put md_i, rd_i into RD;
end for
return RD
```

best solution, it can work well for the small datasets, such as those less than 10,000 peaks. But for the huge datasets, it may not produce a satisfying result. For each FDR, for example 5%, we can find the corresponding index in the target dataset. By using this index as a threshold, we can count *precision*, *recall*, and *F1-measure*.

7.4 Decoy Dataset

Besides the random dataset method, we also design another approach to estimate the FDR, which is called decoy dataset. Here we still use the same aforementioned random dataset, but we try to simulate the score distribution of a decoy dataset which based on the random dataset. The difference between the random and decoy dataset is that we only select several discrete top results from the former, while for the latter we use the cumulative distribution function (CDF) to calculate the score threshold.

After the processing of a random dataset, we have a score for each protein. Although we can use these scores directly to estimate the FDR, there may be a problem with it. These scores are discrete, which cannot guarantee that they represent the authentic distribution of a random dataset. Therefore, we would like to simulate the score distribution for the random

Algorithm 14: FDR: estimating the false discovery rate roughly.

Input: TPL: a target candidate proteins list;
RPL: a random candidate proteins list;
$F(P, p_j)$: the score of protein p_j for the target dataset;
$F_R(P, p_j)$: the score of protein p_j for the random dataset;
Output: FDR: an estimated false discovery rate;

```
iTarget ← 1;
iRandom ← 1;
while iTarget < |TPL| and iRandom < |RPL| do
    if F(P, p_iTarget) > F_R(P, p_iRandom) then
        iTarget++;
    end if
    else
        FDR=iRandom/(iTarget+iRandom);
        iRandom++;
    end if
end while
return FDR
```

dataset, which could depict the score threshold more accurate for a decoy dataset. Here, we take dataset two as an example, and show how to implement the idea of estimating the FDR for a decoy dataset.

First, we select all the scores for the proteins whose matches are great than 0, and the total number is 5,053. Here, we use "dfittool" in MATLAB to fit the distribution of the protein score. Figure 7.3 illustrates the protein score distribution for dataset two by a decoy dataset. The blue histogram represents density of the proteins with different scores, where the bin width is 0.01. The red curve is the distribution fitted by the generalized extreme value (GEV), which is a family of continuous probability distributions developed within extreme value theory. Secondly, we can evaluate the values by cumulative probability of this fitted distribution. Figure 7.4 evaluates the protein score by the generalized extreme value. For example, for an evaluated protein score of 1.2, the cumulative probability score is 0.99959. When plotting above figure, we can get Figure 7.5, where the cumulative probability is based on the evaluated protein score.

Now, with the cumulative probability value, we can find the corresponding protein score,

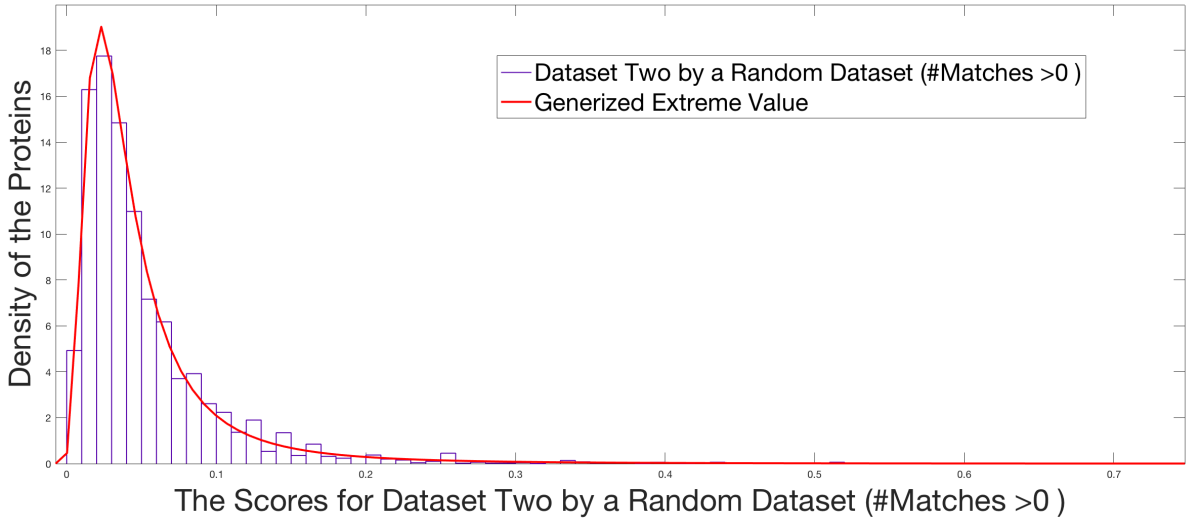

Figure 7.3: The distribution of the protein score (# matches > 0), and its distribution fitted by the generalized extreme value. The blue histogram is the density of the proteins whose scores are within in a small bin, and the red curve is the distribution fitted by the generalized extreme value.

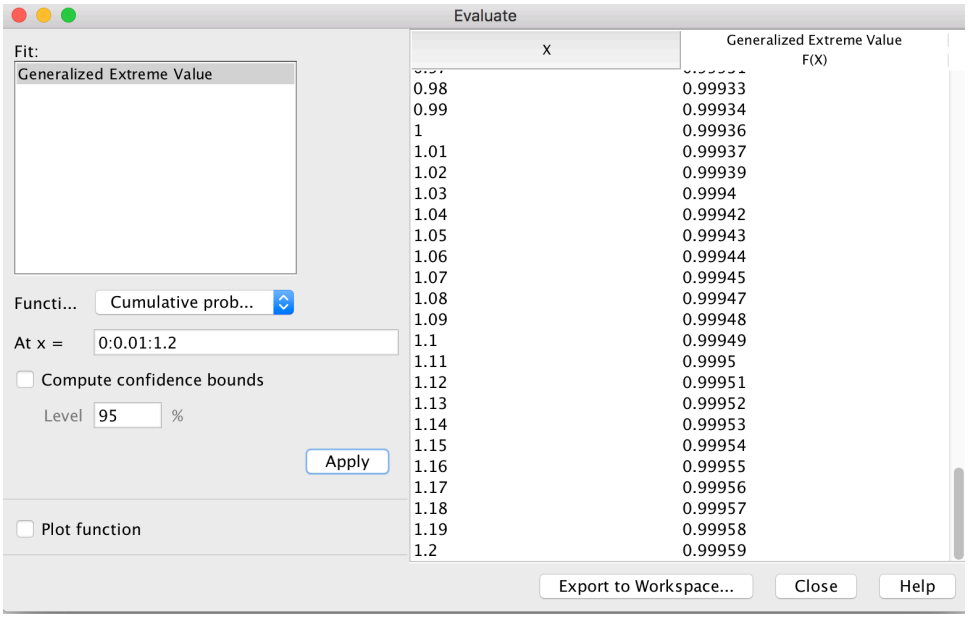

Figure 7.4: The evaluated protein score (# matches > 0) by the generalized extreme value.

Figure 7.5: The distribution fitting evaluation plot for the protein score (# matches > 0) by the generalized extreme value.

and use it in the target database. For example, if we define the cumulative probability score of 99% as our threshold, we will get the protein score of 0.31, which can be used to report the proteins in the target database as a score threshold. Additionally, the protein score is 0.14 for the cumulative probability score of 95%.

With aforementioned method, we can find a protein threshold which can facilitate us to report proteins. But the selected threshold is too low compared to protein scores in the target database. Therefore, we restrict that the number of matches for a protein should be great than 1 or 2, and we can get greater score which is more suitable as a threshold. The total number of proteins (peptide matches that are greater than 1) is 699, and Figure 7.6 illustrates the protein score distribution for dataset two by a decoy dataset. After calculations, the protein score is 0.42 for the cumulative probability score of 99%, and 0.21 for 95%. Additionally, the total number of proteins (peptide matches that are greater than 2) is 61, and Figure 7.7 illustrates it. Consequently, the protein score is 0.59 for 99%, and 0.28 for 95%. Again, we can use thees protein scores as our thresholds in the target database searching.

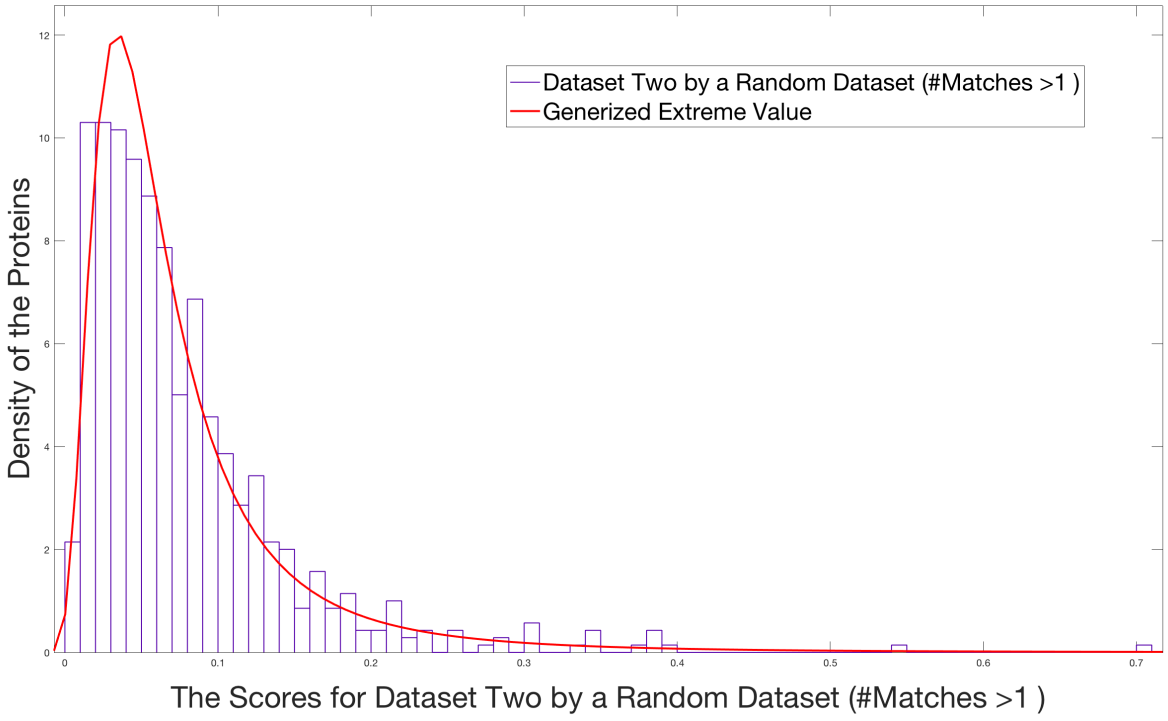

Figure 7.6: The distribution of the protein score (# matches > 1), and its distribution fitted by the generalized extreme value.

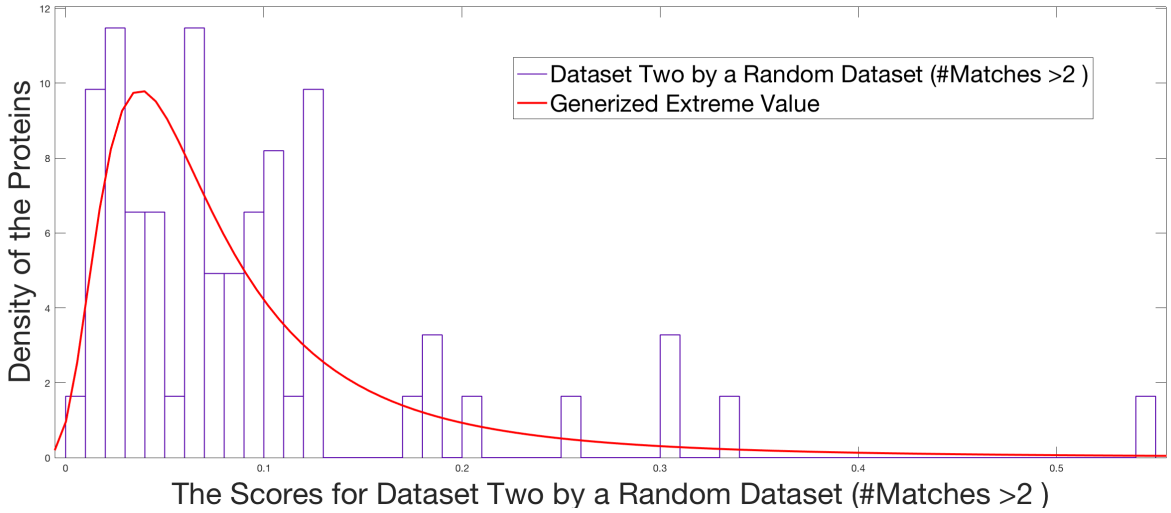

Figure 7.7: The distribution of the protein score (# matches > 2), and its distribution fitted by the generalized extreme value.

7.5 Experiments

Now, we have an input set of experimental peaks P. We also have a database of proteins D. For PMF, our objective is to find a set of proteins $Z \in D$ that contains all the proteins whose scores are greater than a threshold Δ. Additionally, we would like to know the FDR in our identified proteins. Consequently, we use target-decoy strategy to facilitate us for estimating the FDRs.

Here we still use the same three datasets, and generate three random datasets with normal distribution. For each of them, we test two methods mentioned above: random dataset and decoy dataset, to find proper thresholds. The testing results demonstrate that our two approaches could work well under some situations, although they are still not perfect.

7.5.1 Dataset One

Again, we generated the simulated dataset from a wet-lab experiment with 248 proteins for Human, which identified by MS/MS with Mascot. We selected the top 89 proteins and their 770 matched peptides as the input dataset, with a tolerance δ=2ppm. We can identify all the proteins whose matches are greater than two, and the top 61 ones are all TP. The target decoy strategy does not work very well for most of the algorithms. But the best solution, HeatS&ORT&LR&AP&PF, could achieve satisfying result. Here we use normal distribution to generate the random dataset. For each FDR, for example 5%, we can find the corresponding index in the target dataset. By using this index as a threshold, we can count the *precision*, *recall*, and *F1-measure*. The relationship of FDR and other measurements is shown in Table 7.2, where FDR and *precision* can be roughly calculated by a random dataset.

Besides random dataset searching, we also use decoy dataset searching strategy. From aforementioned introduction for decoy dataset, we can estimate the FDR from the distribution of decoy dataset. Here, the protein score is 16.7 for 99.9%, 11.6 for 99.5%, 9.9 for 99%, and 7.0 for 95%, where the matched peptides for the proteins are greater than 0. Additionally, with

Table 7.2: The performance of dataset one for HeatS&ORT&LR&AP&PF by a random dataset

FDR	*Precision*	*Recall*	*F1-measure*
1%	97%	70%	81%
2%	97%	70%	81%
5%	96%	72%	82%
10%	87%	83%	85%

these FDRs, we can set the corresponding score threshold in target database, then calculate the *Precision*, *Recall*, and *F1-measure*. For example, for the protein score of 16.7 in the decoy dataset, we set it as the score threshold in the target dataset. There are total 47 proteins whose scores are greater than 16.7, and 43 of them are TPs. Consequently, the *Precision* is $43/47 = 94\%$, the *Recall* is $43/89 = 49\%$, and the *F1-measure* is $0.94 * 0.49 * 2/(0.94 + 0.49) = 65\%$. The performance of dataset one for HeatS&ORT&LR&AP&PF by a decoy dataset is shown in Table 7.3.

Table 7.3: The performance of dataset one for HeatS&ORT&LR&AP&PF by a decoy dataset

FDR	*Precision*	*Recall*	*F1-measure*
0.1%	94%	49%	65%
0.5%	93%	69%	79%
1%	93%	70%	80%
5%	78%	79%	80%

7.5.2 Dataset Two

The target-decoy strategy does not work for most of the algorithms. But for the best solution, HeatS&ORT&LR, it can work well for the lower estimated FDR, such as 1%. We also add the noise with normal distribution to the target dataset, and get random dataset. The performance of estimated FDR and other measurements by a random dataset is shown in Table 7.4. Here we consider 49 expected proteins and 20 bonus expected proteins as the ground-truth proteins.

Besides random dataset searching, we also use decoy dataset searching strategy. Based

Table 7.4: The performance of dataset two for HeatS&ORT&LR by a random dataset

FDR	*Precision*	*Recall*	*F1-measure*
1%	96%	62%	75%
2%	96%	62%	75%
5%	85%	83%	83%
10%	63%	90%	74%

on aforementioned analysis, for a decoy dataset, the protein score is 1.6 for 99.9%, 0.8 for 99.5%, 0.62 for 99%, and 0.5 for 98%, where the matched peptides for the proteins are greater than 2. Consequently, the performance of dataset two on different estimated FDRs for HeatS&ORT&LR by a decoy dataset is shown in Table 7.5.

Table 7.5: The performance of dataset two for HeatS&ORT&LR by a decoy dataset

FDR	*Precision*	*Recall*	*F1-measure*
0.1%	96%	66%	79%
0.5%	85%	89%	87%
1%	70%	95%	81%
1.5%	65%	96%	78%

7.5.3 Dataset Three

The target-decoy strategy does not work for most of the algorithms. Even for the best solution, HeatS&ORT&LR&AP, it is still not work very well. Here we use normal distribution to generate the random dataset. The performance of FDRs by a random dataset is shown in Table 7.6.

Table 7.6: The performance of dataset three for HeatS&ORT&LR&AP by a random dataset

FDR	*Precision*	*Recall*	*F1-measure*
1%	92%	29%	45%
2%	89%	42%	57%
5%	75%	56%	64%
10%	73%	62%	67%

Besides random dataset searching, we also use decoy dataset searching strategy. From

aforementioned introduction for decoy dataset, for a decoy dataset, the protein score is 1.84 for 99.9%, 1.18 for 99.5%, 0.97 for 99%, 0.6 for 95%, and 0.47 for 90%. Here the proteins are selected from that whose matched peptides are greater than 2. Consequently, the performance of dataset three on different estimated FDRs for HeatS&ORT&LR&AP by a decoy dataset is shown in Table 7.7.

Table 7.7: The performance of dataset three for HeatS&ORT&LR&AP by a decoy dataset

FDR	*Precision*	*Recall*	*F1-measure*
0.1%	93%	16%	28%
0.5%	90%	27%	42%
1%	90%	32%	48%
5%	82%	49%	62%
10%	75%	56%	65%

Chapter 8

Conclusion and Future Work

8.1 Conclusion

We have introduced novel network-based inference (NBI) algorithms for peptide mass fingerprinting (PMF) in a database search. By analyzing peptide-protein bipartite network, we designed new peptide protein matching score functions. We present two methods: the static one, ProbS, models the peptide probability independently; and the dynamic one, HeatS, extracts peptide dependent score from dynamic input dataset. In order to tackle the bias between large and small proteins, we adjust the raw matching score with linear regression, according to the masses of proteins. Additionally, we consider the order of retention time (RT) to further adjust the protein score. Here, a dynamic programming is applied to find a subset with maximum score, whose order of RTs is consistent with that of predicted RTs. In the post processing, we develop two algorithms: assignment of peaks, and protein filtration. The former restricts that a peak can only be assigned to one peptide in order to reduce random matches; and the latter tries to filter out false positive proteins, assuming that each peak can only be assigned to one protein. Moreover, we design two new target-decoy search strategies (random dataset and decoy dataset) to estimate the false discovery rate (FDR), which could facilitate us to report the accuracy of the result. Finally, we demonstrated that our methods achieved significant

improvement in performance compared to several state-of-the-art approaches.

In practice PMF is popular and works well, since it is relatively fast to compute PMF scores when database searching. Although PMF cannot obtain peptide sequence information, it still has some advantages over MS/MS. For example, MALDI-TOF MS is significantly less expensive and faster than MS/MS. Additionally, in the host cell proteins (HCPs) detection, MS and PMF should be used. Because the abundance of HCPs is usually too low to be identified by MS/MS. Moreover, some proteins (such as histones) would be cleaved to short peptides (3-4 amino acid residues) after digestion, which are very hard to be identify by MS/MS. As a result, MS and PMF should be applied for them.

8.2 Future Work

In the future, with the advance of mass accuracy for instrumentation and increase of computer performance, NBPMF may be expected to achieve better accuracy combined with LC-MS/MS for protein identification, compared to using LC-MS or LC-MS/MS separately. For example, trapped ion mobility spectrometry is a relatively new gas-phase separation approach, which provides a powerful separation platform prior to mass analysis [88]. Therefore, for the LC-MS with TIMS, we would have four-dimensional information for each peak, including: mass, RT, intensity, and mobility. As a result, NBPMF would perform better with these new datasets than the former ones.

Another future work is to combine NBPMF with MS/MS to solve protein inference problem. After database searching for MS/MS, we have three kinds of peptides: unique peptides identified by MS/MS, shared peptides identified by MS/MS, and unidentified peptides from MS. We would initialize these three types of peptides with different weights, and adjust them by our NBI algorithms. Now, we could use these modified scores to infer proteins. Consequently, the combination of NBPMF and MS/MS may have the possibility to improve the performance for protein inference, compared to using MS or MS/MS separately.

Bibliography

[1] Aebersold, Ruedi, and Matthias Mann. Mass spectrometry-based proteomics. *Nature* 422, no. 6928 (2003): 198-207.

[2] Alves, Pedro, Randy J. Arnold, Milos V. Novotny, Predrag Radivojac, James P. Reilly, and Haixu Tang. Advancement in protein inference from shotgun proteomics using peptide detectability. *In Pacific Symposium on Biocomputing* vol. 12, pp. 409-420. 2007.

[3] Anderson, Leigh, and Christie L. Hunter. Quantitative mass spectrometric multiple reaction monitoring assays for major plasma proteins. *Molecular & Cellular Proteomics* 5, no. 4 (2006): 573-588.

[4] Anderson, D. C., Weiqun Li, Donald G. Payan, and William Stafford Noble. A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores. *Journal of proteome research* 2, no. 2 (2003): 137-146.

[5] Baczek, Tomasz, and Roman Kaliszan. Predictions of peptides' retention times in reversed-phase liquid chromatography as a new supportive tool to improve protein identification in proteomics. *Proteomics* 9, no. 4 (2009): 835-847.

[6] Bandeira, Nuno, Dekel Tsur, Ari Frank, and Pavel A. Pevzner. Protein identification by spectral networks analysis. *Proceedings of the National Academy of Sciences* 104, no. 15 (2007): 6140-6145.

[7] Barabasi, Albert-Laszlo, and Zoltan N. Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews genetics* 5, no. 2 (2004): 101-113.

[8] Barrow, Mark P., William I. Burkitt, and Peter J. Derrick. Principles of Fourier transform ion cyclotron resonance mass spectrometry and its application in structural biology. *Analyst* 130, no. 1 (2005): 18-28.

[9] Behjati, Sam, and Patrick S. Tarpey. What is next generation sequencing? *Archives of disease in childhood-Education & practice edition* (2013): edpract-2013.

[10] Bianco, Luca, Jennifer A. Mead, and Conrad Bessant. Comparison of novel decoy database designs for optimizing protein identification searches using ABRF sPRG2006 standard MS/MS data sets. *Journal of proteome research* 8, no. 4 (2009): 1782-1791.

[11] Boswell, Paul G., Jonathan R. Schellenberg, Peter W. Carr, Jerry D. Cohen, and Adrian D. Hegeman. Easy and accurate high-performance liquid chromatography retention prediction with different gradients, flow rates, and instruments by back-calculation of gradient and flow rate profiles. *Journal of Chromatography A* 1218, no. 38 (2011): 6742-6749.

[12] Boswell, Paul G., Jonathan R. Schellenberg, Peter W. Carr, Jerry D. Cohen, and Adrian D. Hegeman. A study on retention "projection" as a supplementary means for compound identification by liquid chromatography-mass spectrometry capable of predicting retention with different gradients, flow rates, and instruments. *Journal of Chromatography A* 1218, no. 38 (2011): 6732-6741.

[13] Brenton, A. Gareth, and A. Ruth Godfrey. Accurate mass measurement: terminology and treatment of data. *Journal of the American Society for Mass Spectrometry* 21, no. 11 (2010): 1821-1835.

[14] Brosch, Markus, Lu Yu, Tim Hubbard, and Jyoti Choudhary. Accurate and sensitive peptide identification with Mascot Percolator. *Journal of proteome research* 8, no. 6 (2009): 3176-3181.

[15] Brin, Sergey, and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks* 56, no. 18 (2012): 3825-3833.

[16] Busch, Kenneth L. Units in mass spectrometry. *DigitalCommons © Kennesaw State University*, 2011.

[17] Bochet, Pascal, Frank Rugheimer, Tina Guina, Peter Brooks, David Goodlett, Peter Clote, and Benno Schwikowski. Fragmentation-free LC-MS can identify hundreds of proteins. *Proteomics* 11, no. 1 (2011): 22-32.

[18] Bonet-Costa, Carles, Marta Vilaseca, Claudio Diema, Olivera Vujatovic, Alejandro Vaquero, Nuria Omenaca, Lucia Castejon, Jordi Bernues, Ernest Giralt, and Fernando Azorin. Combined bottom-up and top-down mass spectrometry analysis of the pattern of post-translational modifications of Drosophila melanogaster linker histone H1. *Journal of proteomics* (2012).

[19] Carr, Steven, Ruedi Aebersold, Michael Baldwin, A. L. Burlingame, Karl Clauser, and Alexey Nesvizhskii. The need for guidelines in publication of peptide and protein identification data working group on publication guidelines for peptide and protein identification data. *Molecular & Cellular Proteomics* 3, no. 6 (2004): 531-533.

[20] Chamrad, Daniel C., Gerhard Korting, Kai Stuhler, Helmut E. Meyer, Joachim Klose, and Martin Bluggel. Evaluation of algorithms for protein identification from sequence databases using mass spectrometry data. *Proteomics* 4, no. 3 (2004): 619-628.

[21] Chen, Hailin, Heng Zhang, Zuping Zhang, Yiqin Cao, and Wenliang Tang. Network-based inference methods for drug repositioning. *Computational and mathematical methods in medicine* 2015 (2015).

[22] Chernobrovkin, A. L., O. P. Trifonova, N. A. Petushkova, E. A. Ponomarenko, and A. V. Lisitsa. Selection of the peptide mass tolerance value for protein identification with

peptide mass fingerprinting. *Russian journal of bioorganic chemistry* 37, no. 1 (2011): 119-122.

[23] Choi, Hyungwon, Damian Fermin, and Alexey I. Nesvizhskii. Significance analysis of spectral count data in label-free shotgun proteomics. *Molecular & cellular proteomics* 7, no. 12 (2008): 2373-2385.

[24] Clauser, Karl R., Peter Baker, and Alma L. Burlingame. Role of accurate mass measurement (± 10 ppm) in protein identification strategies employing MS or MS/MS and database searching. *Analytical chemistry* 71, no. 14 (1999): 2871-2882.

[25] Colinge, Jacques, Alexandre Masselot, Marc Giron, Thierry Dessingy, and Jeroe Magnin. OLAV: Towards high-throughput tandem mass spectrometry data identification. *Proteomics* 3, no. 8 (2003): 1454-1463.

[26] Craig, Robertson, and Ronald C. Beavis. A method for reducing the time required to match protein sequences with tandem mass spectra. *Rapid communications in mass spectrometry* 17, no. 20 (2003): 2310-2316.

[27] Craig, Robertson, J. C. Cortens, David Fenyo, and Ronald C. Beavis. Using annotated peptide mass spectrum libraries for protein identification. *Journal of proteome research* 5, no. 8 (2006): 1843-1849.

[28] Crick, F.H.C. On Protein Synthesis. *The Symposia of the Society for Experimental Biology* 12, (1958): 138-163.

[29] Cui, Weidong, Henry W. Rohrs, and Michael L. Gross. Top-down mass spectrometry: recent developments, applications and perspectives. *Analyst* 136, no. 19 (2011): 3854-3864.

[30] Dasari, Surendra, Matthew C. Chambers, Robbert J. Slebos, Lisa J. Zimmerman, Amy-Joan L. Ham, and David L. Tabb. TagRecon: high-throughput mutation identification through sequence tagging. *Journal of proteome research* 9, no. 4 (2010): 1716-1726.

[31] David, Matei, Misko Dzamba, Dan Lister, Lucian Ilie, and Michael Brudno. SHRiMP2: sensitive yet practical short read mapping. Bioinformatics 27, no. 7 (2011): 1011-1012.

[32] Desiere, Frank, Eric W. Deutsch, Nichole L. King, Alexey I. Nesvizhskii, Parag Mallick, Jimmy Eng, Sharon Chen, James Eddes, Sandra N. Loevenich, and Ruedi Aebersold. The peptideatlas project. *Nucleic acids research* 34, no. suppl 1 (2006): D655-D658.

[33] De Las Rivas, Javier, and Celia Fontanillo. Protein-protein interactions essentials: key concepts to building and analyzing interactome networks. *PLoS Comput Biol* 6, no. 6 (2010): e1000807.

[34] Demaine, Erik D., Shay Mozes, Benjamin Rossman, and Oren Weimann. An optimal decomposition algorithm for tree edit distance. *ACM Transactions on Algorithms (TALG)* 6, no. 1 (2009): 2.

[35] Elias, Joshua E., and Steven P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature methods* 4, no. 3 (2007): 207-214.

[36] Elias, Joshua E., and Steven P. Gygi. Target-decoy search strategy for mass spectrometry-based proteomics. *Proteome Bioinformatics* (2010): 55-71.

[37] Eng, Jimmy K., Ashley L. McCormack, and John R. Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry* 5, no. 11 (1994): 976-989.

[38] Fenyo, David, and Ronald C. Beavis. A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Analytical Chemistry* 75, no. 4 (2003): 768-774.

[39] Finn, Robert D., Penelope Coggill, Ruth Y. Eberhardt, Sean R. Eddy, Jaina Mistry, Alex L. Mitchell, Simon C. Potter et al. The Pfam protein families database: towards a more sustainable future. *Nucleic acids research* 44, no. D1 (2016): D279-D285.

[40] Frank, Ari, and Pavel Pevzner. PepNovo: *de novo* peptide sequencing via probabilistic network modeling. *Analytical chemistry* 77, no. 4 (2005): 964-973.

[41] Frank, Ari, Stephen Tanner, Vineet Bafna, and Pavel Pevzner. Peptide sequence tags for fast database search in mass-spectrometry. *Journal of proteome research* 4, no. 4 (2005): 1287-1295.

[42] Frewen, Barbara E., Gennifer E. Merrihew, Christine C. Wu, William Stafford Noble, and Michael J. MacCoss. Analysis of peptide MS/MS spectra from large-scale proteomics experiments using spectrum libraries. *Analytical chemistry* 78, no. 16 (2006): 5678-5684.

[43] Gao, Zhen, Ruizhe Zhao, and Jianhua Ruan. A genome-wide cis-regulatory element discovery method based on promoter sequences and gene co-expression networks. *BMC genomics* 14, no. 1 (2013): S4.

[44] Geer, Lewis Y., Sanford P. Markey, Jeffrey A. Kowalak, Lukas Wagner, Ming Xu, Dawn M. Maynard, Xiaoyu Yang, Wenyao Shi, and Stephen H. Bryant. Open mass spectrometry search algorithm. *Journal of proteome research* 3, no. 5 (2004): 958-964.

[45] Gorshkov, A. V., M. L. Pridatchenko, T. Yu Perlova, I. A. Tarasova, M. V. Gorshkov, and V. V. Evreinov. Applicability of the critical chromatography concept to proteomics problems: I. Effect of the stationary phase and the size of the chromatographic column on the dependence of the retention time of peptides and proteins on the amino acid sequence. *Journal of Analytical Chemistry* 71, no. 1 (2016): 110-125.

[46] Gorg, Angelika, Walter Weiss, and Michael J. Dunn. Current two-dimensional electrophoresis technology for proteomics. *Proteomics* 4, no. 12 (2004): 3665-3685.

[47] Gygil, Steven P., Beate Rist, Scott A. Gerber, Frantisek Turecek, Michael H. Gelb, and Ruedi Aebersold. Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nature biotechnology* 17, no. 10 (1999): 994-999

[48] Han, Xuemei, Aaron Aslanian, and John R. Yates. Mass spectrometry for proteomics. *Current opinion in chemical biology* 12, no. 5 (2008): 483-490.

[49] Han, Xuemei, Mi Jin, Kathrin Breuker, and Fred W. McLafferty. Extending top-down mass spectrometry to proteins with masses greater than 200 kilodaltons. *Science* 314, no. 5796 (2006): 109-112.

[50] He, Zengyou, Chao Yang, Can Yang, Robert Z. Qi, Jason Po-Ming Tam, and Weichuan Yu. Optimization-based peptide mass fingerprinting for protein mixture identification. *Journal of Computational Biology* 17, no. 3 (2010): 221-235.

[51] Henzel, William J., Colin Watanabe, and John T. Stults. Protein identification: the origins of peptide mass fingerprinting. *Journal of the American Society for Mass Spectrometry* 14, no. 9 (2003): 931-942.

[52] Huang, Ting, and Zengyou He. A linear programming model for protein inference problem in shotgun proteomics. *Bioinformatics* 28, no. 22 (2012): 2956-2962.

[53] Huang, Ting, Jingjing Wang, Weichuan Yu, and Zengyou He. Protein inference: a review. *Briefings in bioinformatics* (2012): bbs004.

[54] Ilie, Lucian, Bahlul Haider, Michael Molnar, and Roberto Solis-Oba. SAGE: string-overlap assembly of genomes. *BMC bioinformatics* 15, no. 1 (2014): 302.

[55] Ibrahim, Ashraf, Lian Yang, Chad Johnston, Xiaowen Liu, Bin Ma, and Nathan A. Magarvey. Dereplicating nonribosomal peptides using an informatic search algorithm for

natural products (iSNAP) discovery. *Proceedings of the National Academy of Sciences* 109, no. 47 (2012): 19196-19201.

[56] Jaitly, Navdeep, Anoop Mayampurath, Kyle Littlefield, Joshua N. Adkins, Gordon A. Anderson, and Richard D. Smith. Decon2LS: An open-source software package for automated processing and visualization of high resolution mass spectrometry data. *BMC bioinformatics* 10, no. 1 (2009): 87.

[57] Jansson, Jesper, and Zeshan Peng. Algorithms for finding a most similar subforest. *In Annual Symposium on Combinatorial Pattern Matching*, pp. 377-388. Springer Berlin Heidelberg, 2006.

[58] Jensen, Ole N., Alexandre V. Podtelejnikov, and Matthias Mann. Identification of the components of simple protein mixtures by high-accuracy peptide mass mapping and database searching. *Analytical chemistry* 69, no. 23 (1997): 4741-4750.

[59] Junqueira, Magno, Victor Spirin, Tiago Santana Balbuena, Patrice Waridel, Vineeth Surendranath, Grigoriy Kryukov, Ivan Adzhubei, Henrik Thomas, Shamil Sunyaev, and Andrej Shevchenko. Separating the wheat from the chaff: unbiased filtering of background tandem mass spectra improves protein identification. *Journal of proteome research* 7, no. 8 (2008): 3382-3395.

[60] Krokhin, Oleg V. Sequence-specific retention calculator. Algorithm for peptide retention prediction in ion-pair RP-HPLC: application to 300-and 100-A pore size C18 sorbents. *Analytical chemistry* 78, no. 22 (2006): 7785-7795.

[61] Krokhin, Oleg V., and Vic Spicer. Peptide retention standards and hydrophobicity indexes in reversed-phase high-performance liquid chromatography of peptides. *Analytical chemistry* 81, no. 22 (2009): 9522-9530.

[62] Krokhin, Oleg V., R. Craig, V. Spicer, W. Ens, K. G. Standing, R. C. Beavis, and J. A. Wilkins. An Improved Model for Prediction of Retention Times of Tryptic Peptides in

Ion Pair Reversed-phase HPLC Its Application to Protein Peptide Mapping by Off-Line HPLC-MALDI MS. *Molecular & Cellular Proteomics* 3, no. 9 (2004): 908-919.

[63] Lam, Henry, Eric W. Deutsch, James S. Eddes, Jimmy K. Eng, Stephen E. Stein, and Ruedi Aebersold. Building consensus spectral libraries for peptide identification in proteomics. *Nature methods* 5, no. 10 (2008): 873-875.

[64] Lane, W. S., A. I. Nesvizhskii, B. Searle, D. L. Tabb, J. A. Kowalak, and S. L. Seymour. Bioinformatic Evaluation of Datasets Derived from the ABRF sPRG Proteomics Standard. *ABRF* (2007).

[65] Lanucara, Francesco, and Claire E. Eyers. Top-down mass spectrometry for the analysis of combinatorial post-translational modifications. *Mass Spectrometry Reviews* (2012).

[66] Li, Yifeng, Chih-yu Chen, Alice M. Kaye, and Wyeth W. Wasserman. The identification of cis-regulatory elements: A review from a machine learning perspective. *Biosystems* 138 (2015): 6-17.

[67] Li, Yiwei, and Lucian Ilie. SPRINT: Ultrafast protein-protein interaction prediction of the entire human interactome. *arXiv preprint arXiv*:1705.06848 (2017).

[68] Liang, Zhewei. Efficient algorithms for local forest similarity. *PhD diss., Thesis (M. Sc), School of Graduate and Postdoctoral Studies*, University of Western Ontario, London, Ontario, Canada, 2011.

[69] Liang, Zhewei, Gilles Lajoie, and Kaizhong Zhang. NBPMF: Novel network-based inference methods for peptide mass fingerprinting. *In 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pp. 213-219. IEEE, 2017.

[70] Liang, Zhewei, Gilles Lajoie, and Kaizhong Zhang. NBPMF: Novel Peptide Mass Fingerprinting Based on Network Inference. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)* 11, no. 4 (2017): 41-65.

[71] Liang, Zhewei, and Kaizhong Zhang. Algorithms for Forest Local Similarity. *Combinatorial Optimization and Applications* (2012): 163-175.

[72] Liang, Zhewei, and Kaizhong Zhang. Algorithms for local similarity between forests. *Journal of Combinatorial Optimization* 27, no. 1 (2014): 14-31.

[73] Lipton, Mary S., Ljiljana Pasa-Tolic, Gordon A. Anderson, David J. Anderson, Deanna L. Auberry, John R. Battista, Michael J. Daly et al. Global analysis of the Deinococcus radiodurans proteome by using accurate mass tags. *Proceedings of the National Academy of Sciences* 99, no. 17 (2002): 11049-11054.

[74] Link, Andrew J., Jimmy Eng, David M. Schieltz, Edwin Carmack, Gregory J. Mize, David R. Morris, Barbara M. Garvik, and John R. Yates III. Direct analysis of protein complexes using mass spectrometry. *Nature biotechnology* 17, no. 7 (1999): 676-682.

[75] Liu, Yi, Bin Ma, Kaizhong Zhang, and Gilles Lajoie. An approach for peptide identification by de novo sequencing of mixture spectra. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 14, no. 2 (2017): 326-336.

[76] Liu, Yi, Weiping Sun, Gilles Lajoie, Bin Ma, and Kaizhong Zhang. An Approach for Matching Mixture MS/MS Spectra with a Pair of Peptide Sequences in a Protein Database. *In International Symposium on Bioinformatics Research and Applications*, pp. 223-234. Springer, Cham, 2015.

[77] Liu, Yi, Weiping Sun, Baozhen Shan, and Kaizhong Zhang. DISC: DISulfide linkage Characterization from tandem mass spectra. *Bioinformatics* 33, no. 23 (2017): 3861-3870.

[78] Lu, Bingwen, Akira Motoyama, Cristian Ruse, John Venable, and John R. Yates III. Improving protein identification sensitivity by combining MS and MS/MS information for shotgun proteomics using LTQ-Orbitrap high mass accuracy data. *Analytical chemistry* 80, no. 6 (2008): 2018-2025.

[79] Lu, Linyuan, Matus Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics Reports* 519, no. 1 (2012): 1-49.

[80] Ma, Bin. Challenges in computational analysis of mass spectrometry data for proteomics. *Journal of Computer Science and Technology* 25, no. 1 (2010): 107-123.

[81] Ma, Bin, and Richard Johnson. De novo sequencing and homology searching. *Molecular & Cellular Proteomics* 11, no. 2 (2012).

[82] Ma, Bin, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. PEAKS: powerful software for peptide *de novo* sequencing by tandem mass spectrometry. *Rapid communications in mass spectrometry* 17, no. 20 (2003): 2337-2342.

[83] Matthiesen, Rune. *Mass spectrometry data analysis in proteomics.* Vol. 367. Springer, 2007.

[84] Mann, Matthias, Peter Hojrup, and Peter Roepstorff. Use of mass spectrometric molecular weight information to identify proteins in sequence databases. *Biological Mass Spectrometry* 22, no. 6 (2003): 338-345.

[85] Mann, M., and M. Wilm. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. *Analytical chemistry* 66, no. 24 (1994): 4390-4399.

[86] McHugh, Leo, and Jonathan W. Arthur. Computational methods for protein identification from mass spectrometry data. *PLoS Comput Biol* 4, no. 2 (2008): e12.

[87] McKenna, Aaron, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome research* 20, no. 9 (2010): 1297-1303.

[88] Michelmann, Karsten, Joshua A. Silveira, Mark E. Ridgeway, and Melvin A. Park. Fundamentals of trapped ion mobility spectrometry. *Journal of The American Society for Mass Spectrometry* 26, no. 1 (2015): 14-24.

[89] Millares, Paul, E. James LaCourse, Samirah Perally, Deborah A. Ward, Mark C. Prescott, Jane E. Hodgkinson, Peter M. Brophy, and Huw H. Rees. Proteomic profiling and protein identification by MALDI-TOF mass spectrometry in unsequenced parasitic nematodes. *PloS one* 7, no. 3 (2012): e33590.

[90] Murray, Kermit K., Robert K. Boyd, Marcos N. Eberlin, G. John Langley, Liang Li, and Yasuhide Naito. Definitions of terms relating to mass spectrometry (IUPAC Recommendations 2013). *Pure and Applied Chemistry* 85, no. 7 (2013): 1515-1609.

[91] Molnar, Michael, and Lucian Ilie. Correcting illumina data. *Briefings in bioinformatics* (2014): bbu029.

[92] Monroe, Matthew E., Nikola Tolic, Navdeep Jaitly, Jason L. Shaw, Joshua N. Adkins, and Richard D. Smith. VIPER: an advanced software package to support high-throughput LC-MS peptide identification. *Bioinformatics* 23, no. 15 (2007): 2021-2023.

[93] Moruz, Luminita, and Lukas Kall. Peptide retention time prediction. *Mass spectrometry reviews* (2016).

[94] Moruz, Luminita, Michael R. Hoopmann, Magnus Rosenlund, Viktor Granholm, Robert L. Moritz, and Lukas Kall. Mass fingerprinting of complex mixtures: protein inference from high-resolution peptide masses and predicted retention times. *Journal of proteome research* 12, no. 12 (2013): 5730-5741.

[95] Nesvizhskii, Alexey I. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of proteomics* 73, no. 11 (2010): 2092-2123.

[96] Nesvizhskii, Alexey I., and Ruedi Aebersold. Interpretation of shotgun proteomic data the protein inference problem. *Molecular & cellular proteomics* 4, no. 10 (2005): 1419-1440.

[97] Nesvizhskii, Alexey I., Andrew Keller, Eugene Kolker, and Ruedi Aebersold. A statistical model for identifying proteins by tandem mass spectrometry. *Analytical chemistry* 75, no. 17 (2003): 4646-4658.

[98] Nissen, Poul, Jeffrey Hansen, Nenad Ban, Peter B. Moore, and Thomas A. Steitz. The structural basis of ribosome activity in peptide bond synthesis. *Science* 289, no. 5481 (2000): 920-930.

[99] Nusinow, David P., Adam Kiezun, Daniel J. O'Connell, Joel M. Chick, Yingzi Yue, Richard L. Maas, Steven P. Gygi, and Shamil R. Sunyaev. Network-based inference from complex proteomic mixtures using SNIPE. *Bioinformatics* 28, no. 23 (2012): 3115-3122.

[100] Pappin, D. J., P. Hojrup, and A. J. Bleasby. Rapid identification of proteins by peptide-mass fingerprinting. *Current biology*: CB 3, no. 6 (1993): 327.

[101] Patterson, Scott D., and Ruedi Aebersold. Mass spectrometric approaches for the identification of gel-separated proteins. *Electrophoresis* 16, no. 1 (1995): 1791-1814.

[102] Peng, Zeshan. Algorithms for local forest similarity. *Algorithms and Computation* (2005): 704-713.

[103] Perkins, David N., Darryl J.C. Pappin, David M. Creasy, John S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* 20, no. 18 (1999): 3551-3567.

[104] Picotti, Paola, Bernd Bodenmiller, Lukas N. Mueller, Bruno Domon, and Ruedi Aebersold. Full Dynamic Range Proteome Analysis of *S. cerevisiae* by Targeted Proteomics. *Cell* 138, no. 4 (2009): 795-806.

[105] Pluskal, Tomas, Sandra Castillo, Alejandro Villar-Briones, and Matej Oresic. MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC bioinformatics* 11, no. 1 (2010): 395.

[106] Ramakrishnan, Smriti R., Christine Vogel, John T. Prince, Rong Wang, Zhihua Li, Luiz O. Penalva, Margaret Myers, Edward M. Marcotte, and Daniel P. Miranker. Integrating shotgun proteomics and mRNA expression data to improve protein identification. *Bioinformatics* 25, no. 11 (2009): 1397-1403.

[107] Reid, Gavin E., and Scott A. McLuckey. 'Top down' protein characterization via tandem mass spectrometry. *Journal of mass spectrometry* 37, no. 7 (2002): 663-675.

[108] Rich, Alexander. The Era of RNA Awakening: Structural biology of RNA in the early years. *Quarterly reviews of biophysics* 42, no. 02 (2009): 117-137.

[109] Samuelsson, Jim, Daniel Dalevi, Fredrik Levander, and Thorsteinn Rognvaldsson. Modular, scriptable and automated analysis tools for high-throughput peptide mass fingerprinting. *Bioinformatics* 20, no. 18 (2004): 3628-3635.

[110] Sanger, Frederick, Steven Nicklen, and Alan R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences* 74, no. 12 (1977): 5463-5467.

[111] Shendure, Jay, and Hanlee Ji. Next-generation DNA sequencing. *Nature biotechnology* 26, no. 10 (2008): 1135-1145.

[112] Shevchenko, Andrej, Matthias Wilm, Ole Vorm, and Matthias Mann. Mass spectrometric sequencing of proteins from silver-stained polyacrylamide gels. *Analytical chemistry* 68, no. 5 (1996): 850-858.

[113] Shi, Jinhong, and Fang-Xiang Wu. Protein inference by assembling peptides identified from tandem mass spectra. *Current Bioinformatics* 4, no. 3 (2009): 226-233.

[114] Siuti, Nertila, and Neil L. Kelleher. Decoding protein modifications using top-down mass spectrometry. *Nature methods* 4, no. 10 (2007): 817-821.

[115] Smith, Anthony David, Satya Prakash Datta, and G. Howard Smith. Oxford dictionary of biochemistry and molecular biology. *Oxford University Press*, 1997.

[116] Smith, Richard D., Gordon A. Anderson, Mary S. Lipton, Ljiljana Pasa-Tolic, Yufeng Shen, Thomas P. Conrads, Timothy D. Veenstra, and Harold R. Udseth. An accurate mass tag strategy for quantitative and high-throughput proteome measurements. *Proteomics* 2, no. 5 (2002): 513-523.

[117] Smith, Temple F., and Michael S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology* 147, no. 1 (1981): 195-197.

[118] Spivak, Marina, Jason Weston, Daniela Tomazela, Michael J. MacCoss, and William Stafford Noble. Direct maximization of protein identifications from tandem mass spectra. *Molecular & Cellular Proteomics* 11, no. 2 (2012): M111-012161.

[119] Stein, Stephen E., and Donald R. Scott. Optimization and testing of mass spectral library search algorithms for compound identification. *Journal of the American Society for Mass Spectrometry* 5, no. 9 (1994): 859-866.

[120] Sun, Weiping, Gilles A. Lajoie, Bin Ma, and Kaizhong Zhang. A Novel Algorithm for Glycan de novo Sequencing Using Tandem Mass Spectrometry. *In International Symposium on Bioinformatics Research and Applications*, pp. 320-330. Springer, Cham, 2015.

[121] Tabb, David L., Anita Saraf, and John R. Yates III. GutenTag: high-throughput sequence tagging via an empirically derived fragmentation model. *Analytical chemistry* 75, no. 23 (2003): 6415-6421.

[122] Taylor, J. Alex, and Richard S. Johnson. Implementation and uses of automated *de novo* peptide sequencing by tandem mass spectrometry. *Analytical chemistry* 73, no. 11 (2001): 2594-2604.

[123] Teng, Ben, Can Zhao, Xiaoqing Liu, and Zengyou He. Network inference from ap-ms data: computational challenges and solutions. *Briefings in bioinformatics* (2014): bbu038.

[124] Tran, John C., Leonid Zamdborg, Dorothy R. Ahlf, Ji Eun Lee, Adam D. Catherman, Kenneth R. Durbin, Jeremiah D. Tipton et al. Mapping intact protein isoforms in discovery mode using top-down proteomics. *Nature* 480, no. 7376 (2011): 254-258.

[125] Tuloup, M., Hernandez, C., Coro, I., Hoogland, C., Binz, P.A., Appel, R.D. Aldente and BioGraph: an improved peptide mass fingerprinting protein identification environment. *In: Swiss Proteomics Society 2003 Congress:* Understanding biological systems through proteomics, (2003):147-176.

[126] Wang, Guanghui, Wells W. Wu, Zheng Zhang, Shyama Masilamani, and Rong-Fong Shen. Decoy methods for assessing false positives and false discovery rates in shotgun proteomics. *Analytical chemistry* 81, no. 1 (2008): 146-159.

[127] Wang, Xing, Alan K. Hunter, and Ned M. Mozier. Host cell proteins in biologics development: Identification, quantitation and risk assessment. *Biotechnology and bioengineering* 103, no. 3 (2009): 446-458.

[128] Washburn, Michael P., Dirk Wolters, and John R. Yates III. Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nature biotechnology* 19, no. 3 (2001): 242-247.

[129] Yan, Yan, Anthony J Kusalik, and Fang-Xiang Wu. Recent developments in computational methods for de novo peptide sequencing from tandem mass spectrometry (MS/MS). *Protein and peptide letters* 22, no. 11 (2015): 983-991.

[130] Yan, Yan, and Kaizhong Zhang. Spectra library assisted de novo peptide sequencing for HCD and ETD spectra pairs. *BMC bioinformatics* 17, no. 17 (2016): 538.

[131] Yang, Bo, Junying Zhang, Yaling Yin, and Yuanyuan Zhang. Network-based inference framework for identifying cancer genes from gene expression data. *BioMed research international* 2013 (2013).

[132] Yates, John R., Scott F. Morgan, Christine L. Gatlin, Patrick R. Griffin, and Jimmy K. Eng. Method to compare collision-induced dissociation spectra of peptides: potential for library searching and subtractive analysis. *Analytical chemistry* 70, no. 17 (1998): 3557-3565.

[133] Zhang, Bing, Matthew C. Chambers, and David L. Tabb. Proteomic parsimony through bipartite graph analysis improves accuracy and transparency. *Journal of proteome research* 6, no. 9 (2007): 3549-3557.

[134] Zhang, Jing, Lei Xin, Baozhen Shan, Weiwu Chen, Mingjie Xie, Denis Yuen, Weiming Zhang, Zefeng Zhang, Gilles A. Lajoie, and Bin Ma. PEAKS DB: De Novo sequencing assisted database search for sensitive and accurate peptide identification. *Molecular & Cellular Proteomics* 11, no. 4 (2012).

[135] Zhang, Kaizhong, and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing* 18, no. 6 (1989): 1245-1262.

[136] Zhang, Kaizhong, and Yunkun Zhu. Algorithms for Forest Pattern Matching. *In Proceedings of the 21th Symposium on Combinatorial Pattern Matching* (CPM), pages 1–12, 2010.

[137] Zhang, Wenzhu, and Brian T. Chait. ProFound: an expert system for protein identification using mass spectrometric peptide mapping information. *Analytical chemistry* 72, no. 11 (2000): 2482-2489.

[138] Zhao, Can, Dao Liu, Ben Teng, and Zengyou He. BagReg: Protein inference through machine learning. *Computational biology and chemistry* 57 (2015): 12-20.

[139] Zhou, Tao, L-L. Jiang, R-Q. Su, and Y-C. Zhang. Effect of initial configuration on network-based recommendation. *EPL (Europhysics Letters)* 81, no. 5 (2008): 58004.

[140] Zhou, Tao, Zoltan Kuscsik, Jian-Guo Liu, Matus Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107, no. 10 (2010): 4511-4515.

[141] Zhou, Tao, Jie Ren, Matus Medo, and Yi-Cheng Zhang. Bipartite network projection and personal recommendation. *Physical Review E* 76, no. 4 (2007): 046115.

[142] Zhou, Tao, Ri-Qi Su, Run-Ran Liu, Luo-Luo Jiang, Bing-Hong Wang, and Yi-Cheng Zhang. Accurate and diverse recommendations via eliminating redundant correlations. *New Journal of Physics* 11, no. 12 (2009): 123008.

[143] Zhu, Yunkun. Algorithms for Forest Pattern Matching and Local Forest Similarity. *Thesis(M.Sc), School of Graduate and Postdoctoral Studies*, University of Western Ontario, London, Ontario, Canada, 2010.

[144] http://bioinfo.bisr.res.in/project/crat/pictures/codon.jpg

[145] http://www.schulich.uwo.ca/lrpc/bmsl/instrumentation/index.html

[146] http://www.bioinfor.com

[147] https://www.ebi.ac.uk/

[148] National Center for Biotechnology Information. *Fasta format description.* Available: http://www.ncbi.nlm.nih.gov/blast/fasta.shtml. Accessed 15 December 2007.

[149] https://geneed.nlm.nih.gov/

[150] https://courses.lumenlearning.com/microbiology/chapter/structure-and-function-of-rna/

[151] Motifs database. http://subviral.med.uottawa.ca/cgi-bin/motifs.cgi.

[152] https://ghr.nlm.nih.gov

[153] http://science-explained.com/theory/dna-rna-and-protein/

[154] http://www.uniprot.org/

[155] https://tools.thermofisher.com

[156] https://en.wikipedia.org/wiki

Curriculum Vitae

Name:	Zhewei Liang
Place of Birth:	Hunan, China
Post-secondary Education and Degrees:	The University of Western Ontario London, Ontario, Canada Ph.D Candidate, 2011-date M.Sc, 2009-2011
	Xiangtan University Xiangtan, Hunan, China B.Eng, 1997-2001
Selected Honours and Awards:	Gold Award and Best Innovation Award in Pan-Pearl River Delta Region Universities IT Project Competition, China, Supervisor, 2007
	Computer Systems Analyst, China, 2004
	First Class Prize in China Undergraduate Mathematical Contest in Modeling, 1999
Selected Work Experience:	Teaching Assistant & Research Assistant The University of Western Ontario, 2009-2017
	Assistant Professor Xiangtan University, 2006-2009
	Teaching Assistant Xiangtan University, 2001-2006

Selected Publications:

Liang, Zhewei, Gilles Lajoie, and Kaizhong Zhang.
NBPMF: Novel Peptide Mass Fingerprinting Based on Network Inference.
International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) 11, no. 4 (2017): 41-65.

Liang, Zhewei, Gilles Lajoie, and Kaizhong Zhang.
NBPMF: Novel network-based inference methods for peptide mass fingerprinting.
In 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI CC)*, pp. 213-219. IEEE, 2017.

Liang, Zhewei, and Kaizhong Zhang.
Algorithms for local similarity between forests.
Journal of Combinatorial Optimization 27, no. 1 (2014): 14-31.

Liang, Zhewei, and Kaizhong Zhang.
Algorithms for Forest Local Similarity.
Combinatorial Optimization and Applications (2012): 163-175.
Student Travel Award.