2013

QuasiNovo: Algorithms for De Novo Peptide Sequencing

James Paul Cleveland
University of South Carolina

QuasiNovo: Algorithms for De Novo Peptide Sequencing

By

James Paul Cleveland

Bachelor of Science
University of South Carolina 2007

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosphy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2013

Accepted by:

John Rose, Major Professor

Homayoun Valafar, Committee Member

Ian Dryden, Committee Member

Max Alekseyev, Committee Member

Jose Vidal, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

Abstract

High-throughput proteomics analysis involves the rapid identification and characterization of large sets of proteins in complex biological samples. Tandem mass spectrometry (MS/MS) has become the leading approach for the experimental identification of proteins. Accurate analysis of the data produced is a computationally challenging process that relies on a complex understanding of molecular dynamics, signal processing, and pattern classification. In this work we address these modeling and classification problems, and introduce an additional data-driven evolutionary information source into the analysis pipeline.

The particular problem being solved is peptide sequencing via MS/MS. The objective in solving this problem is to decipher the amino acid sequence of digested proteins (peptides) from the MS/MS spectra produced in a typical experimental protocol. Our approach sequences peptides using only the information contained in the experimental spectrum (de novo) and distributions of amino acid usage learned from large sets of protein sequence data. In this dissertation we pursue three main objectives: an ion classifier based on a neural network which selects informative ions from the spectrum, a peptide sequencer which uses dynamic programming and a scoring function to generate candidate peptide sequences, and a candidate peptide scoring function. Candidate peptide sequences are generated via a dynamic programming graph algorithm, and then scored using a combination of the neural network score, the amino acid usage score, and an edge frequency score. In addition to a complete de novo peptide sequencer, we also examine the use of amino acid usage models independently for reranking candidate peptides.

Contents

List of Tables

List of Figures

Chapter 1

Proteomics and Tandem Mass Spectrometry

1.1 Introduction

Proteins are the building blocks of the machinery of life. They are the principal components of the protoplasm of all cells, and the principal components of the physiological metabolic pathways of all cells. All proteins are built from chain-like polymers whose subunits are the twenty amino acids. The amino acids can be connected together in any order to form an infinite variety of proteins, or one of millions of known proteins with a bewildering array of complex structure and function. The backbone of a protein is a chain of carbon and nitrogen atoms having the pattern ···N-C-C-N-C-C-N-C-C···· with the N-C-C subunit common to all amino acids. The uniqueness of each amino acid is due to differences in the side chain that is attached to the N-C-C group. The structure, chemical reactivity, and function of a protein is determined by its folded three-dimensional structure, which is uniquely determined by the sequence of amino acids.

Proteomics is the large-scale characterization and identification of proteins, in particular their sequence, structure, and function. Unlike Genomics, where an organism's genome is relatively constant, the proteome varies depending on cell type and the physiological state of the cell. Through genetic variations, alternatively spliced RNA transcripts, and posttranslational modifications, a single gene can code for many different molecular forms of a protein (called proteoforms). Thus, the proteome is defined as the proteins present in a tissue sample, organism, or cell culture at a par-

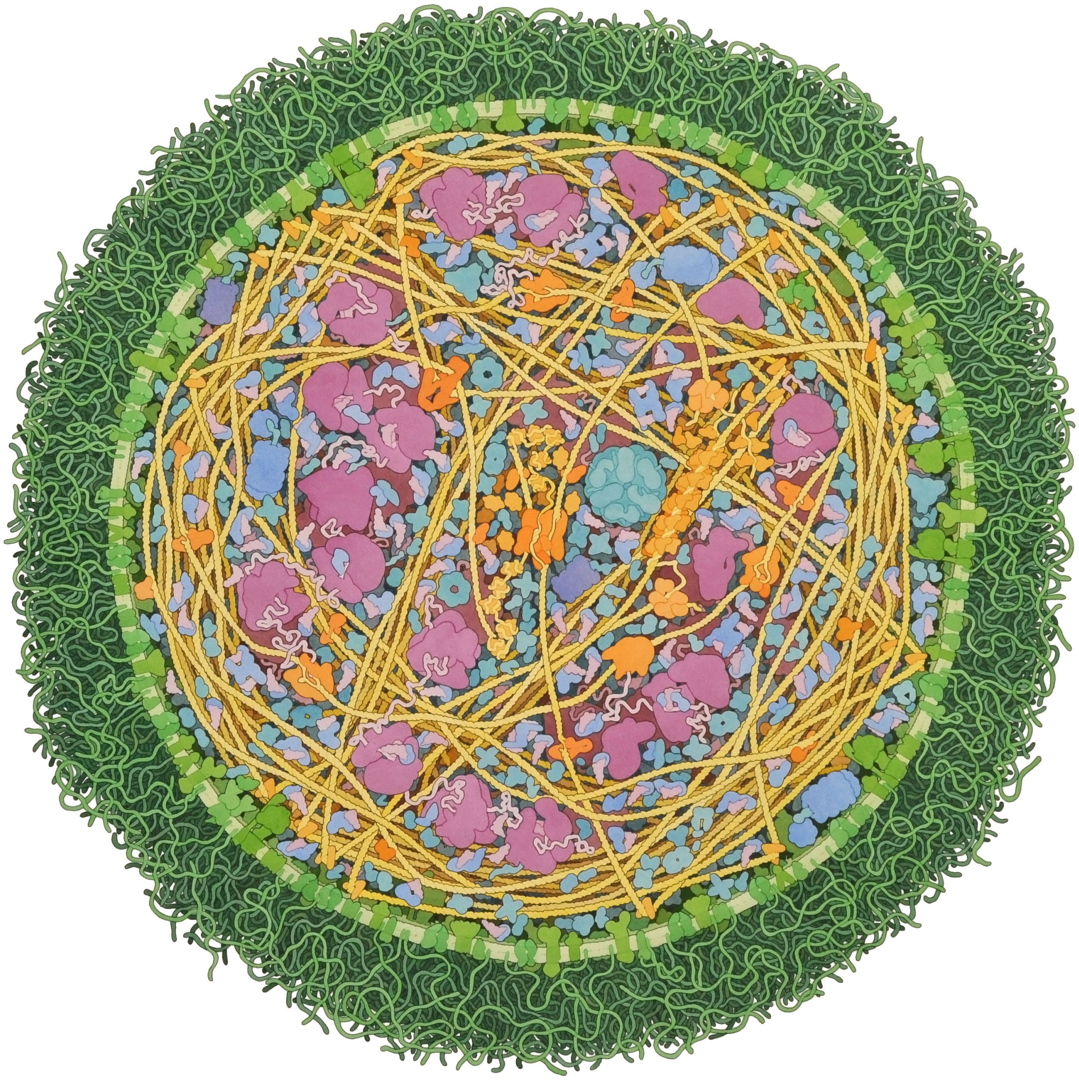

Figure 1.1: Mycoplasma mycoides and its molecular machinery. Illustration by David S. Goodsell, 2011, The Scripps Research Institute.

ticular point in time. Genomics begins with the gene and makes predictions about its protein products, whereas proteomics begins with the *in vitro* protein and works back to the gene or organism responsible for its production. While proteomics is complementary to genomics, it is far more complex.

Mass spectrometry (MS) technology allows proteins to be analyzed rapidly and with high sensitivity at a relatively low cost, and serves as the experimental foundation of the study of proteins. In what is generally known as "bottom-up" proteomics, proteins are enzymatically digested into smaller peptides using a protease such as trypsin. These peptides are then introduced into the mass spectrometer and identified by peptide mass fingerprinting or tandem mass spectrometry. Due to its success in the postgenomic era, tandem mass spectrometry (MS/MS) has become the established approach for the identification and characterization of protein primary sequence. High accuracy is generally achieved by comparing the proteolytic peptide's MS/MS spectra with theoretical spectra based on genomic predictions of proteins from a sequence database, or by comparing the MS/MS spectra to spectra in an annotated peptide spectral library. However there are significant drawbacks to this "database" approach. This approach only works when the peptide is present in a sequence database, *i.e.,* the genome of the organism that produced the protein, or a close homologue, has been published. If the organism is novel and there are no homologous proteins having the exact amino acid sequence of the peptide as a substring, then the database approach is unlikely to produce correct or reliable results. Even if the gene that produced the protein is present in a sequence database, the database approach may fail due to modifications of the primary sequence that cannot be predicted from the gene sequence. Not only does translation from mRNA potentially cause differences, but many proteins are also subjected to chemical modifications after translation, which are critical to the protein's function. In the event of post translational modifications (PTMs) the database approach is again unlikely to produce correct results.

The focus of this dissertation is de novo peptide sequencing, which attempts to solve the above problems by sequencing the peptide using the MS/MS spectrum alone. De novo approaches suffer from their own problems that will be discussed in more depth in the following chapters. While protein identification via MS/MS has yielded a diverse set of experimental methods over the years, analyzing the data remains a weak point in the process. Noisy data and inadequate modeling of molecular dynamics create computational challenges that limit the accuracy of peptide sequencing via MS/MS. Physicochemical and structural properties of proteins and peptides combined with MS experimental conditions can yield unpredictable outcomes. The inability to control or understand these cumulative effects makes peptide sequencing an inherently difficult problem. In this dissertation I will describe several novel approaches and algorithms to de novo peptide sequencing.

Our novel contribution to the field of computational mass spectrometry is a software package called *QuasiNovo*, which is summarized as follows. First, an ion classifier selects informative peaks from the MS/MS spectrum. Our approach uses a staged neural network to model ion fragmentation patterns and estimate the posterior probability of each ion type. This work yielded two conference publications[6, 7] and a journal manuscript that is currently under peer review. Second, a novel scoring technique is used for reranking candidate peptides produced by a modified standard dynamic programming approach. The scoring function integrates the fragmentation model, an amino acid usage model (making this approach a *quasi* de novo sequencing algorithm), a novel edge-frequency score, and a pair-wise cleavage frequency score. Preliminary results for reranking candidate peptides based on amino acid usage yielded a single conference paper.[36]

1.2 Overview of Dissertation

The Dissertation is arranged as follows. Chapter 2 introduces MS/MS, the typical experimental protocol, and the data generation process, specifically peptide fragmentation via collision induced dissociation. Understanding peptide fragmentation is key to developing an ion classifier. We will then outline existing approaches to peptide sequencing.

The peptide sequencing problem is to derive the correct sequence of amino acids for peptide given its MS/MS spectrum. De novo peptide sequencing follows a general formulation: A peptide is fragmented in a mass spectrometer producing a mass spectrum. From this experimental spectrum peaks that likely correspond to b-/y-ions are selected. Chapter 3 describes our novel ion classification (peak selection) approach. These peaks are then used to generate candidate peptides (peptides that may have generated the experimental spectrum). Finally, the candidate peptides are scored and the best candidate is selected as the best peptide-spectrum match. Chapter 4 describes the procedure for creating a spectrum graph, which is used to generate candidate peptides. Chapter 5 introduces a novel candidate peptide scoring function that is used to select the most likely candidate peptide that generated the spectrum. Chapter 6 explores amino acid usage models and their use in reranking sets of candidate peptides produced by arbitrary means. Chapter 7 concludes the dissertation and discusses future work.

Chapter 2

Background

2.1 MS/MS Experimental Protocol

A mixture of proteins is digested into peptides by enzymes. Peptides of interest are then separated through various wet lab methods, and then analyzed via tandem mass spectrometry. The MS/MS experiment consists of two sequential MS runs (Figure 2.1). The objective of the first run (MS) is to isolate peptides by their mass and charge. The objective of the second run (MS/MS) is to analyze a peptide of interest. In the first MS step the mass spectrometer ionizes the peptides and measures their mass/charge (m/z) ratios yielding a mass spectrum. The x-axis corresponds to the m/z ratio and the y-axis corresponds to the relative abundance of the ion. In the second MS/MS run, a peptide (*precursor ion*) from the MS spectrum is selected for analysis. Peptides with the m/z ratio corresponding to the selected peak are then fragmented by either *collision induced dissociation* (CID) or *electron transfer dissociation* (ETD), producing the MS/MS spectrum containing the product ions. Analysis of the spectrum can often yield the sequence of the peptide that generated the MS/MS spectrum.

The MS/MS spectrum of a peptide is determined by its sequence, its charge, and its energy. The goal of peptide sequencing via MS/MS is to determine the peptide that most likely produced the experimental spectrum.

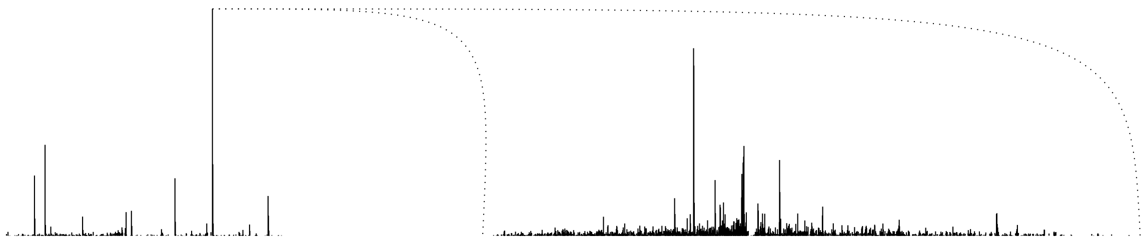

Figure 2.1: MS and MS/MS spectra. A peak from the MS spectrum (left) is selected for fragmentation, producing the MS/MS spectrum (right).

CID MS/MS Spectra

In CID a large number of peptides are ionized and fragmented. A peptide bond at a random position is broken (cleaved) and each peptide is fragmented into two *complementary* product ions: an N-terminal *b-ion* and a C-terminal *y-ion*. The pairwise cleavage frequency between amino acids can vary depending on the N-terminal and C-terminal amino acids, and peptide composition. Figures A.1,A.2,A.3, and A.4 show the pairwise cleavage frequency for several motifs, which vary by peptide composition.

Numerous other types of ions are also produced during fragmentation (Figure 2.2). Detailed descriptions of peptide fragmentation can be found in the literature [20, 21, 40, 45, 31, 23].

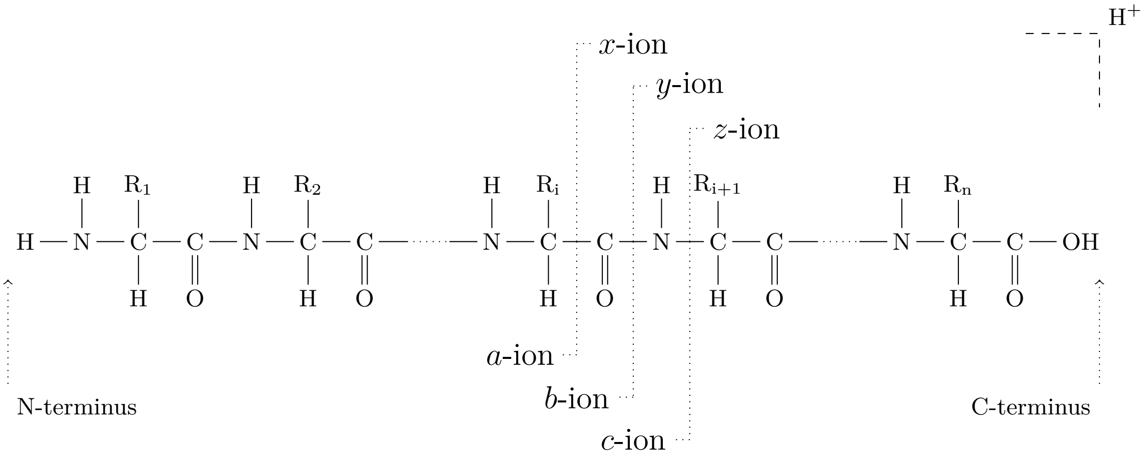

Figure 2.2: Cleavage positions of the ion types.

The relative abundances of different ion types in MS/MS spectra depend on many factors, including instrumentation, peptide structure, and collision energy. Lower-energy CID tends to produce b-/y-ions, internal ions, and immonium ions, while higher-energy CID tends to produce these and the additional additional c, d, v, w, x, and z ion types. [45]

Variability in MS/MS spectra

Fragmentation characteristics are often highly instrument dependant. Different instruments can have different m/z ranges for ion detection. Ion abundance can vary due to ion ion counting or shot noise, and is more pronounced for low intensity peaks resulting in low signal-to-noise. These peaks can disappear unpredictably when they fall below a certain detection threshold. Impurities in the precursor ion can occur when multiple precursor ions fall within the instrument's precursor ion m/z tolerance. This results in unexplained peaks in a spectrum. Due to the variability in MS/MS spectra any machine learning attempts to classify b-/y-ions based on fragmentation characteristics will need to be trained on data similar to the instrument used for testing.

The peptide sequence itself can also result in variability in the spectrum, *e.g.,* different reaction pathways, different protonation motifs, and energy dependent rate constants [45]. Sequence sources of variability can also be modeled through machine learning but the high dimensionality of the problem causes approaches to suffer from high complexity and overfitting issues.

2.2 Peptide Sequencing Approaches

There are three general approaches to solving the peptide sequencing problem via MS/MS. Database methods, de novo sequencing, and tagging/hybrid approaches. These approaches are briefly outlined below.

Database Methods

Database methods are the most commonly used in industry and academia. These methods work by treating the experimental spectrum as a fingerprint and comparing it with theoretical spectra predicted for peptides present in a database. A database comparison attempts to find the theoretical spectrum in the database that most closely resembles the experimental spectrum. The resemblance measure is typically a cross correlation function or some other alignment score. The peptide sequence having the theoretical spectrum with the highest match score is considered to be the most likely candidate peptide to have produced the experimental spectrum. SEQUEST [11] and Mascot [33] are two established database search algorithms. These databases are generated by taking sequenced genomes, predicting expressed proteins from coding genes, simulating digestion of the protein, and computing the theoretical mass spectrum that would result from the fragmentation of the resulting peptide.

Despite the maturity of database methods they can only be effective when a peptide of interest was produced by an organism whose genome has been sequenced and is available in a database. This is especially problematic in the case of microbial samples. It has been estimated that only 1%-10% of microbes in the environment can be cultured, and thus their genomes have not been sequenced. There are also countless other microbes that have yet to be identified and it would likewise be difficult to analyze their novel peptides via MS/MS database methods. Even when a microbial genome has been sequenced it is possible that a peptide will exhibit post translational modifications. These modifications cannot be predicted from the genome and can complicate comparisons between the experimental spectrum and the theoretical spectrum.

De Novo Sequencing Overview

De novo peptide sequencing algorithms attempt to reconstruct the sequence using only the information contained in the experimental spectrum, without the aid of a database of theoretical spectra for comparison. De novo algorithms search the space of all possible peptides that are consistent with peaks in the experimental spectrum to find the peptide sequence that scores best. Modern approaches to solving this problem map the peaks in the spectrum to a secondary data structure that is then analyzed using dynamic programming. The two predominant structures used are the *mass array* [26, 29] and the *spectrum graph* [9, 4, 25, 16, 12]. The spectrum graph is the key computational technique behind de novo peptide sequencing[17]. In the spectrum graph approach the vertices in the graph correspond to peaks, edges correspond to mass differences consistent with the mass of an amino acid, and paths through the graph correspond to peptide sequences.

Since de novo algorithms do not rely on information in a database they are the only way to sequence the peptides of novel organisms, or the peptides of known organisms that have undergone modification. However, state of the art de novo algorithms still suffer from low accuracy when sequencing from low resolution instruments. In this dissertation we will describe several algorithmic approaches that increase the accuracy of de novo peptide sequencing.

Hybrid approaches

Frequently the peptide that produced a given MS/MS spectrum is not present in a protein sequence database. This occurs when the expressed proteins differ from proteins predicted from an organism's genome due to post-translational modifications, or when an organism's genome has not been sequenced. Despite the relatively low accuracy of state of the art de novo algorithms, they can be combined with database approaches to improve protein identification/sequencing results. To accomplish this, de novo algorithms predict short sequence *tags* that are used to filter a protein database[28, 41, 18, 3], perform homology based error-tolerant database searches[19, 37], validate database search results[44], construct spectral networks[2], or perform shotgun protein sequencing[1]. Tagging has been shown to speed up database searches by two orders of magnitude compared to conventional methods[14]. Homology based error-tolerant database approaches such as Spider[19] take advantage of the fact that de novo sequencing errors have different probabilities compared to evolutionary mutation probabilities, by aligning individual de novo sequences to a database and treating unaligned portions as either de novo sequencing errors, mutations, or post-translational modifications.

2.3 Sequencing Errors

Homeometric peptides, peptides with differing sequence but similar MS/MS spectra, are a prominent source of sequencing errors in low precision MS/MS. There is a 30% chance that an arbitrary peptide of length 10 will have at least one homeometric peptide[17]. Homeometric peptides can occur if we switch two sequential residues in a candidate peptide sequence, or if we switch between prefix and suffix vertices in the spectrum graph. This problem is generally solved with higher precision instruments, however no solution exists for low precision MS/MS. Another common source of sequencing errors is due to *isobaric* amino acids, where two or more amino acids have similar or identical mass. The amino acids Leucine (L) and Isoleucine (I) have the same mass (113.08 Da), and the amino acids Lysine (K) (128.095 Da) and Glutamine (Q) (128.059 Da) have similar mass, differing by only 0.036 Da. K and Q are indistinguishable in low precision MS/MS, and I and L are indistinguishable regardless of precision.

In chapter 4 we discuss a solution to these sequencing errors that uses amino acid usage–among other features–to rerank candidate peptides. In this reranking schema we leverage the existence of homologous sequences to give preference to one peptide spectrum match over another, where the peptides may be homeometric, or contain isobaric amino acids.

Chapter 3

Peak Selection

3.1 Peptide fragmentation

Let Σ be the alphabet of 20 amino acids. A peptide P consists of a sequence of amino acids (residues), $P = a_1 a_2 \cdots a_n$, where $a_i \in \Sigma$. When an amino acid is chained together with others as part of a (poly)peptide it is referred to as a *residue*. Note that in what follows the terms residue and amino acid are considered equivalent. The mass of an atom or molecule is represented by $\| \cdot \|$, *e.g.,* the mass of an amino acid is defined as $\|a_i\|$. The parent (or precursor) ion mass reported by the mass spectrometer for peptide P is denoted pI, and it's mass $\|pI\|$ is defined as

$$\|pI\| = \|H\| + \|P\| + \|H\| + \|OH\|$$

where $\|P\| = \sum_{i=1}^{n} \|a_i\|$. The first Hydrogen is attached to the N-terminal end of the peptide, the second Hydrogen is the proton from CID, and the Hydroxyl (OH) is attached to the C-terminal end of the peptide. The total residue mass can then be calculated by subtracting the mass of a three Hydrogens and an Oxygen (19.023 Da) from the parent ion.

$$\|P\| = \|pI\| - 19.023$$

Any errors in the reported parent ion mass can easily be corrected with available software, and publicly available datasets typically have already run a correction algorithm and report the correct mass.

When P is fragmented by MS/MS each prefix and suffix of the peptide will produce several of the fragment ion types mentioned previously. Since we are focusing on

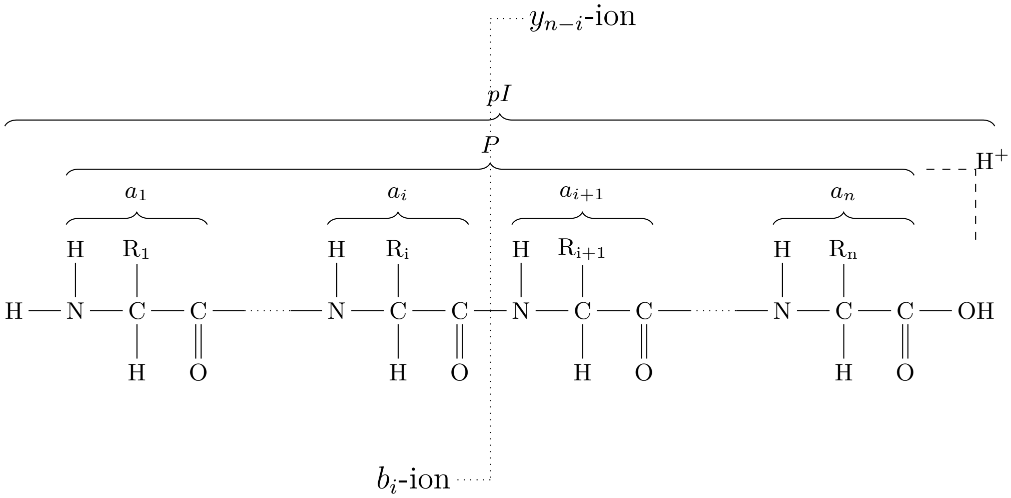

Figure 3.1: Breakdown of peptide mass components.

CID spectra we expect *b*-/*y*-ions to be the most abundant in the spectrum. These prefix and suffix ions in an ideally fragmented peptide will form two *ladders*. In the case of *b*-ions, the ladder refers to the peaks corresponding to the prefix ions observed sequentially in the spectrum (b_1-ion, b_2-ion, ..., b_n-ion) with each prefix offset from the previous by the mass of an amino acid. Likewise, for the *y*-ion ladder we expect to see suffix ions sequentially in the spectrum (y_1-ion, y_2-ion, ..., y_n-ion). These prefix and suffix ions are complementary such that the prefix b_i-ion and the y_{n-i} ions are complements and represent a cleavage between a_i and a_{i+1} in the peptide (Figure 3.1). We can use this knowledge to derive the peptide sequence. By concatenating the amino acids corresponding to the sequential mass differences in the *b*-ion ladder we construct the peptide sequence. Likewise, by concatenating the amino acids corresponding to the sequential mass differences in the *y*-ion ladder we construct the reverse peptide sequence.

Due to the nature of peptide fragmentation, the de novo peptide sequencing problem is that of identifying the subset of peaks in the spectrum that correspond to the *b*-/*y*-ion ladder, then computing the peptides that are most consistent with the ion ladder, and finally selecting the peptide that best "explains" the experimental spectrum. These three steps roughly correspond to the three main divisions in the QuasiNovo algorithm, *i.e.,* peak selection, candidate generation, and candidate scoring.

Peak Selection An ideal tandem mass spectrum would be free of noise and contain all possible prefix (N-terminal *b*-ion) and suffix (C-terminal *y*-ion) fragments, and only these fragments. In reality, mass spectra are far from ideal and contain a complex mixture of fragments and uninterpretable 'noise' peaks. Ions often produce two or three mass peaks due to isotopic carbon atoms contained in the ion. An ion may also produce peaks corresponding to neutral losses such as water or ammonia. Different ion types such as *a*-, *c*-, *x*-, or *z*-ions may be produced if the breakage doesn't occur at the amide bond. Internal fragments, which occur when an ion undergoes a second or third fragmentation, may also be present in the spectrum.

Since we prefer to sequence the peptide using a *b*-/*y*-ion ladder the spectrum is filtered to select the 'signal' peaks that likely correspond to *b*-/*y*-ions. A careful balance must be maintained between the precision and recall of peaks that are selected for further processing and candidate peptide generation. If too many peaks are selected the search space will be too large and the problem becomes intractable. If too few peaks are selected cleavage sites will be missed, the resulting candidate peptides will have large gaps, and sequencing results will be poor. For this reason pre-filtering of MS/MS spectra and accurate selection of peaks for peptide candidate generation is essential to any de novo peptide sequencing algorithm.

Peak selection is an important preprocessing step in de novo sequencing. As a practical matter, it is important that the number of peaks be reduced so that the candidate peptide search space is constrained. A reduction in the number of peaks used to create the spectrum graph makes it possible to process spectra faster. It also makes it possible to process longer peptides than would otherwise be impractical.

Quality of search space is as important as reduction of search space. It is critical that those peaks corresponding to *b*-/*y*-ions be identified so that the resulting candidate search space contains the correct peptide. The results presented below demonstrate that a staged neural network approach results in fewer peaks being selected. The resulting search space is smaller.

The general approach used by other prominent de novo peptide sequencing algorithms depends primarily on relative peak intensity. PepNovo+ uses a sliding window of width 56 across the spectrum and keeps any peaks that are in the top 3 when ranked by intensity[16]. ms2preproc uses the same sliding window approach, in addition to other intensity based methods.[35] MSNovo selects peaks by using a sliding window of width 100 and selects the top 6 peaks from each window[29]. PILOT keeps only the top 125 peaks of highest intensity in the spectrum[10]. pNovo selects the top 100 peaks by intensity[5].

In our experiments we found that selecting peaks based on relative intensity alone could miss a nontrivial portion of *b*-/*y*-ions. If the complex dynamics of peptide fragmentation–including relative peak intensity–can be modeled and incorporated into a predictive ion-type classifier, then the accuracy of peak selection will be superior to the accuracy of a peak classifier that uses peak intensity alone. We demonstrate that this superior approach can be implemented via a staged probabilistic neural network. A neural network approach was used because it allows us to construct a predictive model that does not require the complete understanding of the complex dynamics of peptide fragmentation. The Staged Neural Network (SNN) ion classifier described below selects peaks with higher precision and recall than other preprocessing and de novo peptides sequencing algorithms.

Increasing recall leads to better candidates in the candidate peptide search space. If recall is held fixed and the precision is increased, then the result will be a significantly smaller candidate peptide search space, without sacrificing the best candidate contained in the search space. Given the computational limits that all de novo algorithms face, low precision can render any de novo algorithm computationally impractical. Low recall will result in missing peaks, which in turn will result in large gaps in the spectrum graph. This in turn leads to an exploding combinatorial search as permutations of residues consistent with these large gaps must be considered. It is clear that a careful balance of improved precision and recall is important for peptide candidate generation.

3.2 Methods

Two datasets were used in this study. The datasets were limited to doubly charged peptides since this charge state is most common in MS/MS experiments. The first dataset (NIJ) is a comprehensive full factorial LC-MS/MS benchmark dataset[43] from the Nijmegen Proteomics Facility of Radbound University. NIJ consists of 59 LC-MS/MS analyses, in Mascot generic peak list format, of 50 protein samples extracted individually from Escherichia coli K12, yielding a total of 482 604 spectra. We then filtered the dataset for peptides of length 8 to 20. NIJ consists of 59 separate analyses ($\mathbf{D}_{\text{NIJ}} = \cup_{i=1}^{59} D_i$). The scans in each analysis set D_i were randomly divided into a training set (D_i^T) and an evaluation set (D_i^E). Each D_i^T was trained separately, and each D_i^E was classified using the classifier yielded by D_i^T. The classified scans in each D_i^E were then combined ($\mathbf{D}_{\text{NIJ}}^E = \cup_{i=1}^{59} D_i^E$) for calculating statistics. The same scans in $\mathbf{D}_{\text{NIJ}}^E$ were used to compute statistics for PepNovo+, pNovo, and ms2preproc. The results of this comparison are presented in Figure 3.5.

The second dataset (PNL) came from Pacific Northwest National Laboratory. PNL consists of 8 610 mass spectra from Salmonella Typhimurium,[1]. The dataset was filtered for peptides 8 to 24, and then randomly divided in half for 2-fold cross validation ($\mathbf{D}_{\text{PNL}} = D_1 \cup D_2$). First, D_1 was used for training ($D^T \leftarrow D_1$) and

[1] http://omics.pnl.gov/view/dataset_80292.html

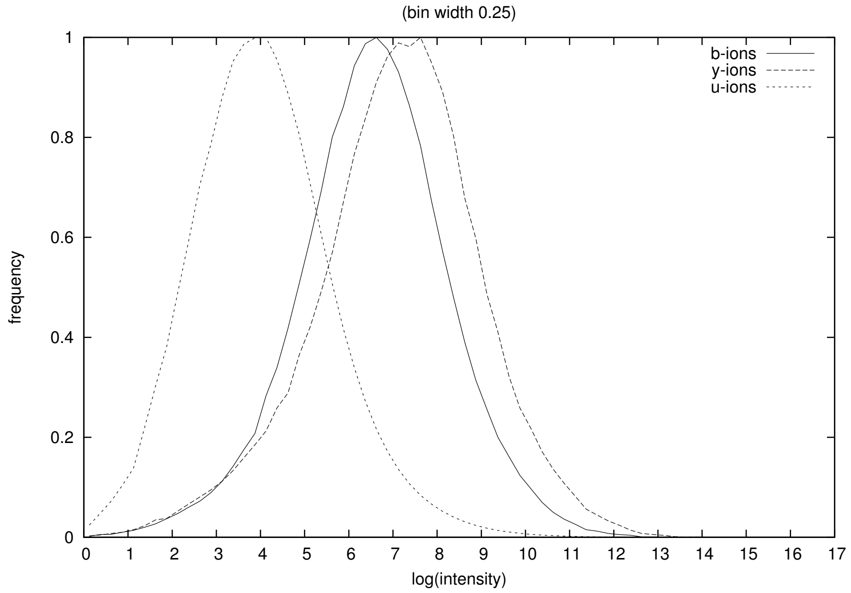

Figure 3.2: Distribution of peak intensity by ion type.

D_2 for classification/evaluation ($D^E \leftarrow D_2$), and then the reverse ($D^T \leftarrow D_2$ and $D^E \leftarrow D_1$). The results from each fold were averaged and are presented in Figure 3.6.

For each spectrum in the training dataset we first removed peaks with intensity below an experimentally derived threshold, in this case 50, which dramatically sped up the training of the neural network without sacrificing performance or accuracy. Figure 3.2 shows the effect of this threshold. By removing all peaks with an intensity below 50 (3.9 on the log scale), we remove half of the noise, but only a small percentage of *b*-/*y*-ions.

Before the neural network can be trained D_i^T must be transformed. Each peak in D_i^T is assigned its correct class label (target vector), either *b*-ion, *y*-ion, or *u*-ion (unknown ion), each of which is a binary vector of length three. A virtual spectrum is constructed based on known CID fragmentation [20, 45] of doubly charged peptides, giving the expected *b*-/*y*-ion masses for the peptide. A peak within 0.2 Da of the expected mass for a *b*-/*y*-ion is labeled accordingly and assumed to be ground truth.

Algorithm 1 Staged Neural Network Training and Classification

```
$net_1 \leftarrow train(D^{T_B}, D^{T_V})$
$D^T \leftarrow classify(D^T, net_1)$ {peaks in $D_T$ now have b-/y-/u-ion probability estimates}
$net_2 \leftarrow train(D^{T_B}, D^{T_V})$
$D^E \leftarrow classify(D^E, net_1)$
$D^E \leftarrow classify(D^E, net_2)$
```

For each peak in D_i a feature vector (pattern) is generated that will later be presented to the input layer of the neural network for training and classification. The features used are described in Table 3.1 and the following section. D_i^T is randomly divided again such that 95% of the spectra were used for backpropagation ($D_i^{T_B}$) and 5% of the spectra for validation (stopping criteria) ($D_i^{T_V}$). $D_i^{T_B}$ is then filtered so that there are an equal number of b, y, and u ions.

The training process of the neural network requires the use of an objective error function. In our implementation the output ($\mathbf{o}$) of the neural network represents an estimate of the posterior probability that the input pattern belongs to the respective class in the target vector ($\mathbf{t}$). When interpreting the outputs as probabilities it is appropriate to use the cross entropy error function[38].

$$\text{network error} = -\sum_{i=0}^{2} [\mathbf{t}_i ln(\mathbf{o}_i) + (1 - \mathbf{t}_i) ln(1 - \mathbf{o}_i)]$$

The neural network is trained on all of the patterns in the backpropagation training set numerous times (epochs) until the network performance no longer improves. This is determined by classifying the patterns in D^{T_V} after each epoch until the error on D^{T_V} begins to increase, at which point the training terminates.

In our classifier, two neural networks are used in succession for peak classification. We refer to this architecture as a staged neural network. Each network is trained in the manner described above except for differences in the feature vector used. The structure of each neural network consists of an input layer with as many nodes as features in the pattern, a single hidden layer with twice as many nodes as the input layer, and an output layer with three nodes corresponding to the three possible classes. A general formulation for training the neural networks is given in Algorithm 1. In the first neural network, net_1, the peak features are computed from data in the spectrum alone as described below and in Table 3.1. In the second neural network, net_2, the outputs from net_1 are leveraged as additional features in net_2. This is described in Table 3.1. In the net_2 input pattern, the complement ion feature is modified by replacing the feature value of the complementary peak with the maximum of the b-/y-ion probability estimates in the output from net_1 for the complementary peak. In the net_2 input pattern there are two additional features corresponding to flanking residues on the N and C terminal sides of the current peak. We use "current peak" to denote the peak for which the feature vector is being computed. The N-terminal flanking residue feature is computed by taking the maximum b-/y-ion probability (as estimated by net_1) of any peak with a mass offset from the current peak equivalent to the mass of an amino acid. The C-terminal flanking residue feature is computed similarly. The reasoning for these 'leveraged' features is that if the current peak has a complement or flanking peak with a high probability of being a $b-/y-$ion, then the current peak has increased probability of being a $b-/y-$ion itself. Our experiments show that leveraging the output from net_1 to train a second neural network in this way yields a higher recall than does classification with net_1 alone.

3.3 Description of features

The features listed in Table 3.1 capture known fragmentation characteristics and correlations between b-/y-ion peaks and other mass peaks produced by CID peptide fragmentation. In the following exposition, let pI denote the parent ion mass, and let $\mathbf{I} = \langle I_0, I_1, \ldots I_k \rangle$ be the MS/MS spectrum. For ion I_i in the spectrum, the m/z value is denoted x_i, and the intensity (or abundance) is denoted y_i.

The intensity feature ($\mathcal{F}_{intensity}$) is the normalized and discretized relative peak

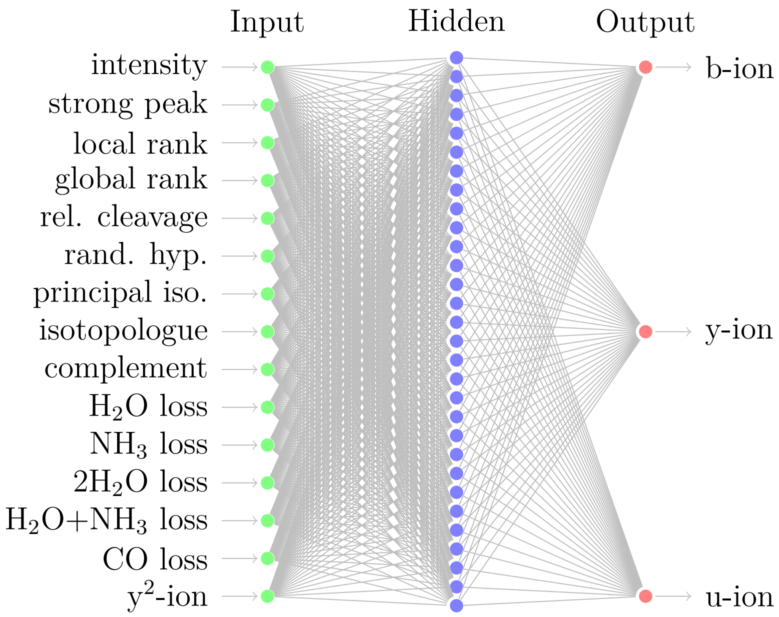

Figure 3.3: Topology of net_1

Table 3.1: Pattern features: N denotes a normalized value, D denotes a discretized value, B denotes a binary value, H denotes a histogram value, $\mathcal{N}$ denotes value sampled from the normal distribution, and P denotes a probability estimate. Each peak in the spectrum is classified by both neural networks in successive passes over the spectrum. net_2 features depend on the classification results of net_1.

net_1 pattern features		net_2 pattern features	
feature	value	feature	value
intensity	N, D	intensity	N, D
strong peak	B	strong peak	B
local intensity rank	N	local intensity rank	N
global intensity rank	N	global intensity rank	N
relative cleavage position	N, D	relative cleavage position	N, D
principal isotope	H	principal isotope	H
isotopologue	B	isotopologue	B
complement	$\mathcal{N}$	complement	P_{net_1}
H_2O neutral loss	N, D	H_2O neutral loss	N, D
NH_3 neutral loss	N, D	NH_3 neutral loss	N, D
H_2O-H_2O neutral loss	N, D	H_2O-H_2O neutral loss	N, D
H_2O-NH_3 neutral loss	N, D	H_2O-NH_3 neutral loss	N, D
CO neutral loss (*a*-ion)	N, D	CO neutral loss (*a*-ion)	N, D
		N-term flanking ion	P_{net_1}
		C-term flanking ion	P_{net_1}

intensity of the current peak I_i, the peak for which a feature vector is being created. Normalized intensities are computed by dividing each peak intensity by the maximum peak intensity in the spectrum. Given n discrete intensity bins, the normalized discretized feature is defined as

$$\mathcal{F}_{intensity}(I_i) = \lfloor n\,(y_i/y_{max}) \rfloor /n$$

where the y_{max} is the most intense peak in the spectrum, and y_i/y_{max} is the normalized intensity for I_i.

The strong peak feature is a binary value that indicates whether or not the current peak was selected as a ‘strong peak’ using a sliding window method; in this case the top three peaks by intensity were selected in a sliding window of width 56 Da.

The local and global intensity ranks give the normalized rank by intensity of the current peak within a local window, or globally. These first four peak intensity based features are informative due to the fact that b-/y-ions tend to be of higher abundance than other ion types in CID spectra.

The relative cleavage position is a categorical set of features defined as $\mathcal{F}_{position}(I_i)$. These categories reflect equally sized regions of the spectrum based on the mass of the current peak relative to the parent ion mass (pI). For example, if we assume the number of regions $n = 5$, the lowest mass peak in the spectrum would have the feature value $\mathcal{F}_{position}(I_0) = \langle 1, 0, 0, 0, 0\rangle$, and the highest mass peak would have the value $\mathcal{F}_{position}(I_k) = \langle 0, 0, 0, 0, 1\rangle$. These features capture the variation in peak intensity across the mass range of the instrument. Typically, peaks tend to be more intense near the center of the peptide and less intense or missing near the terminal ends. Fragmentation characteristics can also vary based on the relative cleavage position. The input layer of the neural network has a node corresponding to each of the categorical features, $c = 0, 1, \ldots, n-1$, which are assigned either 0 or 1 as follows:

$$\mathcal{F}_{position}(I_i)_c = \begin{cases} 1 & \text{if } \frac{c}{n} \leqslant x_i/pI < \frac{c+1}{n} \\ 0 & \text{otherwise} \end{cases}$$

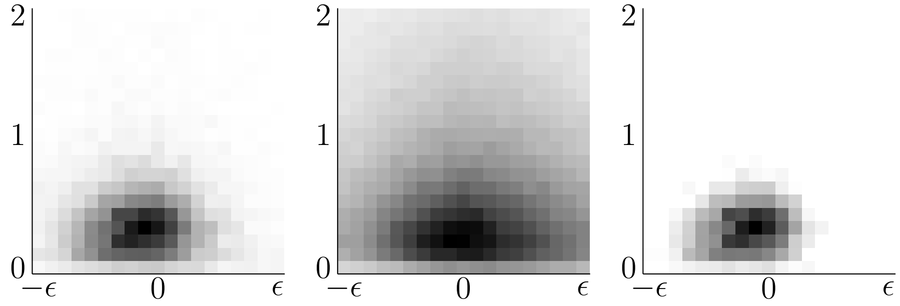

Figure 3.4: (a) The experimental mass offset (δ') and relative intensity (y') for the first isotopologue of a *b*-/*y*-ion. (b) The experimental mass offset and relative intensity for the first isotopologue of an unknown ion (not *b*-/*y*-ion). (c) The log-odds ratio between (a) and (b)

where x_i is the mass of I_i, and pI is the mass of the parent ion.

When a cleavage occurs between two amino acids, there are several other peaks that are observed with high frequency. These peaks correspond to isotopologues, neutral losses, doubly charged ions, and complementary ions. The remaining features use relative intensity and mass offsets to compute feature values. Given the current peak I_i, an expected offset δ, and a mass tolerance ϵ; the offset peak I_j is the maximum intensity peak in the range $[x_i + \delta - \epsilon, x_i + \delta + \epsilon]$. The experimental offset is then defined as $\delta' = x_i - x_j + \delta$. For a given offset peak I_j the relative intensity is defined as $y'_j = y_j / y_i$.

The principal isotope feature is taken from a two dimensional histogram that models the relative intensity and mass offset of the first isotopologue. This histogram, shown in Figure 3.4, was computed by summing the frequency of mass offset and relative intensity within bins of size 0.05 and 0.1, respectively, for *b*-/*y*-ions in the dataset. The presence of a lower intensity isotopologue at offset $\delta = 1$ serves as positive evidence that the current peak is a *b*-/*y*-ion. This can be demonstrated by building a histogram for peaks that are labeled *u*-ions, and then computing the log-odds ratio between these two distributions, as shown in Figure 3.4. The principal isotope feature value is sampled from this histogram based on the δ' and y' values of a candidate isotopologue of I_i. The isotopologue feature indicates the current peak is an isotopologue of the peak at offset -1 Da. If the current peak has been labeled as an isotopologue, then the isotopologue feature value will be 1, and 0 otherwise. Adding this feature increases precision without affecting recall. This is due to the observation that, if a peak's isotopologue feature is 1, it will most likely be classified as a *u*-ion by the neural network.

If the current peak is a *b*-ion then we will often see the complimentary *y*-ion peak, and likewise for the converse. It is tempting to use a 2D histogram to model this feature. However performance degrades if we consider relative intensity since we do not want to penalize a candidate complementary peak for having a non-average relative intensity. The complement feature in the net_1 pattern is taken from a normal distribution centered at the offset where a complementary ion is expected to be if the current peak is a *b*-/*y*-ion. By constructing a one dimensional histogram of the complementary ion mass offset, it was observed that the offset frequency is approximately Gaussian and can be modeled as $X \sim \mathcal{N}(0, 0.1)$, where the 0 mean is centered around the expected complementary ion mass $x_j = pI - x_i + 1$. The feature value is then defined as

$$\mathcal{F}_{complement}(I_i) = X(\delta^c)$$

where δ^c is the difference between the expected and the experimental complementary ion mass. In the case of the net_2 pattern the complement feature gives the maximum *b*-/*y*-ion probability estimate using net_1 of any peak found at the expected complement mass offset.

The H_2O, NH_3, H_2O-H_2O, H_2O-NH_3, and CO neutral loss features are computed by summing the relative intensity and the Gaussian estimate of the offset frequency, as described above. For example, given the neutral loss peak I_j at the offset $\delta = -18.015$, the feature value is defined as

$$\mathcal{F}_{-\mathrm{H_2O}}(I_i) = y'_j + X(\delta')$$

The N-term and C-term flanking ion features are the maximum *b*-/*y*-ion probability estimates using net_1 for any peaks that are found at a mass offset from the current peak corresponding to the mass of a single amino acid. If the current peak is indeed a *b*-/*y*-ion then we expect it to be part of an ion ladder, and thus we expect to find other peaks that are likely *b*-/*y*-ions at mass offsets equivalent to the mass of an amino acid. This feature value is the flanking peak's net_1 probability estimate, not the mass difference, and therefore does not capture any sequence information.

3.4 Experimental Results

Results comparing precision and recall are shown in Figures 3.5 and 3.6. We compared the performance of the staged neural network (SNN) peak selection with two other prominent de novo peptide sequencing algorithms. The window method selects peaks by choosing the 3 most intense peaks in a window of width 56 Da. We used ms2preproc to implement this method, which Frank describes in the original PepNovo publication[16]. Peak selection in PepNovo+ was subsequently improved. As shown in this figure, the actual performance of PepNovo+ is substantially better with respect to recall than the window method. The actual performance of PepNovo+ was determined by modifying the source code to output the peaks from the raw spectrum that are used to construct the spectrum graph.

Note that the SNN precision is consistently greater than that of PepNovo+. The number of peaks selected has a direct impact on the size of the search space. This effect can be seen when the search space of candidate peptide sequences is generated using the peaks selected by the two algorithms (Figure 3.7). We implemented a basic dynamic programming approach as described in Lu and Chen [25] to generate candidate peptides using the peaks selected by PepNovo+ and the SNN. Keep in mind that programs such as PepNovo+ use much more sophisticated approaches to generate candidates from the spectrum graph. This allows them to avoid an exhaustive search

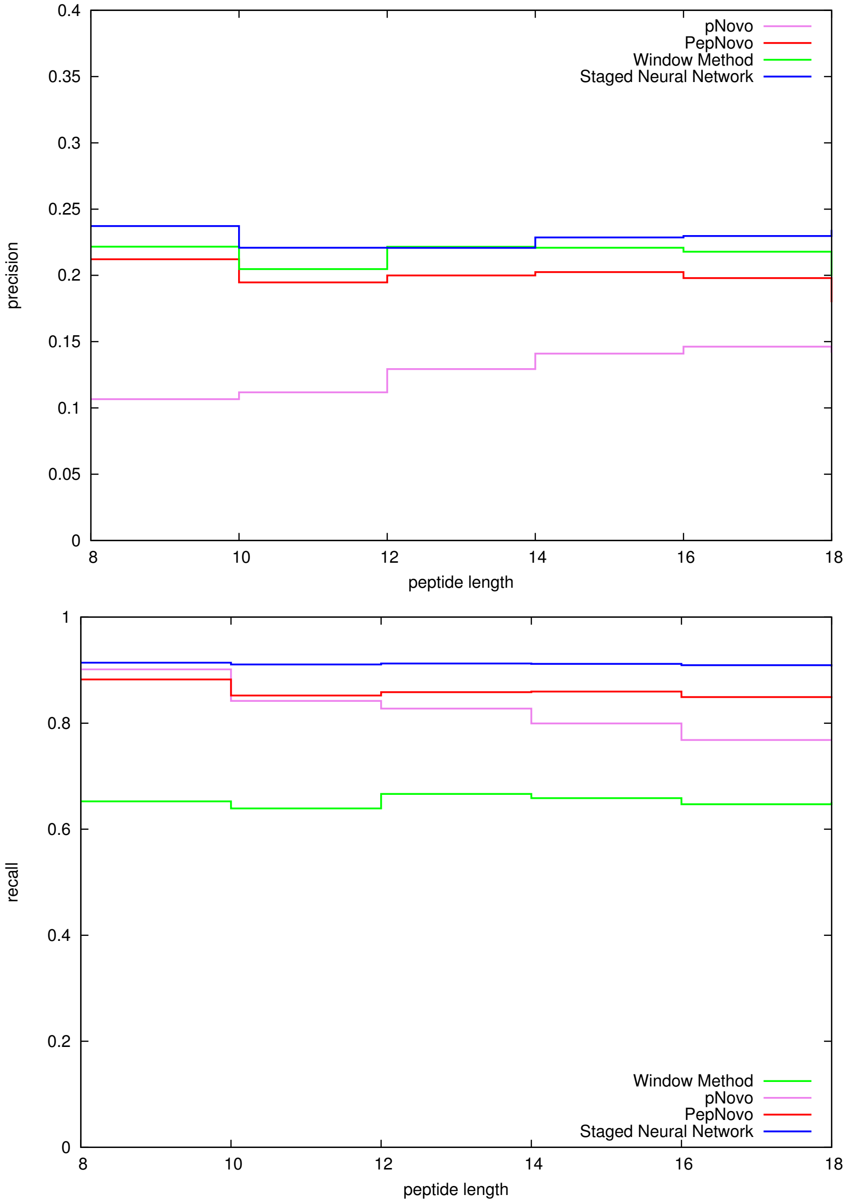

Figure 3.5: Results for $\mathbf{D}_{NIJ}$ comparing the precision and recall for b-/y-ion selection across varying peptide length. Neural network approach compared to PepNovo+, pNovo, and the ms2preproc window method

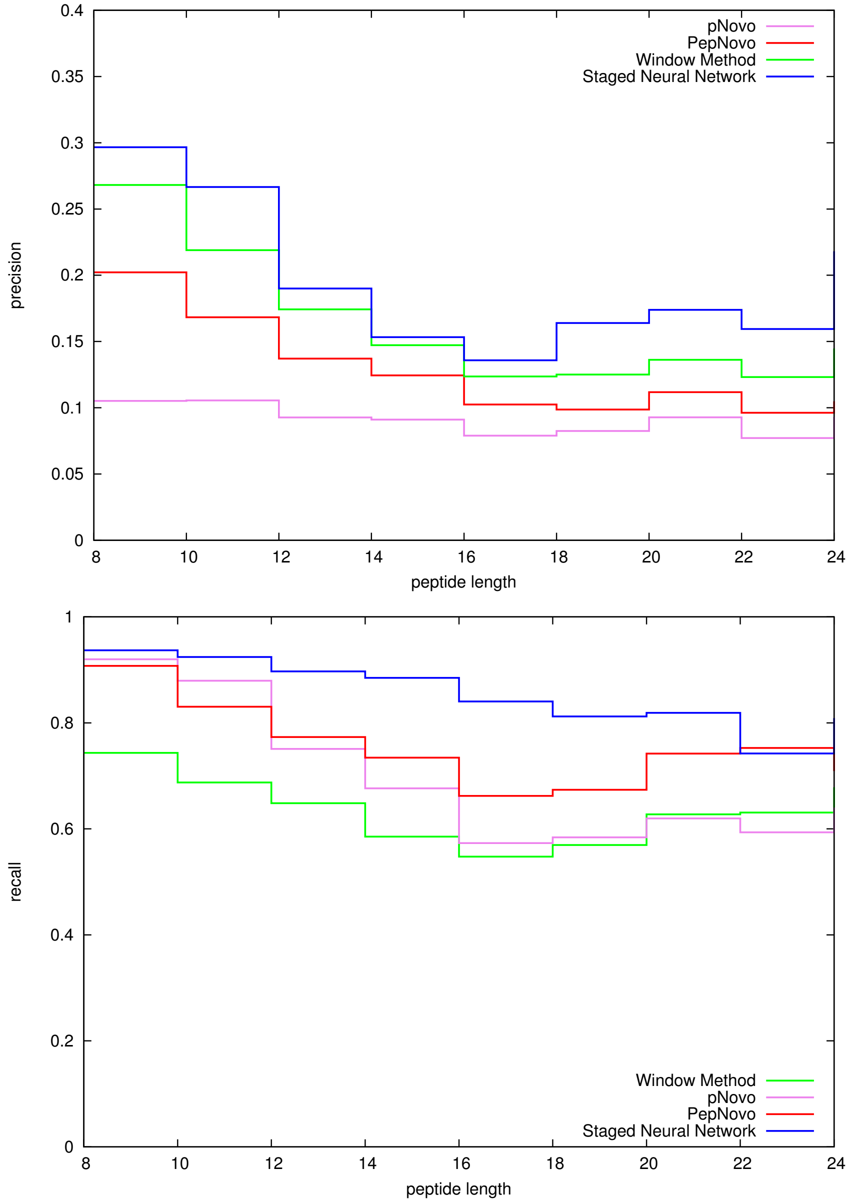

Figure 3.6: Results for $\mathbf{D}_{PNL}$ comparing the precision and recall for b-/y-ion selection across varying peptide length. Neural network approach compared to PepNovo+, pNovo, and the ms2preproc window method

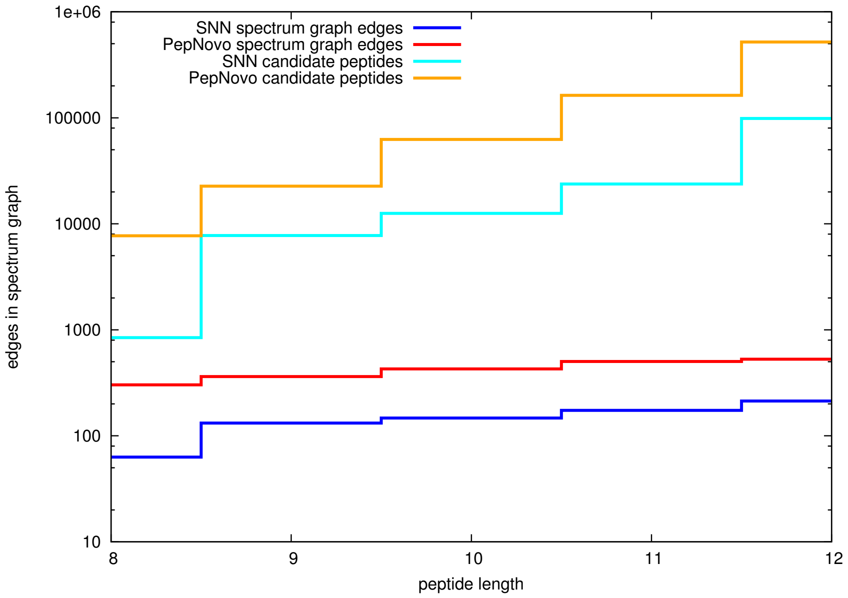

Figure 3.7: Comparison of the median number of edges in the spectrum graph (bottom pair) and the median number of candidate peptides generated from the spectrum graph (top pair). Note that the y scale is logarithmic and the relationship between the number of edges in the spectrum graph and the number of candidate peptides is exponential.

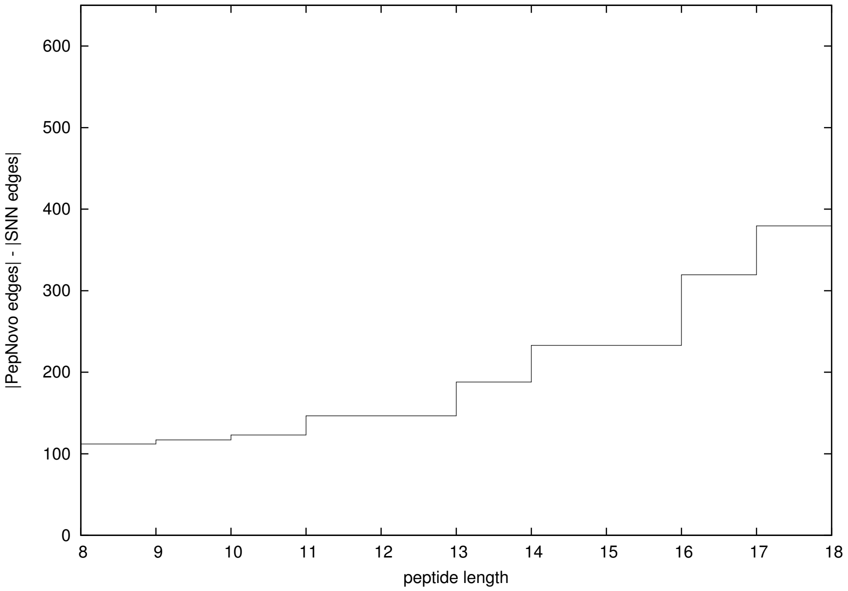

Figure 3.8: SNN net reduction in spectrum graph edges compared to PepNovo+. These data points were produced by subtracting the median number of edges in the SNN spectrum graph from the median number of edges in the PepNovo+ spectrum graph for peptide length bins of width 2.

of the implicit search space. The size of the candidate peptide search space generated by the de novo algorithm using PepNovo+'s peak selection is larger than the size of the candidate peptide search space generated by the de novo algorithm using SNN peak selection. The difference in the size of the candidate peptide search space is shown in Figure 3.7 for peptides of length 8 to 12. The size of the candidate peptide search space is exponentially proportional to the number of edges in the spectrum graph. Consequently we use this as a measure of search space to extend our results to longer peptides by comparing the number of edges in the spectrum graphs produced by each peak selection algorithm (Figure 3.8) without having to exhaustively enumerate the candidate peptides.

It should be noted that on balance the b-/y-ion recall of the SNN peak selection is greater for all peptide lengths included in our experiments. Thus on balance we can expect that the top scoring candidate peptides that would be generated by these spectrum graphs will be of higher quality. When we compare the number of edges in the spectrum graphs we find that the number of edges generated by PepNovo+'s peak selection contain on average approximately 300 more edges than the corresponding SNN spectrum graph. The difference in edges increases as peptide length increases resulting in smaller search spaces for larger peptides that can often be too large to explore.

Chapter 4

Candidate Generation

In the candidate generation step the peaks selected using the methods described in the previous chapter are used to generate a set of candidate peptides that could have produced the spectrum, *i.e.,* a set of candidate peptide-spectrum matches. This step is the focus of most de novo peptide sequencing algorithms, while peak selection and candidate scoring are generally considered to be preprocessing and postprocessing steps, respectively. It is during candidate generation that the effects of inadequate peak selection become problematic and the space and time complexity of the problem becomes apparent.

The goal of a de novo peptide sequencing algorithm is to find the peptide that most likely produced the experimental spectrum. A *global* approach to candidate generation would be to construct the search space of all peptides that sum to the total residue mass, and then to select the peptide that scores best when compared to the experimental spectrum. However a search space constructed this way is extremely large and impractical to enumerate. As a result, de novo algorithms must construct a search space in a bottom-up *local* fashion. In our approach a *spectrum graph* [4] is used to construct the search space. A spectrum graph is a directed acyclic graph. Vertices in the graph represent prefix fragment masses situated along a number line. Typically, additional vertices are generated to allow each peak to be interpreted as a *b*-ion regardless of its actual ion type, and vertices are subsequently merged within some mass tolerance. Edges connect vertices that have a mass difference equivalent to the mass of an amino acid, and the edge is labeled with that amino acid. Edges may also connect vertices with mass differences corresponding to pairs of amino acids–in the case of a missing-peak interpretation–or mass differences corresponding to post-translationally modified amino acids. Thus, a de novo peptide sequencing algorithm using the spectrum graph approach is a special case of graph search where the peptide sequence is generated by finding the highest scoring *antisymmetric* path in the graph and concatenating the edge labels along the path. Antisymmetric means that a given peak can only be used once in a path as either a b-ion or y-ion interpretation of the original peak, but not both.

4.1 The Spectrum Graph

The construction of a spectrum graph is based on the work of previous authors [9, 4, 25]. Given an unknown peptide P with total residue mass $\|P\|$, and k fragment ions $I_1, \ldots, I_k$ with masses $\|I_1\|, \ldots, \|I_k\|$ we construct a spectrum graph $G_S = (V_S, E_S)$ as follows. Let $m = 2k + 1$.

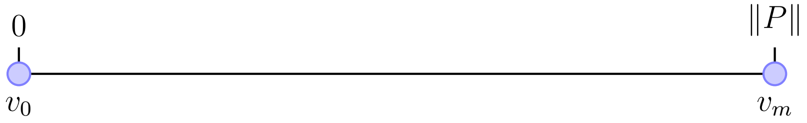

Figure 4.1: Spectrum graph initialization.

On a number line we create vertices v_0 and v_m representing zero mass and the total residue mass (nominal precursor ion mass) $\|P\|$ respectively. The total residue mass is equivalent to the precursor ion mass minus the masses of water and a Hydrogen atom.

For each fragment ion I_i we are unsure whether or not it is a b-ion or a y-ion so we must interpret the ion as both ion types and create a pair of vertices. The objective is to take each interpretation of each fragment ion and create a vertex corresponding to the nominal b-ion on the number line. If a b-ion is missing but the complementary y-ion is present in the spectrum we will generate the missing b-ion. This will allow us to sequence the peptide from the N-terminus to the C-terminus. If an ion is in fact a b-ion, then we can compute the nominal b-ion mass (total residue mass of a prefix ion) $\|I_i\|_b$ by subtracting the mass of a Hydrogen.

$$\|I_i\|_b = \|I_i\| - \|H\|$$

If an ion is in fact a y-ion, then we can compute the nominal y-ion mass $\|I_i\|_y$ by subtracting the mass of two Hydrogen and one Hydroxyl.

$$\|I_i\|_y = \|I_i\| - (\|H\| + \|OH\| + \|H\|)$$

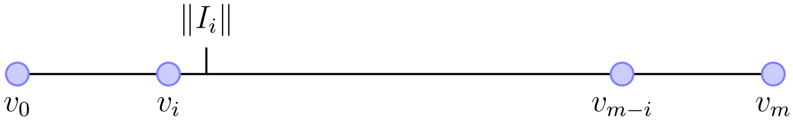

Figure 4.2: Dual interpretation of fragment ion.

This leads us to the following two interpretations of an unknown ion fragment shown in Figure 4.2. (1) We interpret the ion as a b-ion by adding vertex v_i to the number line at mass $\|I_i\|_b$. (2) We interpret the ion as a y-ion and create the complementary nominal b-ion by adding vertex v_{m-i} to the number line at mass $\|P\| - \|I_i\|_y$. If I_i is indeed a b-/y-ion then only one of these interpretations can be correct.

If a peak exists in the experimental spectrum for each ion type for a given fragmentation event, then the dual interpretation of the ion as described above will result in two vertices for the two interpretations of the peak, and there will be two such pairs–one pair corresponding to the actual b-ion and another pair corresponding to the actual y-ion. In other words, if a peak for each expected ion type exists in the experimental spectrum, then the dual interpretation of the peaks will result in four vertices on the number line as two overlapping pairs of vertices. For example, the peptide GEEK (Figure 4.3) fragments between the amino acids EE, resulting in a pair

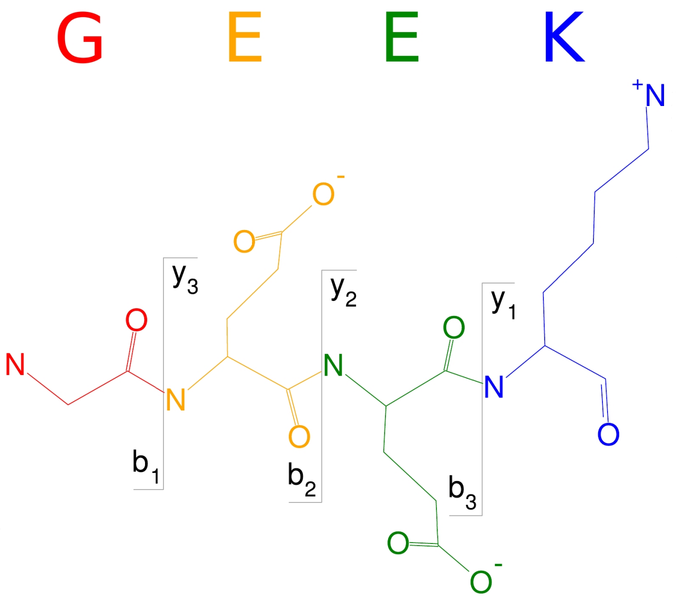

Figure 4.3: Example peptide showing b-/y-ions for each possible cleavage event.

of peeks at 187 Da and 276 Da, corresponding to b_2-ion GE and y_2-ion EK, respectively. Since we do not know *a priori* which is a b-ion and which is a y-ion, we create vertices for each interpretation. The peak at 187 Da yields vertices at 186 Da and 275 Da, and the peak 276 Da yields vertices at 275 Da and 186 Da. The two pairs of overlapping vertices are then merged, and of the resulting two vertices only one of them corresponds to a correct b-ion interpretation.

4.2 Vertex Scoring Function

The vertex scoring function for a given vertex is derived from its respective ion (I) in the experimental spectrum. A good vertex scoring function will typically estimate the probability that a vertex/edge in the graph is part of the b-/y-ion ladder. This can be accomplished through a correlation function based on known fragmentation patterns. In our approach we use the log-odds of the staged neural network probability estimates described in the previous chapter as our vertex scoring function during candidate generation. Assume we are creating vertices v_i and v_{m-i} as shown in Figure 4.2

where v_i is created for the b-ion interpretation of ion I_i and v_{m-i} is created for the y-ion interpretation of I_i. The vertex score for each vertex is given below.

$$f_{SNN}(v_i) = ln\left(\frac{p(I_i = b\text{-ion})}{p(I = u\text{-ion})}\right), \quad f_{SNN}(v_{m-i}) = ln\left(\frac{p(I_i = y\text{-ion})}{p(I = u\text{-ion})}\right) \tag{4.1}$$

Since the staged neural network estimates the probability that a given peak is either a b, y, or unknown ion, this probability estimate is a logical and effective method for scoring paths in the spectrum graph.

4.3 Graph Edges

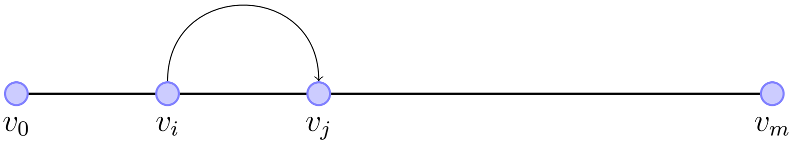

Figure 4.4: Edge connecting v_i and v_j that differ by the mass of an amino acid.

If the mass difference between two vertices v_i and v_j is equivalent to the mass of an amino acid, or pair of amino acid in the case of a missing peak, then a directed edge connects v_i and v_j from low to high mass. The spectrum graph is then a directed acyclic graph with vertices situated along the number line.

When we add an edge to the spectrum graph that corresponds to a pair of residues we must verify that the edge is not redundant, *e.g.*, Figure 4.5. In the case of a redundant edge we remove the higher mass pair of residues in favor of the *primary edges* that include along their path the vertex corresponding to the presumed missed cleavage in the edge corresponding to the pair of residues. However we must be careful to only remove edges that are truly redundant. In Figure 4.6 we have the masses $\|\mathsf{A}\| = 71.0\,\text{Da}$, $\|\mathsf{S}\| = 87.0\,\text{Da}$, $\|\mathsf{AS}\| = \|\mathsf{GT}\| = 158.1\,\text{Da}$. Since $\|\mathsf{AS}\|$ is not unique we cannot remove its edge. It is possible that v_j is a random noise peak and an interpreting it as a primary edge would cause us to eliminate the possibility that

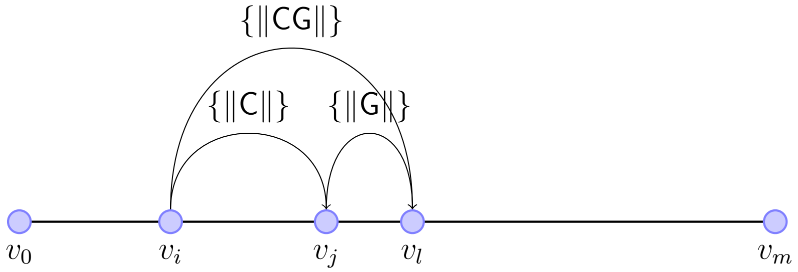

Figure 4.5: $v_i \to v_l$ is redundant; edge can be discarded.

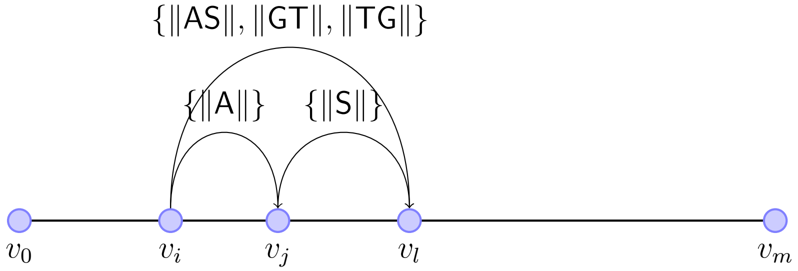

Figure 4.6: $v_i \to v_l$ is not redundant; edge must be kept.

the mass difference between v_i and v_l actually corresponds to the pair GT with a missing peak between G and T. Therefore we cannot remove $v_i \to v_l$.

4.4 Suboptimal Solutions

Once the spectrum graph is constructed we wish to generate candidate peptides by finding high scoring paths from v_0 to v_m. A *feasible path* (or solution) for the spectrum graph is a path from v_0 to v_m that uses each ion from the spectrum at most once. That is, for a given pair of vertices that result from the dual interpretation of a peak, only one of those vertices can be used in the feasible path. Since vertex scores are derived from ions, the following holds for a given vertex v that is constructed from ion I.

$$f_{SNN}(v) = f_{SNN}(I)$$

A suboptimal solution is defined as follows: Assume that P is the top scoring path for the spectrum graph, which can be found via depth-first-search. Let $f_{max} = f(P)$.

Given a ratio α such that $0 < \alpha \leq 1$, if a feasible path Q satisfies $f(Q) \geq \alpha \cdot f_{max}$, then Q is a suboptimal solution for the spectrum graph. Therefore, the suboptimal de novo peptide sequencing problem is to find all α-suboptimal feasible paths given a spectrum graph. In our implementation we used $f_{SNN}(v)$ as the vertex scoring function. The details for how to implement the α-suboptimal dynamic programming algorithm are given in detail in Lu's publication[25]. The standard algorithm was modified to map paths (solutions) to larger sets of candidate peptides, which are then scored using a novel scoring function described in the next chapter.

Chapter 5

Scoring Candidate Peptides

The peptide sequencing problem is defined as follows: Given a spectrum graph and a vertex scoring function find a maximum scoring path from v_0 to v_m. Without a good scoring schema random noise in the spectrum will generate numerous false interpretations of vertices and edges in the spectrum graph. These spurious vertices will dramatically increase the number of paths in the spectrum graph, and therefore the number of candidate peptides that correspond to those paths. The objective of a good vertex scoring function is to distinguish vertices that correspond to b-/y-ions from vertices corresponding to noise or other ion types, and thus assign the highest scoring path to the edges that corresponds to the actual peptide. The vertex scoring function we use during candidate generation was described in the previous chapter, and is given in equation 4.1.

During the candidate generation step the spectrum graph yields a set of candidate peptide-spectrum matches. Naturally, we now turn to the task of scoring these candidate peptides. We have implemented a peptide-spectrum match (PSM) candidate scoring function $f_{PSM}(\cdot)$ that combines the information content of an *amino acid usage* (AAU) model, the staged neural network, and an *edge frequency score.*

These three disparate sources of information represent three distinct scoring strategies applied to candidate peptides: (1) The staged neural network model estimates the probability that a fragment ion corresponds to a b-/y-ion by classifying the ion based on its fragmentation pattern and parameters relevant to the ion's fragmentation. In other words, it estimates the probability that an ion is a prefix or suffix ion

$$\begin{array}{ccc|ccc|c|cc} \cline{4-7} R_1 & R_2 & R_3 & R_4 & R_5 & R_6 & R_7 & R_8 & R_9 \\ \cline{4-7} \multicolumn{3}{c}{} & \multicolumn{1}{c}{R_{n-L}} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{R_{n-1}} & \multicolumn{1}{c}{R_n} & \multicolumn{2}{c}{} \end{array}$$

Figure 5.1: Conditional probability of residue $n = 7$ with tuple length $L = 4$; *i.e.,* $\Pr(R_7|R_4R_5R_6)$

based on information contained in the spectrum alone. (2) The amino acid usage distribution estimates the probability that the candidate peptide represents a probable sequence of amino acids given the amino acid usage distribution. Each residue corresponds to an edge in the spectrum graph ($v_{i,j}$), which corresponds to a mass difference between two peaks in the experimental spectrum. (3) The edge frequency score estimates the probability of a given edge (mass difference between two peaks) in a given spectrum. The latter two scoring strategies are discussed below.

5.1 Amino Acid Usage Score

An amino acid usage (AAU) distribution is a univariate frequency distribution that tabulates conditional probabilities derived from protein sequence data. It captures the *mutual information* present in adjacent residues in the protein sequences. There are three possible causes for mutual information (or statistical dependence) in protein sequence. First, evolutionarily related proteins have similar sequences due to shared ancestry. This similarity is called *homology* and it is captured by mutual information. Second, organisms with similar *GC content* will have similar amino acid composition [32, 39] that influences the AAU distribution. Third, there is some evidence that α-helix secondary structure constraints weakly bias amino acid usage [24].

The AAU frequency distribution can be computed by conditioning on single amino acids or tuples of length $L > 1$. A conditional probability for a tuple of length L is defined as $\Pr(R_n|R_{n-L}^{n-1})$, shown in Figure 5.1. Longer tuple lengths ($L \gtrsim 5$) will yield AAU distributions that are sparse and driven by sequence homology. It is difficult to collect enough peptide data to adequately populate a larger table. Even in the case of tuples of length 6, not all combinations of length 6 are observed so it is necessary to initialize those entries to some small epsilon value. The number of unique tuples is exponential in the length of the tuple, *i.e.*, 20^L (since there are 20 amino acids). Thus anything larger than 6 becomes unmanageable. Shorter tuple lengths (di-/tri-peptide tuples) will yield AAU distributions that model the influence of GC content, any known or unknown secondary structure bias, and weak sequence homology. We will refer to models of long and short tuple length as *strong* and *weak* homology models respectively. If the source protein is expected to have strong sequence homology then the AAU component of the scoring function should use longer tuples and an AAU distribution with the appropriate *taxonomic resolution.* If the source protein is not expected to have weak sequence homology then the scoring function should use shorter tuples and an AAU distribution with the appropriate GC content. It is not yet clear how the determination of strong or weak homology should be made in a high throughput setting. It will likely require that additional information about the source organism be available, such as 16S rRNA analysis of the organism prior to MS/MS analysis.

These distributions were created by selecting a number of proteomes from which to compile a composite amino acid distribution. (In practice, we started with translations from genome sequences.) The proteins from the selected proteomes were processed in the following manner. First, we chose a tuple length L. We then tabulated the frequency of occurrence of each tuple using a sliding window of length L. Let $< R_1R_2 \ldots R_n >$ be a contiguous sequence of n amino acids. There are $n - L + 1$ tuples of length L in this sequence: $< R_1R_2 \ldots R_L >$, $< R_2R_3 \ldots R_{L+1} >$, ..., $< R_{n-L+1}R_{n-L+2} \ldots R_n >$. Finally, the frequencies are then normalized to give the probability of each tuple in the composite set of peptides. From these 6-tuples we derive conditional probabilities of the form $\Pr(R_n|R_{n-L}^{n-1})$

Computing the AAU score for a given residue is straightforward:

$$f_{AAU}(R_i) = \begin{cases} Pr(R_i) & \text{if } i = 0 \\ Pr(R_i | R_{i-j}^{i-1}), \quad j = \max(1, i - L) & \text{otherwise} \end{cases} \tag{5.1}$$

where R_i is a residue in the candidate peptide, and L is the tuple length.

We have already demonstrated the effectiveness of amino acid usage in ranking candidate peptides in a previous conference publication [36], which is discussed in the following chapter.

5.2 Edge Frequency Score

The edge frequency score, $f_{EF}(\cdot)$, is is based on the realization that an edge having a specific mass difference occurs frequently outside the context of the feasible path for a candidate peptide. Recall that an edge connects two vertices that represent a pair of cleavages N- or C-terminal to one or more consecutive residues in a candidate peptide, A feasible path begins at the vertex corresponding to zero mass, and terminates at the vertex corresponding to the parent ion mass. Each edge along the feasible path connects two peaks with a mass difference equivalent to the mass of an amino acid (or pair of amino acids), and the usage of each vertex along the path is antisymmetric (as described in the previous chapter).

Let $v_{i,j}$ be an edge in the spectrum graph that has mass difference $\delta_{i,j}$ such that $\delta_{i,j} = \|a\| = \|I_j\| - \|I_i\|$ and $a \in \Sigma$. If we assume that $v_{i,j}$ corresponds to a true edge in the spectrum graph, i.e., v_i and v_j are both true b-ions in the spectrum, then the residue a is part of the correct candidate peptide. We can then assume that pairs of peaks with the mass difference $\delta_{i,j}$ will be overrepresented in the spectrum, when compared to pairs of peaks with the mass difference of a residue that is not part of the correct candidate peptide. And so, edges having mass difference $\delta_{i,j}$ will occur more frequently in the spectrum graph. This effect is expected due to neutral losses, internal fragment ions, and to a lesser extent, isotopic peaks.

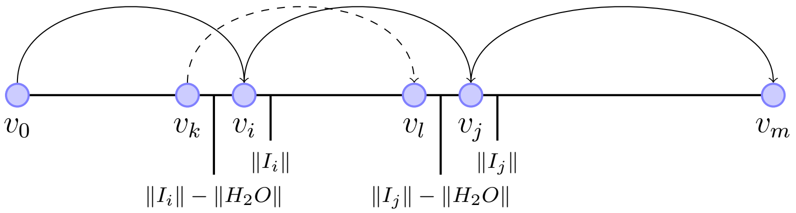

Figure 5.2: The solid line corresponds to a feasible path through the spectrum, and the dashed line corresponds to a neutral loss edge. The mass difference is the same for edges $v_{i,j}$ and $v_{k,l}$, but only $v_{i,j}$ is along a feasible path. The presence of edge $v_{k,l}$ serves as positive evidence that $v_{i,j}$ corresponds to a true residue in the candidate peptide since the neutral loss of water is common for b-ions. Note that this figure assumes that each vertex is created from a b-ion interpretation of the ions as described in the previous chapter.

Neutral losses are common in MS/MS data. For example, if we have an edge corresponding to $\|I_j\| - \|I_i\|$ in the spectrum, it is likely that there will be an edge corresponding to $(\|I_j\| - \|H_2O\|) - (\|I_i\| - \|H_2O\|)$ in the spectrum graph. This relationship is shown in Figure 5.2. While the mass difference is obviously equivalent, the position of the resulting vertex in the spectrum graph differ by $\|H_2O\|$. Since the loss of water is common for b-ions, the presence of the edge corresponding to the loss of water serves as evidence that the mass difference is due to a true residue in the candidate peptide. Thus, a candidate peptide containing that amino acid should receive a slight boost in its score.

Internal fragment ions occur when an ion fragments more than once. Consider the following example based on the peptide AFDQIDNAPEEK. Assume we have created vertices for cleavages between AP (v_i) and PE (v_j), and we have created the edge corresponding to residue P ($v_{i,j}$). The vertex v_i corresponds to the b-ion for AFDQIDNA, and v_j corresponds to the b-ion for AFDQIDNAP. If secondary fragmentations occurs, say, between FD, then DQIDNA and DQIDNAP will be secondary *internal* fragment ions in the spectrum with the respective vertices v_k and v_l created in the spectrum graph, and the edge $v_{k,l}$ corresponding to residue P. The edge $v_{k,l}$ is not part of the correct feasible path for the peptide, but it does correspond to a true residue in the

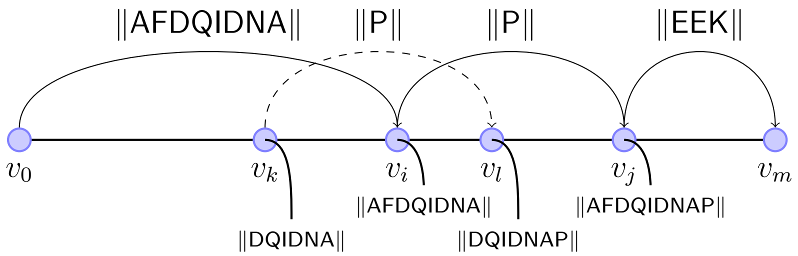

Figure 5.3: The solid line corresponds to a feasible path through the spectrum, and the dashed line corresponds to an internal fragment ion edge. The mass difference is the same for edges $v_{i,j}$ and $v_{k,l}$, but only $v_{i,j}$ is along a feasible path. The presence of edge $v_{k,l}$ serves as positive evidence that $v_{i,j}$ corresponds to a true residue in the candidate peptide since internal fragment ions are common. Note that this figure assumes that each vertex is created from a b-ion interpretation of the ions as described in the previous chapter. Also, note that for the sake of simplicity a single edge is shown for the $v_{0,i}$ prefix ion ‖AFDQIDNA‖ and the $v_{j,m}$ suffix ion ‖EEK‖, which is not permitted in the construction of a spectrum graph due to the large mass difference.

candidate peptide. Just as above in the case of neutral loss edges, this edge serves as additional evidence that P is part of the correct candidate peptide sequence. Figure 5.3 visualized this example.

The edge frequencies used in the scoring function are computed for each spectrum individually. For each spectrum we build a hash table, T_{EF}, which relates the masses of all *tags* (up to pairs of residues) to their edge frequency score. For clarity, we will hereafter use the term *tag*, or the symbol R^+ to refer to one or more amino acids. During the construction of the spectrum graph we use T_{EF} to count the frequency of each edge (mass difference). For each edge in the spectrum graph, the value in T_{EF} with a matching mass ($\pm 0.25\,\mathrm{Da}$) has its count incremented accordingly. These frequencies are then normalized by dividing each by the maximum edge frequency for the given spectrum. Thus, the edge frequency score for a given edge is the normalized frequency of that edge's mass difference in the spectrum. The score $f_{EF}(R^+)$ is retrieved from T_{EF}, which maps mass differences to normalized frequencies.

$$f_{EF}(R^+) = T_{EF}(\|R^+\|)$$

5.3 SNN Score

We have already demonstrated the effectiveness of a neural network for initial peak selection. The neural network described in Chapter 3 is a predictive model that doesn't require the complete understanding of the complex dynamics of peptide fragmentation, but models the fragmentation nevertheless. The neural network outputs a probability estimate that a peak is a *b*-/*y*-ion, which can be treated as the fragmentation model component of our vertex scoring function, $f_{SNN}(\cdot)$. This implementation is straightforward and follows the modular conception of the vertex scoring function introduced by previous authors [9, 16]. Since we are scoring candidate peptides our scoring function operates on the individual residues of a peptide. The appearance of a peak in a spectrum is interpreted as a pairwise cleavage between two sequential residues in a peptide. The neural network score for each peak, which is computed by 4.1, must then be mapped to individual residues in order to score candidate peptides. This is not always possible since, in the case of a missing peak in the *b*-/*y*-ion ladder, a vertex score may refer to more than one residue in the peptide. An edge $v_{i,j}$ in the spectrum graph between two vertices maps to a tag in a candidate peptide, and so the neural network score for a tag is equivalent to the neural network score for the respective edge in the spectrum graph, which is equivalent to the neural network score for the vertex with the higher mass (v_j). This relationship is shown below and in Figure 5.4.

$$f_{SNN}(R^{+}_{i,j}) = f_{SNN}(v_{i,j}) = f_{SNN}(v_j)$$

It is important to note that there is a one-to-many relationship between edges and tags, as there may be numerous combinations of amino acids that sum to the same mass. In other words, an edge may map to many tags, and each tag may have be composed of one or more residues. Each combination of amino acids that make up a unique tag for a given edge will have the same neural network score since this score

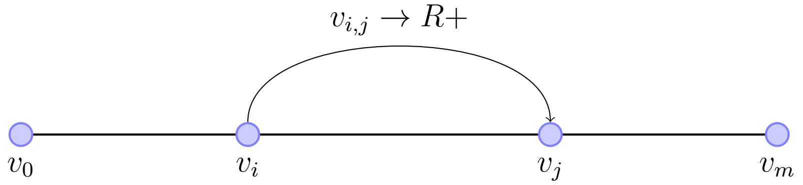

Figure 5.4: An edge between vertex v_i and v_j corresponds to one or more residues ($R+$). The neural network score for the residue(s) is equivalent to the neural network score for v_j.

is computed irrespective of the sequence of residues. The same holds true for the edge frequency score, since this score is also computed from edges in the spectrum graph.

5.4 Combined Scoring Function for Candidate Peptides

The edge frequency score is the normalized frequency of a mass difference in the spectrum, and the neural network score is a probability estimate for an edge's membership in the *b*-/*y*-ion ladder. Each of these edge scores map to one or more unique tags. In practice we treat these edge scores as individual residue scores by applying the tag score to its constituent residues. If a tag corresponds to a single residue then this is a trivial task, the tag score is the residue score. If a tag corresponds to two residues then the tag score is applied to each residue with a penalty since we wish to discourage the use of longer tags when a single residue is sufficient to explain an edge. The reasoning for this tag length restriction is due to the exponential combinatorial explosion of unique tags that match a given mass difference if we allow three or more residues to be used to construct a tag. Figures A.5 and A.6 illustrate the problem by showing the frequency of unique tags for lengths up to two and three. If missed cleavages occurred with high frequency then we would have to allow longer tag lengths in order to insure that the correct tag is included in the candidate peptide, and we would have to accept the combinatorial explosion along with it. However missed cleavages occur infrequently enough that we do not need to consider sequential pairs of missed cleavages. This can be seen in Figures A.1,A.2,A.3, and A.4, which show the pairwise cleavage frequencies for common sequence motifs.

The *QuasiNovo* scoring function for a candidate peptide C, consisting of residues $R_1 R_2 \cdots R_n$, combines the three scores into a single candidate peptide score via linear combination as shown below.

$$f_{PSM}(C) = \alpha \sum_{i=0}^{|C|-1} f_{SNN}(R_i) + \beta \sum_{i=0}^{|C|-1} f_{EF}(R_i) + \gamma \sum_{i=0}^{|C|-1} f_{AAU}(R_i) \tag{5.2}$$

The scalar values for each of the three scores were determined through grid-search to be $\alpha = 1.0$, $\beta = 10.0$, and $\gamma = 1.5$.

5.5 Experimental Results

To evaluate the *QuasiNovo* scoring function we used the NIJ dataset described in Chapter 3. We restricted the test set to peptides of length 8-12 and randomly selected 10 peptides for each length for a total of 50 peptides. For each MS/MS spectrum we used the SSN described in Chapter 3 to compute ion-type probability estimates for each peak, and then select peaks that likely correspond to *b*-/*y*-ions for generating vertex scores in the candidate generation step. The methods described in Chapter 4 were then used to compute a search space of candidate peptides. For these results the AAU score used tuples of length 5 ($Pr(R_i|R_{i-4}^{i-1})$), and the AAU distribution was constructed from 205 *Gammaproteobacteria* proteomes not including *E. coli.*, yielding approximately 23 million tryptic peptides. The candidate peptides were then scored using the *QuasiNovo* scoring function described above, and the top scoring candidate was used to generate the statistics presented below. We compared our results to what is currently the state of the art de novo peptide sequencing algorithm, PepNovo+. To measure the accuracy of our method we use the longest common subsequence in-place (LCSIP) metric. This metric is identical to longest common subsequence,

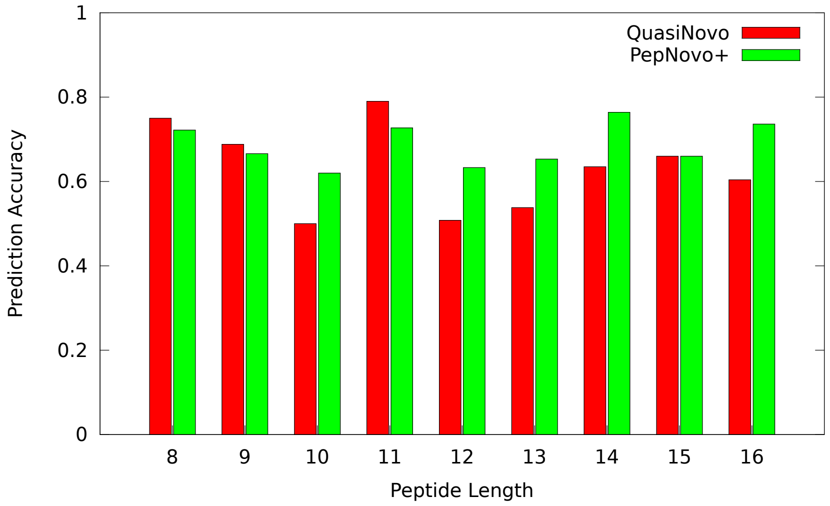

Figure 5.5: De novo results for peptides of length 8-16 comparing PepNovo+ and *QuasiNovo*.

but with the added constraint that a residue must be at the correct mass offset from either the N-terminus or the C-terminus to be considered correct (see Listing A.7). From this it is natural to define prediction accuracy (PA) as:

$$prediction\ accuracy = \frac{LCSIP}{length\ of\ correct\ peptide}.$$

When comparing the prediction accuracy we find that *QuasiNovo* achieves 62.4% and PepNovo achieves 68.8% for peptides of length 8 to 16. Results for each peptide length are shown in Figure 5.5. The peptides used to compute these prediction accuracy averages are given below in Table 5.1, along with comparisons between PepNovo+ and *QuasiNovo* top candidates for each peptide.

Table 5.1: Results for randomly selected NIJ peptides showing (a) the correct peptide, (b) the length of the correct peptide, (c) the top PepNovo+ candidate, (d/f) the prediction accuracy for peptide P for the top candidate, and (e) the top *QuasiNovo* candidate.

Correct peptide P $^{(a)}$	$\|P\|$ $^{(b)}$	PepNovo+ $^{(c)}$	PA $^{(d)}$	*QuasiNovo* $^{(e)}$	PA $^{(f)}$
QWFWQPPK	8	FGEKPPK	0.5	QWFWQPPK	1.0
TLFGDHER	8	TLFGDHER	1.0	TICTHDER	0.37
LGPVYSVR	8	GTVLVFVR	0.25	LGPVYDAR	0.75
LGSLQQAR	8	LGSLQQAR	1.0	IGSLQQAR	0.87
LADELCGR	8	LADELGCGR	1.0	IADELCGGR	0.87
FQDEEVQR	8	FQDEEVQR	1.0	FQDEEQVR	0.75
QGYPIGCK	8	EGYPLCNK	0.12	QGYQQGNK	0.5
GAIDMIVR	8	GALDMLVR	0.75	GAGSGGMD	0.25
AQHSLIHR	8	LSHSLLHR	0.62	AQHSLLHR	0.87
AAIEAAGGK	9	AALEAANK	0.66	AAIEAANK	0.77
VYQLPEATR	9	VYQLGSF	0.44	VYGALPEATR	0.88
IEEDLLGTR	9	LEEDLLGTR	0.88	IEEDLLGTR	1.0
VDAGFAITK	9	VDAGFGDLK	0.66	VSVGCTGGA	0.22
GVFNLVLGR	9	FNLVLGR	0.77	GVFNLVLGR	1.0
MFTINAEVR	9	VPLTASYLSK	0.0	DYYNHNVR	0.22
INQVYVVLK	9	LNQVYVVLK	0.88	INQVYVVLK	1.0
VTVQDAVEK	9	VTVQDAVEK	1.0	VTQVDAVTR	0.55
YLDLIANDK	9	YLDLLATQK	0.66	YLDLLATQK	0.66
SGMHQDVPK	9	THKDVPK	0.55	QFHQDVPK	0.66
STVTITDLAR	10	LLVGYSH	0.0	NLVFTLTLR	0.1
QQIIGLAEVR	10	QQLLGLAEA	0.6	QQLLGLSLVR	0.6
QQLPDDATLR	10	QQLPMVATLR	0.8	QQIPMVATLR	0.7
DHIVGLNCGR	10	DHLVGLNNCR	0.7	DHIVGLHHGR	0.8
CTEEHQAIVR	10	AEEHQALVR	0.7	CAVIAQAQALVR	0.5
LLSPEVANDK	10	LLSPLDAF	0.5	IISTHSNEGK	0.2
AGPTWTPTAK	10	AGPTGDDLRK	0.5	QPTGETTAPK	0.4
ANAYGHGIER	10	ANAYGHGLER	0.9	ANAYGHVAER	0.8
AVQEQVASEK	10	GLQEQVASEK	0.8	AVQEQVASEK	1.0
AGENVGVLLR	10	NVGVLLR	0.7	IGSGGVGVLL	0.5
MLTEANLNSLR	11	LTEANLNSLR	0.90	LMNEEVGNSLR	0.36
ELANVQDLTVR	11	ELANVKDLT	0.72	LEANVQDLTVR	0.81
GCYTGQEMVAR	11	YTGQEMVAR	0.81	NCYTGQEMVAR	0.90
DAWATGNPALR	11	WATGNPALR	0.81	WADATGNPALR	0.72
QALENVSTWVR	11	AQLENWTLTAR	0.45	AQLEATQTWVR	0.54
DLVESAPAALK	11	NNVESAPAALK	0.81	VEVESAPAALK	0.81
ELLTNDPFSSR	11	PLALVYTGVTSR	0.18	ELITNPDTGYR	0.45
FAAACEHFVSR	11	FAAAGCHEFVSR	0.81	FAAAFEHGSDSR	0.54
TMLFDAPLQMK	11	LFDAPLQMK	0.81	TMLFDAPLQMK	1.0
LYNDAGISNDR	11	LYNDAGLSNDR	0.90	LYNDQLSNDR	0.72
GDVLNYDEVMER	12	DGVLNYDEVME	0.75	VWVNYDEVMER	0.66
NQSSDWQQYNIK	12	NQSSDWQQYNLK	0.91	LESSDWQQYNLK	0.75
GVGQIHPIFADR	12	LHPLFWR	0.33	GHFHIQYHDR	0.25
NTSFAPGNVSIK	12	SFANPGVSLK	0.5	SQSFAGGPVGSLA	0.41
EPISVSSQQMLK	12	EPLSVSSQQMLK	0.91	ILLSVSSQGAMLK	0.66
GDIVLCGFEYGR	12	DNAQCEYGR	0.33	VWVLGCCTEYGR	0.58
TLAVVGESGCGK	12	TLAVVGES	0.66	TLAVVGESGCGA	0.91
VQSMPEINDADK	12	NLSMPELSSLKK	0.41	NISMCAVPADVVK	0.25
ITSVNVGGMAFR	12	DVSVNVGGMAFR	0.83	ITSVNVQDFTK	0.5
AAMSGMLSPELK	12	GMLSPELK	0.66	SWSGMLSLLLK	0.58

Chapter 6
Reranking Candidate Peptides

By examining proteins and compiling amino acid usage (AAU) distributions it is possible to characterize likely combinations of amino acids and better distinguish between candidate peptides. In the previous chapter we used AAU in combination with other scoring functions to score candidate peptides. In this chapter we use AAU alone to score and rerank candidate peptides produced by other de novo algorithms, and show that a scoring function that considers amino acid usage patterns is better able to distinguish between candidate peptides. This in turn leads to higher accuracy in peptide prediction.

6.1 Consideration of Amino Acid Usage

One important piece of information that is missing from current probabilistic and cross-correlation scoring function is the prior distribution of amino acid usage. This distribution describes the percentage of each amino acid as well as the probability of combinations of amino acids in peptide sequences. It captures the mutual information present in adjacent residues in the protein sequences from which the distribution was derived. By leaving this information out, one is effectively using a flat prior that treats all combination of amino acids as equally likely. NovoHMM is an interesting exception. Although NovoHMM uses a hidden Markov model instead of likelihood model, it implicitly incorporates information concerning amino acid usage by training with spectrum/peptide pairs[12]. However, NovoHMM's understanding of amino acid usage is inherently limited since it is derived entirely from available spectrum/peptide training pairs.

Researchers have recognized that there is bias in the types of peptides that are consistently observed by current MS/MS technology. These preferentially observed peptides are called proteotyptic peptides[8, 42, 27]. In this case, the bias is not a reflection of the proteome signature but of the experimental protocol and MS/MS technology. PepNovo+ employs a ranking algorithm to rerank candidate peptides produced by its fragmentation model. While the PepNovo+ ranking algorithm considers sequence composition features, it is limited to amino acid triplets that are compiled from proteotypic sequences. The result is a single distribution describing the proteotypic character of triplets averaged over all such training sequences[15].

In contrast, the *QuasiNovo* AAU scoring function described in this chapter recognizes that amino acid usage can vary widely from organism to organism[13, 39]. Typically it is similar between closely related taxa but can be quite different when taxa are distantly related. Consequently, *QuasiNovo*'s understanding of amino acid usage is provided by several models. These models are derived from protein sequence data alone. This data is much more plentiful and accurate than spectrum/peptide pairs and leads to a more detailed and nuanced understanding of amino acid usage. In this chapter we present results supporting the hypothesis that a scoring function that takes amino acid usage into account can significantly improve the accuracy of peptides derived via de novo sequencing.

6.2 Methods and Data

Our investigations were designed to evaluate the utility of a scoring function based on amino acid usage distributions. The AAU distributions were created as described in the previous chapter.

The amino acid distribution models amino acid usage and can be used to estimate the probability of observing an amino acid sequence. This model is used to compute the probability of a peptide of n amino acids by taking the product of the probability of the first tuple of length $L-1$ times the subsequent $n-L+1$ overlapping conditional probabilities based on tuples of length L in the peptide, *i.e.*,

$$P_{AAU}(P|M_{AAU}) = p(P_{1,L-1}) \prod_{i=L}^{n} p(P_i|P_{i-L+1,i-1}) \tag{6.1}$$

In this equation $p(P_{1,L-1})$ is the probability of the first $L-1$ residues in peptide P and $p(P_i|P_{i-L+1,i-1})$ is the conditional probability of the i^{th} amino acid given the preceding $L-1$ amino acids. The probabilities $p(P_{1,L-1})$ and $p(P_i|P_{i-L+1,i-1})$ are defined by the amino acid usage model M_{AAU}, *i.e.*, the normalized amino acid distribution.

If a de novo sequencing algorithm with this type of scoring function could be shown to be competitive with existing de novo sequencing algorithms then one would expect a model that combined a probabilistic fragmentation model with an amino acid usage prior to perform substantially better than one using an implicit flat prior. To this end, we selected the same data set of 280 spectra used by Frank and Pevzner[16]. They used this data set to compare PepNovo with Sherenga, PEAKS, and Lutefisk. This data set comes from two sources, the ISB protein mixture data set[22] and the Open Proteomics Database (OPD)[34]. In this data set, peptides average 10.5 residues in length. Frank and Pevzner demonstrated that PepNovo outperformed Sherenga, Peaks and Lutefisk on this data set. This data set was also used by Fischer et al. to compare NovoHMM with PepNovo, Sherenga, PEAKS, and Lutefisk[12]. In their study NovoHMM outperformed its competitors. Consequently, the focus of our evaluation was a comparison of the results of our scoring function versus PepNovo and NovoHMM.

The 280 spectra in the Frank-Pevzner data set are comprised of spectra from 174 *Escherichia coli* peptides, 27 *Mycobacterium smegmatis* peptides, 67 *Bos taurus* peptides, and 12 *Homo sapiens peptides.* The three major categories represented in this data set are *Gammaproteobacteria* (*E. coli*), *Actinobacteria* (*M. smegmatis*), and *Mammalia* (*B. taurus* and *H. sapiens*). Amino acid distributions were constructed

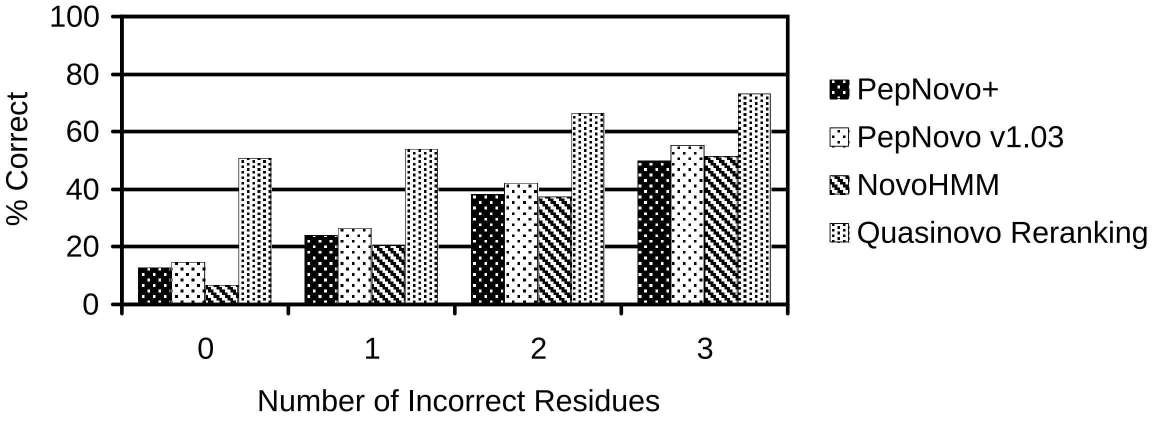

Figure 6.1: Results for set of 280 MS/MS test spectra comparing PepNovo+, PepNovo, NovoHMM, and *QuasiNovo* AAU reranking.

for each of the 3 categories. *E. coli* and *M. smegmatis* peptides were specifically excluded from their respective distributions to demonstrate the ability of sequencing novel peptides. The *Gammaproteobacteria* distribution was constructed from approximately 23 million tryptic peptides from 205 gammaproteobacterial proteomes not including *E. coli.* The *Actinobacteria* distribution was constructed from approximately 7 million tryptic peptides from 57 complete actinobacterial genomes, not including *M. smegmatis.* Similarly, two mammalian distributions were created, one excluding *H. sapiens* and the other excluding *B. taurus.* The mammalian distribution used to score *H. sapiens* peptides was constructed from the complete proteomes of *B. taurus*, *R. norvegicus*, and *M. musculus.* The distribution used to score *B. taurus* peptides was constructed from complete proteomes of *H. sapiens*, *R. norvegicus*, and *M. musculus.*

6.3 Experimental Results and Discussion

Initial results are shown in Figure 6.1. The common practice in the de novo sequencing literature of presenting results in terms of the number of predictions that are correct within one, two, and three amino acids was followed. Each category is cumulative, *e.g.*, the category correct within 3 residues also includes the number of peptides with fewer errors. This figure depicts the accuracy of the top scoring candidate as selected by each method. Both PepNovo and NovoHMM produce a single top scoing candidate. In contrast, using default settings PepNovo+ produces 50 peptides sorted by rank. The *QuasiNovo* AAU scoring function was used to rescore candidate peptides produced by PepNovo, PepNovo+, and NovoHMM. For each spectrum, a set comprised of the top 50 candidates produced by PepNovo+ and the single candidates produced by PepNovo and NovoHMM was created. The *QuasiNovo* AAU scoring function was then used to select the peptide that produced the highest resulting score from this set. These results are labeled *QuasiNovo* Reranking in Figure 6.1. The most striking feature of the results presented in Figure 1 is that the *QuasiNovo* AAU Reranking scores significantly higher than do PepNovo, PepNovo+ and NovoHMM. Recall that this reranking entails taking the 50 peptides suggested by PepNovo+ and the single peptides suggested by PepNovo and NovoHMM and then selecting the peptide with the highest *QuasiNovo* AAU score. These results indicate that amino acid usage carries conditioning information about protein sequences that provides additional precision in mapping from the spectrum to the corresponding peptide.

A common alternative performance metric in the de novo sequencing literature is to present results in terms of the percentage of correct contiguous subsequences[16, 12, 10]. Not all de novo sequencing algorithms predict complete peptides. Often the peaks near the terminal ends are weak or missing. Consequently, the correct subsequences tend to be in the middle of the peptide. Figure 6.2 presents the longest subsequence results for the Frank-Pevzner dataset of 280 spectra. These results were derived by first finding the longest correct subsequence in the data set for each algorithm and then tallying the counts for each length. In this figure, the curve corresponding to the *QuasiNovo* AAU reranking dominates the other curves by a significant amount over all subsequence lengths of four and greater.

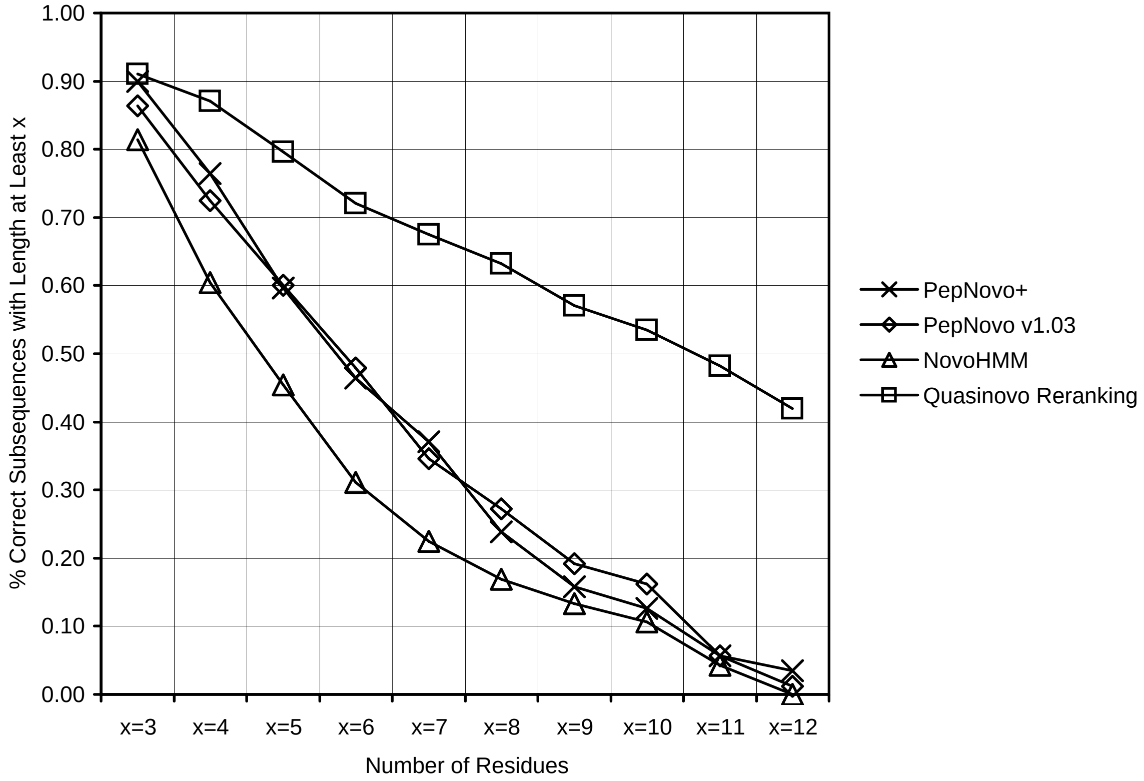

Figure 6.2: Cumulative results for set of 280 MS/MS test spectra illustrating the proportions of predictions that had a correct subsequence of length at least x, for $3 \leq x \leq 12$.

DiMaggio and Floudas used 100 spectra from the Frank-Pevzner data set[16] to compare PILOT with PepNovo, and EigenMS. In their study[12], the PILOT results were slightly better than those of PepNovo and EigenMS. We evaluated *QuasiNovo* AAU reranking of the peptides proposed by PepNovo+ and NovoHMM for this set in order to see what effect the consideration of amino acid usage would have. The results of the reranking are shown in Figure 6.3. Notice that the results for PILOT only indicate the number of peptides (out of 100) that are correct within 2 amino acids and within 3 amino acids. This is because DiMaggio and Floudas did not publish results for completely correct peptides. It is instructive to note that the *QuasiN ovo* AAU reranking of the PepNovo and NovoHMM results increase the number of completely correct peptides from 47 and 50, respectively, to 72. Finally, even in the

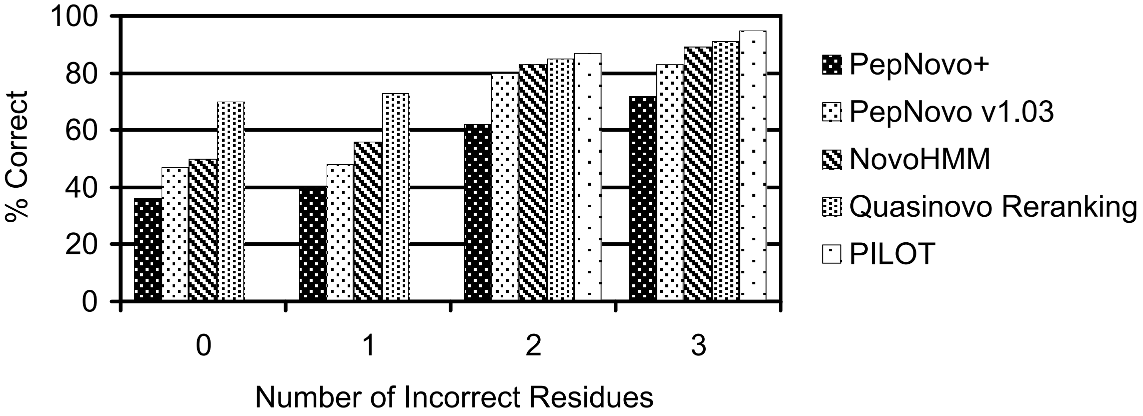

Figure 6.3: Results for set of 100 MS/MS test spectra comparing PepNovo+, PepNovo, NovoHMM, PILOT and *QuasiNovo* AAU reranking.

case of peptides considered correct within 3 amino acids where PILOT achieves 95 out of 100, the *QuasiNovo* AAU reranking results are 93 out of 100, *i.e.*, comparable results.

The results in Figure 6.3 assume isobaric residues to be equivalent since DiMaggio and Floudas published these statistics but did not publish the actual peptides proposed by PILOT for this data set. Specifically, this means that the pairs I/L and Q/K are treated as identical amino acids. For example, the assignment of an isoleucine in the candidate peptide where the actual peptide contains a leucine is considered correct. This is a common practice since de novo sequencing algorithms that do not take amino acid usage into account have no basis for distinguishing between isobaric residues. Since *QuasiNovo* models amino acid usage, its scoring function is able to distinguish among isobaric residues. Consequently, *QuasiNovo* selects the residue with the highest probability in the context of a given peptide. Another weakness of methods that do not consider amino acid usage lies in how they treat missing peaks. This commonly occurs when the b_1-ion (corresponding to the N-terminal amino acid) is missing from the spectrum. Peaks corresponding to other b-/y-ions may also be missing from the spectrum. Since peaks corresponding to b_1-ion are frequently missing, more errors would be expected in the prediction of this terminal residue. If a

Table 6.1: Comparison of Terminal Pair and Overall Accuracy

	terminal ion pair		complete peptide
algorithm	b_2-ion	y_2-ion	
PepNovo+	0.509	0.616	0.702
NovoHMM	0.523	0.759	0.735
QuasiNovo AAU Reranking	0.716	0.813	0.815

peak corresponding to a b_1-ion is missing from the spectrum then a de novo sequencing algorithm must make a prediction based on the next peak in the ion ladder, *i.e.*, the b_2-ion. Table 6.1 shows the accuracy of the predictions made by PepNovo+, NovoHMM, and the *QuasiNovo* AAU reranking for terminal ion pairs in the Frank-Pevzner dataset of 280 spectra. Table 6.1 does not assume isobaric equivalence. The values in the table were derived by tallying the number of correctly predicted terminal pair residues. For example, the values in the b_2-ion column were determined by summing the number of correctly predicted residues in the first two positions in the 280 peptides and then dividing by 560, *i.e.*, 2 residues * 280 peptides. As shown in Table 6.1, the *QuasiNovo* AAU reranking results are superior to those of PepNovo+ and NovoHMM for predicting the correct residue pairs corresponding b_2-ions and y_2-ions. Notice that the accuracy of the amino acids predicted for the y_2-ion for all algorithms in Table 6.1 are closer to the accuracy for the complete peptide than they are to the b_2-ion. The y_1-ion is not as frequently missing from the spectrum as the b_1-ion.

When a peak is missing and can not be inferred, methods that do not model amino acid usage are typically able to propose a combination of residues for that part of the peptide. However, they are not able to specify the particular order in which the combination of residues appear in the peptide. It is for this reason that it has become common practice to present results in terms of percentage of predictions that are correct within one, two, and three amino acids as shown in Figures 6.1 and 6.3. In contrast, *QuasiNovo* uses its model of amino acid usage to distinguish between possible permutations. On this basis it selects the permutation with the greatest

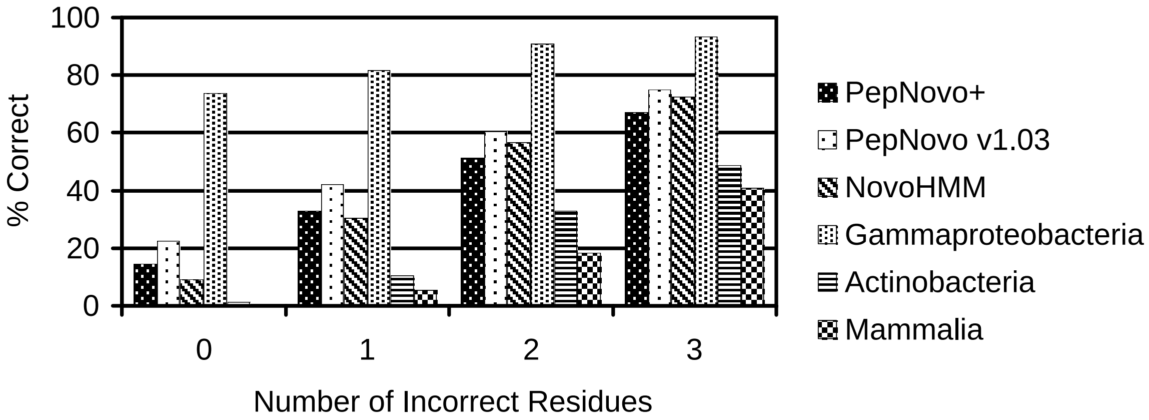

Figure 6.4: Results for set of 76 MS/MS test spectra for E. coli peptides comparing PepNovo+, PepNovo, NovoHMM, with three *QuasiNovo* scoring functions based on amino acid distributions in *Gammaproteobacteria*, *Actinobacteria*, and *Mammalia.*

probability.

The results in Table 6.1 as well as the preceding figures demonstrate the utility of integrating amino acid usage considerations in a scoring function. An obvious question is what influence the choice of proteomes used to build the AAU distributions has on the accuracy of the peptide scoring function. The 280 spectra in the Frank-Pevzner data set are comprised of 3 major categories: *Gammaproteobacteria*, *Actinobacteria*, and *Mammalia.* The experiment shown in Figure 6.3 was repeated. However, this time three different AAU-based scoring functions were used to evaluate the subset of 76 *E. coli* peptides from the original 100 peptides. The scoring function labeled *Gammaproteobacteria* was compiled using only amino acid usage data from *Gammaproteobacteria* proteomes. Similarly, the scoring functions labeled *Actinobacteria* and *Mammalia* were derived exclusively from amino acid usage data from *Actinobacteria* and *Mammalia*, respectively. Given that all 76 peptides are from *E. coli*, it is no surprise that the results for the scoring function derived from gammaproteobacterial peptides are significantly higher than the other two scoring functions as shown in Figure 6.4. This data hints at the sensitivity of the accuracy of the *QuasiN ovo* AAU scoring function to the peptide data from which it is constructed. It should also be noted that the results shown in Figure 6.4 do not assume isobaric residues to be equivalent. One of the strengths of considering amino acid usage is that it provides a statistical basis for choosing between isobaric equivalent residues.

These results show that amino acid usage can be used as prior information to improve significantly the accuracy of the scoring functions used by current de novo sequencing algorithms. They also support the hypothesis that a significant additional increase in sequencing accuracy could be attained by including consideration of amino acid usage as an integral component of a scoring function.

The results of our investigations conclusively demonstrate two results. First, when we use an AAU-based scoring function to re-rank a combination of PepNovo+'s, PepNovo's, and NovoHMM's candidate peptides, the peptide that gets our highest score demonstrates significant improvement in accuracy as compared to PepNovo+'s, PepNovo's, and NovoHMM's first choice. Second, top-performing de novo sequencing programs such as PepNovo+ are able to generate good quality candidate peptides. In addition, they are able to compute candidate peptides very efficiently. For example, DiMaggio and Floudas report that PILOT takes 5-20 seconds to evaluate a spectrum on an Intel Pentium 4 3.0GHz Linux-based computer[10]. Even so, they are often not able to correctly rank them. As a consequence, a suboptimal candidate is selected as the highest ranking peptide and the accuracy is considerably lower than it should be. Put simply, scoring functions that do not consider amino acid usage appropriately are often not able to select the most correct peptide from a pool of candidates.

The comparison of amino acid usage models in Figure 6.4 shows that the choice of amino acid usage model is important. Consequently, this is another important area of investigation. The results of the *Gammaproteobacteria* model in Figure 6.4 are impressive when compared with those of PepNovo, PepNovo+ and NovoHMM. The *Gammaproteobacteria* amino acid usage model in Figure 6.4 was compiled by aggregating data from the 205 proteomes from the *Gammaproteobacteria* class. Even the models constructed for mammalian peptides were not particularly focused, containing data from *H. sapiens*, *B. taurus*, *R. norvegicus*, and *M. musculus.* It is reasonable to expect that the accuracy of the scoring function will be improved by creating statistical models of amino acid usage that are closer to the AAU distribution of the peptides under consideration. It is hypothesized that more focused models at the level of family, and genus will demonstrate greater improvements in accuracy relative to the results presented here.

One argument for pursuing de novo sequencing is the ability to sequence peptides expressed from unsequenced genomes. In the case of such a peptide, it is not possible to have the actual amino acid usage model. However in the case of bacteria, simple physiological tests (*e.g.* Gram stain) for cultured organisms can help limit the data set or limit the taxonomic categories under examination. For un-cultured single cells, equivalent information may also be obtainable. In both cases, it is possible to use universal primers to extract small subunit ribosomal RNA and sequence rRNA genes without having to sequence the entire genome. SSU rRNA databases are already the main source of microbial diversity information owing to rRNAs' role as the gold standard for microbial identification[30]. While the high degree of conservation of rRNA genes reduces their usefulness in resolving fine details at the strain or species level, it nevertheless makes them useful for inference of deep phylogeny. This informaion can be used to select the most appropriate available amino acid usage model.

Chapter 7

Conclusion

One of the primary challenges faced by MS-based proteomics is how to perform faster and more accurate automated data analysis. This includes eliminating the requirement of a sequence database which restricts analysis to organisms that have sequenced genomes and spectral libraries. Towards this goal the field of proteomics needs better de novo sequencing algorithms. While higher precision mass spectrometers have somewhat improved the quality of de novo peptide sequencing, the world is still in need of effective approaches to peptide sequencing that use common low precision instruments and established laboratory protocols.

7.1 Impact of this Research

Accurate analysis of MS/MS data is a challenging process that relies on a complex understanding of molecular dynamics, signal processing, and pattern classification–or at the very least relies on modeling these aspects of MS/MS data. Our contribution to the field of computational mass spectrometry is a complete peptide sequencing software package called *QuasiNovo* that uniquely addresses these concerns and contributes significantly to the field. In this dissertation we described the problem in broad terms, divided the problem into its separable components, and described the specific solutions we brought to bear on each step of the problem. The *QuasiNovo* algorithm is summarized as follows:

First, we developed a novel peak classifier that used a staged neural network to estimate the probability that a given peak is a b-/y-ion. Peak selection is an important preprocessing step in de novo sequencing. As a practical matter, it is important that the number of peaks be reduced so that the candidate peptide search space is constrained. A reduction in the number of peaks used to create the spectrum graph makes it possible to process spectra faster. It also makes it possible to process longer peptides than would otherwise be impractical since the resulting search space is smaller. The staged neural network probability estimates for each ion type are used to filter the raw MS/MS spectrum. The same probability estimates are then used for scoring vertices in the spectrum graph, and then used in combination with other scoring functions to score candidate peptides.

Second, a boilerplate approach to generating feasible paths in a spectrum graph—originally conceived by Dancik *et al.* [9] and later improved by Lu *et al.* [25]—was modified for our purposes to produce candidate peptide sequences.

Third, the candidate peptides are scored using the *QuasiNovo* scoring function. The scoring function has three components: the SNN score which propagates from the initial peak selection step, an amino acid usage score, and an edge frequency score. Each of these scoring functions are novel in the field of de novo peptide sequencing.

Fourth—in what happens to be our earliest work—we explored reranking candidate peptides using amino acid usage distributions.

7.2 Future Work

As *QuasiNovo* continues to be developed there are some obvious improvements and extensions that are anticipated. Much of this work serves as a proof of concept from a software engineering perspective. A capable programmer could find several areas of the software that could benefit from optimization. For example, much of the SNN is written in the scripting language Ruby. While Ruby is an excellent language for rapid prototyping, it is orders of magnitude slower than a compiled language, and so this component of the software will be refactored accordingly to reduce runtime. The scoring and reranking components of the software are written in C++, however we expect the memory footprint can be reduced dramatically during candidate generation, which would reduce the runtime due to the smaller set of candidates that need to be scored and ranked.

The majority of the planned developments for the scientific aspects of the research concern the creation and analysis of AAU distributions. While we have already conducted several studies of different AAU distributions, it is important to continue compiling and investigating AAU distributions at different taxonomic levels and of varying composition. AAU distributions that vary according to GC content, protein family, and proteotypic propensity will be explored, and various information theoretic measures of the distributions will be studied.

Bibliography

[1] Nuno Bandeira, Karl R Clauser, and Pavel A Pevzner, *Shotgun protein sequencing: assembly of peptide tandem mass spectra from mixtures of modified proteins.*, Molecular & cellular proteomics : MCP **6** (2007), no. 7, 1123–34.

[2] Nuno Bandeira, Dekel Tsur, Ari Frank, and Pavel A Pevzner, *Protein identification by spectral networks analysis.*, Proceedings of the National Academy of Sciences of the United States of America **104** (2007), no. 15, 6140–5.

[3] Marshall Bern, Yuhan Cai, and David Goldberg, *Lookup peaks: a hybrid of de novo sequencing and database search for protein identification by tandem mass spectrometry.*, Analytical chemistry **79** (2007), no. 4, 1393–400.

[4] Ting Chen, M Y Kao, M Tepel, J Rush, and G M Church, *A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry.*, Journal of computational biology : a journal of computational molecular cell biology **8** (2001), no. 3, 325–37.

[5] Hao Chi, RX Sun, Bing Yang, CQ Song, and LH, *pNovo: De novo Peptide Sequencing and Identification Using HCD Spectra*, Journal of Proteome (2010), 2713–2724.

[6] James P Cleveland and John R Rose, *A Neural Network Approach to Pre-filtering MS / MS spectra*, ISBRA 2012, 2012, pp. 82–84.

[7] JP Cleveland and JR Rose, *A Neural Network Approach to the Identification of b-/y-ions in MS/MS Spectra*, 2012 IEEE International Conference on Bioinformatics and Biomedicine, 2012, pp. 588–592.

[8] Robertson Craig, John P Cortens, and Ronald C Beavis, *The use of proteotypic peptide libraries for protein identification.*, Rapid communications in mass spectrometry : RCM **19** (2005), no. 13, 1844–50.

[9] V Dančík, T A Addona, K R Clauser, J E Vath, and P A Pevzner, *De novo peptide sequencing via tandem mass spectrometry.*, Journal of computational biology : a journal of computational molecular cell biology **6** (1999), no. 3-4, 327–42.

[10] Peter a DiMaggio and Christodoulos a Floudas, *De Novo Peptide Identification via Tandem Mass Spectrometry and Integer Linear Optimization*, Analytical chemistry **79** (2007), no. 4, 1433–46.

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

[12] Bernd Fischer, Volker Roth, Franz Roos, Jonas Grossmann, Sacha Baginsky, Peter Widmayer, Wilhelm Gruissem, and Joachim M Buhmann, *NovoHMM: a hidden Markov model for de novo peptide sequencing.*, Analytical chemistry **77** (2005), no. 22, 7265–73.

[13] P G Foster and D A Hickey, *Compositional bias may affect both DNA-based and protein-based phylogenetic reconstructions.*, Journal of molecular evolution **48** (1999), no. 3, 284–90.

[14] Ari Frank, *Algorithms for tandem mass spectrometry-based proteomics*, Ph.D. thesis, University of California, San Diego, 2008.

[15] ______, *A Ranking-Based Scoring Function for Peptide-Spectrum Matches*, Journal of proteome research **8** (2009), no. 5, 2241–2252.

[16] Ari Frank and Pavel Pevzner, *PepNovo: de novo peptide sequencing via probabilistic network modeling.*, Analytical chemistry **77** (2005), no. 4, 964–73.

[17] Ari Frank and MM Savitski, *De novo peptide sequencing and identification with precision mass spectrometry*, Journal of proteome . . . (2007), 114–123.

[18] Ari Frank, Stephen Tanner, and Vineet Bafna, *Peptide sequence tags for fast database search in mass-spectrometry*, Journal of proteome (2005), 1287–1295.

[19] Yonghua Han, Bin Ma, and Kaizhong Zhang, *SPIDER: software for protein identification from sequence tags with de novo sequencing error*, Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE, no. Csb, IEEE, January 2004, pp. 206–215.

[20] Yingying Huang, Joseph M Triscari, Ljiljana Pasa-Tolic, Gordon a Anderson, Mary S Lipton, Richard D Smith, and Vicki H Wysocki, *Dissociation behavior of doubly-charged tryptic peptides: correlation of gas-phase cleavage abundance*

with ramachandran plots., Journal of the American Chemical Society **126** (2004), no. 10, 3034–5.

[21] Yingying Huang, Joseph M Triscari, George C Tseng, Ljiljana Pasa-Tolic, Mary S Lipton, Richard D Smith, and Vicki H Wysocki, *Statistical characterization of the charge state and residue dependence of low-energy CID peptide dissociation patterns.*, Analytical chemistry **77** (2005), no. 18, 5800–13.

[22] Andrew Keller, Samuel Purvine, Alexey I Nesvizhskii, Sergey Stolyar, David R Goodlett, and Eugene Kolker, *Experimental protein mixture for validating tandem mass spectral analysis.*, Omics : a journal of integrative biology **6** (2002), no. 2, 207–12.

[23] Jainab Khatun, Kevin Ramkissoon, and M.C. Morgan C Giddings, *Fragmentation characteristics of collision-induced dissociation in MALDI TOF/TOF mass spectrometry.*, Analytical chemistry **79** (2007), no. 8, 3032–40.

[24] Daniel T Lavelle and William R Pearson, *Globally, unrelated protein sequences appear random.*, Bioinformatics (Oxford, England) **26** (2010), no. 3, 310–8.

[25] Bingwen Lu and Ting Chen, *A suboptimal algorithm for de novo peptide sequencing via tandem mass spectrometry.*, Journal of computational biology : a journal of computational molecular cell biology **10** (2003), no. 1, 1–12.

[26] Bin Ma, K Zhang, and C Liang, *An effective algorithm for peptide sequencing from MS/MS spectra*, Journal of Computer and System Sciences **70** (2005), no. 3, 418–430.

[27] Parag Mallick, Markus Schirle, Sharon S Chen, Mark R Flory, Hookeun Lee, Daniel Martin, Jeffrey Ranish, Brian Raught, Robert Schmitt, Thilo Werner, Bernhard Kuster, and Ruedi Aebersold, *Computational prediction of proteotypic peptides for quantitative proteomics.*, Nature biotechnology **25** (2007), no. 1, 125–31.

[28] M Mann and M Wilm, *Error-tolerant identification of peptides in sequence databases by peptide sequence tags.*, Analytical chemistry **66** (1994), no. 24, 4390–9.

[29] Lijuan Mo, Debojyoti Dutta, Yunhu Wan, and Ting Chen, *MSNovo: a dynamic programming algorithm for de novo peptide sequencing via tandem mass spectrometry.*, Analytical Chemistry **79** (2007), no. 13, 4870–4878.

[30] Norman R Pace, *Mapping the tree of life: progress and prospects.*, Microbiology and molecular biology reviews : MMBR **73** (2009), no. 4, 565–76.

[31] Béla Paizs and Sándor Suhai, *Fragmentation pathways of protonated peptides.*, Mass spectrometry reviews **24** (2005), no. 4, 508–48.

[32] Itsik Pe'er, Clifford E Felder, Orna Man, Israel Silman, Joel L Sussman, and Jacques S Beckmann, *Proteomic signatures: amino acid and oligopeptide compositions differentiate among phyla.*, Proteins **54** (2004), no. 1, 20–40.

[33] D N Perkins, D J Pappin, D M Creasy, and J S Cottrell, *Probability-based protein identification by searching sequence databases using mass spectrometry data.*, Electrophoresis **20** (1999), no. 18, 3551–67.

[34] John T Prince, Mark W Carlson, Rong Wang, Peng Lu, and Edward M Marcotte, *The need for a public proteomics repository.*, Nature biotechnology **22** (2004), no. 4, 471–2.

[35] Bernhard Y Renard, Marc Kirchner, Flavio Monigatti, Alexander R Ivanov, Juri Rappsilber, Dominic Winter, Judith a J Steen, Fred a Hamprecht, and Hanno Steen, *When less can yield more - Computational preprocessing of MS/MS spectra for peptide identification.*, Proteomics **9** (2009), no. 21, 4978–84.

[36] John R Rose, James P Cleveland, and Alvin Fox, *An Information Theoretic Approach to Rescoring Peptides Produced by De Novo Peptide Sequencing*, ICBCB 2010: International Conference on Bioinformatics and Computational Biology (Paris, France), World Academy of Science, Engineering and Technology, 2010, pp. 200–205.

[37] Brian C Searle, Surendra Dasari, Mark Turner, Ashok P Reddy, Dongseok Choi, Phillip A Wilmarth, Ashley L McCormack, Larry L David, and Srinivasa R Nagalla, *High-throughput identification of proteins and unanticipated sequence modifications using a mass-based alignment algorithm for MS/MS de novo sequencing results.*, Analytical chemistry **76** (2004), no. 8, 2220–30.

[38] LM Silva and J Marques de Sá, *Data classification with multilayer perceptrons using a generalized error function*, Neural Networks **21** (2008), no. 9, 1302–10.

[39] Gregory A. C. Singer and Donal A. Hickey, *Nucleotide Bias Causes a Genomewide Bias in the Amino Acid Composition of Proteins*, Mol. Biol. Evol. **17** (2000), no. 11, 1581–1588.

[40] David L. Tabb, *Statistical Characterization of Ion Trap Tandem Mass Spectra from Doubly Charged Tryptic Peptides*, Analytical Chemistry **75** (2003), no. 5, 1155–1163.

[41] David L Tabb, Anita Saraf, and John R Yates, *GutenTag: high-throughput sequence tagging via an empirically derived fragmentation model.*, Analytical chemistry **75** (2003), no. 23, 6415–21.

[42] Haixu Tang, Randy J Arnold, Pedro Alves, Zhiyin Xun, David E Clemmer, Milos V Novotny, James P Reilly, and Predrag Radivojac, *A computational approach toward label-free protein quantification using predicted peptide detectability.*, Bioinformatics (Oxford, England) **22** (2006), no. 14, e481–8.

[43] Hans J C T Wessels, Tom G Bloemberg, Maurice van Dael, Ron Wehrens, Lutgarde M C Buydens, Lambert P van den Heuvel, and Jolein Gloerich, *A comprehensive full factorial LC-MS/MS proteomics benchmark data set.*, Proteomics **12** (2012), no. 14, 2276–81.

[44] Natalie Wielsch, Henrik Thomas, Vineeth Surendranath, Patrice Waridel, Ari Frank, Pavel Pevzner, and Andrej Shevchenko, *Rapid validation of protein identifications with the borderline statistical confidence via de novo sequencing and MS BLAST searches.*, Journal of proteome research **5** (2006), no. 9, 2448–56.

[45] Vicki H Wysocki, *Peptide Fragmentation Overview*, Principles of Mass Spectrometry Applied to Biomolecules, Wiley, 2006, pp. 279–300.

This research was supported by NSF award 0959427 and a grant from the Sloan Foundation Indoor Air Program. Some of the experiments were run on an SGI Altix 4700 system with 128 computing cores and 256GB shared memory funded by NSF award 0708391.

Appendix A

Additional Figures and Listings

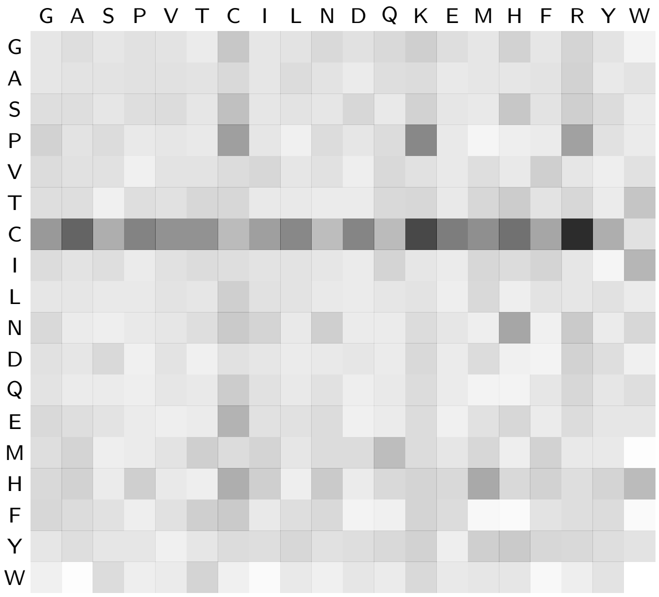

Figure A.1: Pair-wise cleavage probability for *b*-/*y*-ions from peptides that have no internal K/R, and end in K/R, *i.e.,* peptides matching the sequence motif regular expression /^[^KR]*[KR]$/. Black indicates a probability of zero, and white indicates a probability of one.

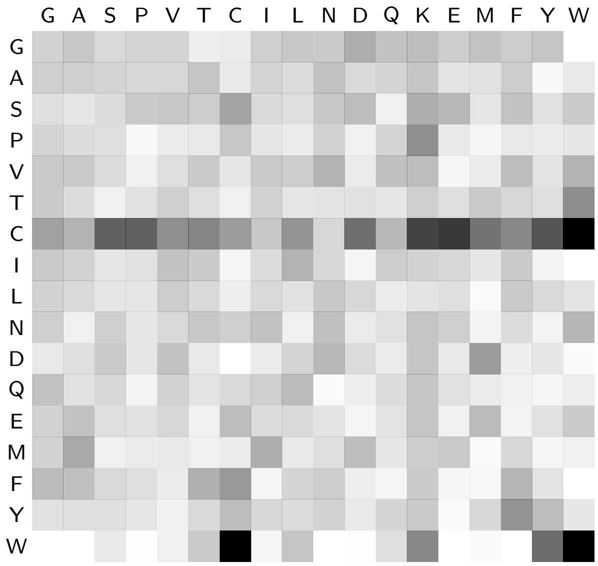

Figure A.2: Pair-wise cleavage probability for *b*-/*y*-ions from peptides that have no internal K/R/H, at least one internal P, and end in K, *i.e.,* peptides matching the sequence motif regular expression /^[^HKR]*P[^HKR]*[K]$/. Black indicates a probability of zero, and white indicates a probability of one.

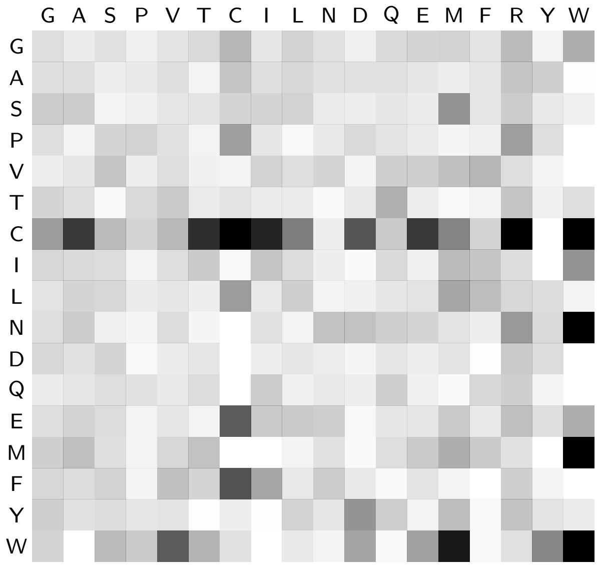

Figure A.3: Pair-wise cleavage probability for *b*-/*y*-ions from peptides that have no internal K/R/H, at least one internal P, and end in R, *i.e.,* peptides matching the sequence motif regular expression /^[^HKR]*P[^HKR]*[R]$/. Black indicates a probability of zero, and white indicates a probability of one.

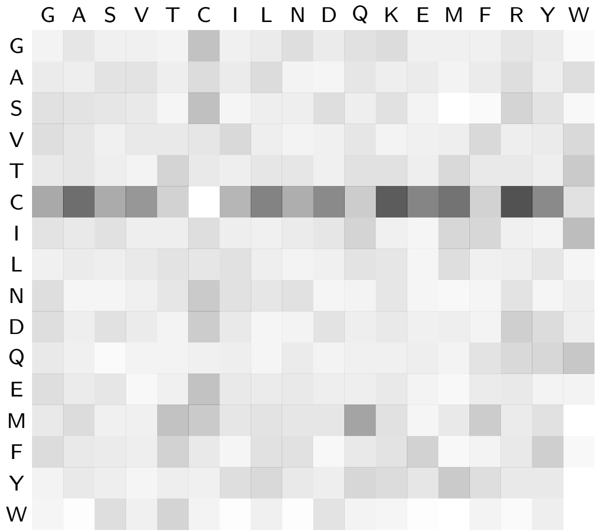

Figure A.4: Pair-wise cleavage probability for b-/y-ions from peptides that have no internal K/R/H/P and end in K/R, *i.e.,* peptides matching the sequence motif regular expression /^[^PHKR]*[KR]$/. Black indicates a probability of zero, and white indicates a probability of one.

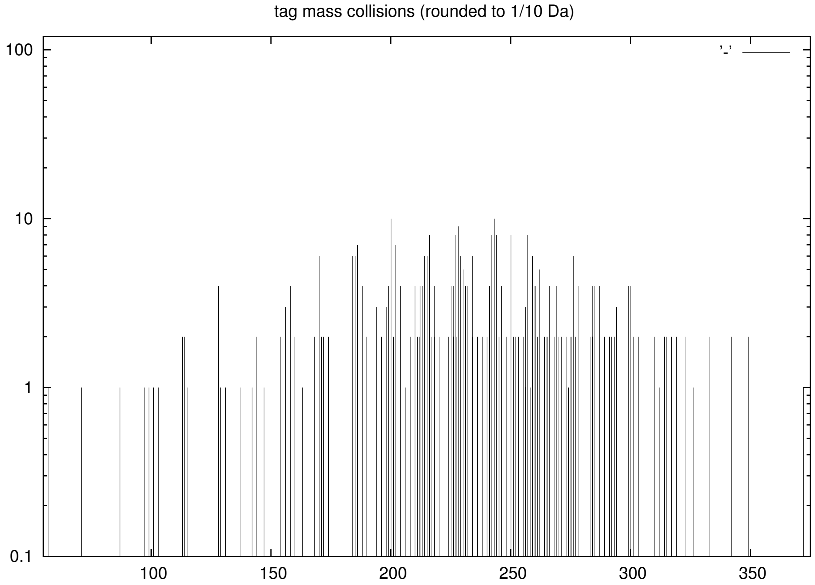

Figure A.5: Unique tag masses up to pairs (single missing peak in the b-/y-ion ladder) that collide within 0.1 Da.

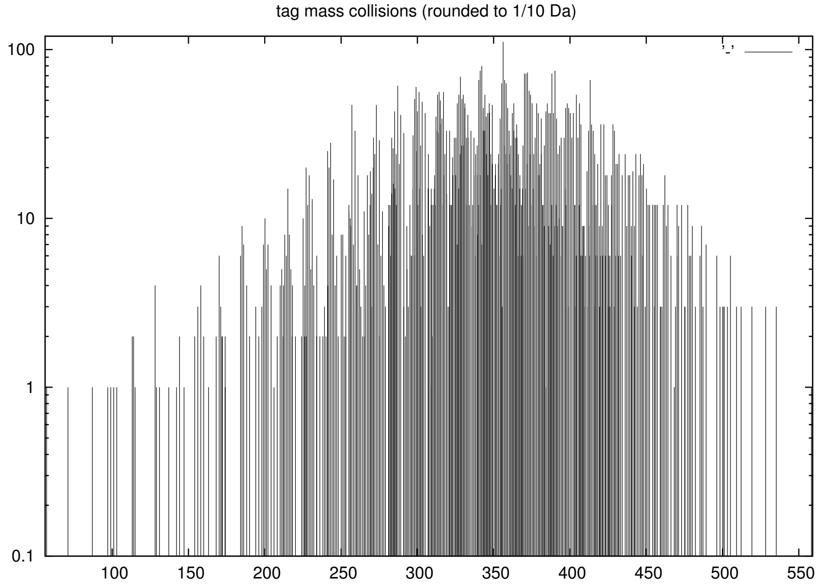

Figure A.6: Unique tag masses up to triplets (two sequential missing peaks in the b-/y-ion ladder) that collide within 0.1 Da.

```
def longest_common_subsequence_in_place( p1, p2, tol=0.5,
                                         isobaric_equivalence=false )
  return 0 if p1.length==0 or p2.length==0
  num = Array.new( p1.length ){ Array.new( p2.length )}
  p1.compute_parent_mass
  p2.compute_parent_mass
  p1_mass_N = p1.n_offset
  p2_mass_N = 0.0
  p1_mass_C = p1.mass
  p2_mass_C = p2.mass
  if isobaric_equivalence then
    p1 = p1.gsub( /[I]/, 'L' )
    p2 = p2.gsub( /[I]/, 'L' )
  end
  for i in 0...p1.length do
    p1_mass_N += AA2MASS[p1[i..i]] #mass of amino acid
    p1_mass_C -= AA2MASS[p1[i..i]]
    p2_mass_N = 0.0
    p2_mass_C = p2.mass
    for j in 0...p2.length do
      p2_mass_N += AA2MASS[p2[j..j]]
      p2_mass_C -= AA2MASS[p2[j..j]]
      if p1[i..i]==p2[j..j] and (
        ( p1_mass_N-p2_mass_N ).abs<=tol or
        ( p1_mass_C-p2_mass_C ).abs<=tol )
        if i==0 or j==0
          num[i][j] = 1
        else
          num[i][j] = 1+num[i-1][j-1]
        end
      else
        if i==0 and j==0
          num[i][j] = 0
        elsif i==0 and j!=0 # first ith element
          num[i][j] = [0, num[i][j-1]].max
        elsif j==0 and i!=0 # first jth element
          num[i][j] = [0, num[i-1][j]].max
        elsif i!=0 and j!=0
          num[i][j] = [num[i-1][j], num[i][j-1]].max
        end
      end
    end
  end
  return num[p1.length-1][p2.length-1]
end
```

Figure A.7: Longest common subsequence in-place algorithm written in Ruby.