Doctoral Thesis

Algorithms for peptide identification by tandem mass spectrometry

Author(s): Roos, Felix Franz

Publication Date: 2006

Permanent Link: https://doi.org/10.3929/ethz-a-005318769

Rights / License: In Copyright - Non-Commercial Use Permitted
Franz Roos

Algorithms for Peptide Identification by Tandem Mass Spectrometry
Algorithms for Peptide Identification by Tandem Mass Spectrometry

A dissertation submitted to

ETH Zurich

for the degree of Doctor of Sciences

presented by

Felix Franz Roos
Dipl. Natw. ETH Zurich
born February 8, 1977
citizen of Zürich and Romoos LU, Switzerland

accepted on the recommendation of

Prof. Dr. Peter Widmayer, examiner
Prof. Dr. Wilhelm Gruissem, co-examiner
Prof. Dr. Joachim Buhmann, co-examiner

2006
Abstract

Bioinformatics is a hybrid science at the interface between biology and computer science. For nearly two centuries, data acquisition in the wet lab and in the field has limited the pace of progress in biology. With the advent of high-throughput technologies such as genome sequencing, data analysis has become an additional bottleneck, and simultaneously a rich source of computational and statistical problems.

In this thesis, we focus on two problems arising in the context of high-throughput protein identification via tandem mass spectrometry. The first problem is how to identify splice sites via protein analysis, and the second how to score the quality of tandem mass spectra. We present comprehensive, tailor-made solutions to these problems, thereby integrating theoretical analysis, algorithm design, machine learning and biological knowledge.

Just a decade ago, protein identification was a fairly painstaking process which took weeks or months per protein. In the meantime, hundreds and even thousands of proteins can be identified in a single day via tandem mass spectrometry and subsequent computational fragment pattern analysis. In a typical experiment, proteins in a complex sample are cut into shorter amino acid chains (peptides). Then, the intact mass and a fragment pattern of the peptide are recorded, yielding a tandem mass spectrum. Such a spectrum is usually sufficient to determine the amino acid sequence of the peptide if the genome of the organism is available. However, in the various production steps from the DNA to the protein, processes take place that increase the diversity of proteins in ways that are often difficult to predict from the genome. This may result in protein variants whose peptide fragment patterns are usually not identified by database searches simply because the counterpart is missing in the predicted pattern database.

In the first part of this thesis, we analyze the problem of how to search spliced variants of proteins. After estimating the size of the search space,
we find that a naive search for splice sites is computationally very expensive. We analyze the problem in the I/O model which pays special attention to the different sizes and speeds of storage levels in current computers. While applying a brute force approach, we exploit the computer architecture such as to maximize the computation speed. We present a lower bound for the problem of searching spectra against a peptide database. Furthermore, we develop and analyze several algorithms that are independent of scoring schemes and that match the lower bound. To apply those algorithms, we choose a hypergeometric model as the spectrum scoring scheme and adapt it slightly for our purposes. We implement the algorithm and the statistical framework in a C++ prototype and show in a timing experiment that with increasing datasets the calculation speed asymptotically approaches the limit given by the CPU and the processor cache. Finally, by applying the software to a biological dataset, we succeed in discovering splice variants on the protein level that could not be identified in a standard protein database search.

In the second part of the thesis, we present an approach to evaluate the quality of tandem mass spectra and demonstrate how it can be used to increase the spectrum identification rate at a moderate cost. In most proteomics experiments, only a fraction of a spectrum dataset can be identified by database searches. Depending on the measurement procedure, spectra may be produced even when the peptide concentration is so low that the resulting signal is not sufficient to identify the peptide. On the other hand, datasets may also contain high quality spectra of unexpected protein variants that are absent from the predicted protein database. We develop a method to detect unidentified high quality spectra by analyzing features of identified and unidentified spectra. Among the unidentified spectra, we search for atypical cases whose properties resemble more those of identified spectra than those of unidentified spectra. We then extract these spectra, yielding a relatively small subset, and submit this to more resource-hungry searches that take into account a greater number of potential protein variants.
Zusammenfassung


aus können Proteinvarianten hervorgehen, deren Peptid-Fragmentmuster bei Datenbanksuchen normalerweise nicht identifiziert werden, und zwar schlicht deshalb, weil ihr Gegenstück in der Datenbank mit den vorhergesagten Mustern fehlt.


Wir entwickeln und analysieren mehrere Algorithmen, die unabhängig von Spektren-Korrelationsmethoden funktionieren und welche die untere Schranke matchen. Um diese Algorithmen anzuwenden, benutzen wir ein hypergeometrisches Modell als Spektren-Korrelationsmodell und passen es etwas für unsere Zwecke an. Wir implementieren den Algorithmus und eine statistische Auswertung in einem C++-Prototyp und zeigen in einem Zeitmessungsexperiment, dass sich die Rechengeschwindigkeit mit wachsender Datensatzgrösse asymptotisch der Grenze annähert, die durch den Takt der CPU und des Prozessor-Caches gegeben ist. Indem wir die Software auf einen biologischen Datensatz anwenden, gelingt es uns schliesslich, auf der Proteinebene Splice-Varianten zu entdecken, welche bei einer gewöhnlichen Protein-Datenbanksuche nicht identifiziert werden können.

Im zweiten Teil dieser Arbeit präsentieren wir einen Ansatz, um die Qualität von Tandem-Massenspektren zu bewerten, und wir zeigen, dass dieser Ansatz dazu benützt werden kann, bei nur mässigem Mehraufwand die Anzahl der identifizierten Spektren zu erhöhen. In den meisten Proteomics-Experimenten kann nur ein Bruchteil eines Spektren-Datensatzes mittels einer gewöhnlichen Protein-Datenbanksuche identifiziert werden. Einer der Gründe dafür ist, dass je nach Messverfahren Spektren auch dann gemessen werden, wenn die momentane Peptid-Konzentration zu tief ist und das daraus resultierende Signal kaum ausreicht, um das Peptid zu identifizieren. Andererseits können Datensätze auch qualitativ hochwertige Spektren von nicht erwarteten Proteinvarianten enthalten, welche in der Datenbank mit den prognostizierten Spektren fehlen. Wir entwickeln eine Methode, um nicht identifizierte qualitativ hochwertige Spektren aufzuspüren, indem wir Eigenschaften von identifizierten und nicht identifizierten Spektren analysieren. Unter den nicht identifizierten Spek-
tren suchen wir nach untypischen Fällen, die von ihren Eigenschaften her eher den identifizierten Spektren gleichen als den nicht identifizierten Spektren. Danach extrahieren wir diese Spektren, welche meist nur einen kleinen Teil des gesamten Datensatzes ausmachen, und unterziehen diese Spektren aufwändigeren Suchen, die dafür aber eine größere Anzahl von potenziellen Proteinvarianten berücksichtigen.
Acknowledgements

Scientific work is rarely done by a single person, and this is all the more true for an interdisciplinary research project such as the one underlying this thesis. I had the great opportunity of working within a network of people coming from a broad range of backgrounds and to move freely across traditional boundaries. I would like to express my gratitude to all of the following people without whose help this work would not have been possible:

First of all, Peter Widmayer, my advisor, for giving me the opportunity to work on this interdisciplinary project and for giving me a great amount of freedom;
Willi Gruissem, my co-examiner, for his support and for providing a fertile research environment within his group;
Joachim Buhmann, my co-examiner, for setting up and maintaining a fruitful collaboration;
Riko Jacob, for his patient supervision, for teaching me theoretical computer science, and for introducing me to the I/O model, which has a bright future in my opinion;
Sacha Baginsky, for his continuous support, his guidance of Jonas and me, and for coming up with biologically relevant ideas at the right point in time;
Jonas Grossmann, for his optimism, for a very memorable trip to California and for being a great PhD partner and a true friend with who I would start a new project any time again;
Bernd Fischer, for being a competent and reliable partner on a broad range of subjects, and for introducing me to the hypergeometric scoring model;
Michel, for the countless tips on C++ and many other things, and for being an excellent listener;

Joerg, also for countless tips on C++, for giving me insight into software development in general and for being Swiss in a very nice and charming way;

Mark Cieliebak, for introducing me into the project and into the group and for giving me precious feedback on the thesis manuscript;

Leon, for being a keen observer, for his constantly positive attitude and for introducing me to the PhD comics;

Conrad, for being a very nice office companion and for many helpful discussions;

Barbara, for a great deal of encouragement and for really caring;

All former and present members of the Widmayer group for their support and friendship and for many good moments in the kicker room;

All former and present members of the Gruissem group, for the friendly welcome at all times;

Alexey Nesvizhskii, for his patience and competent advice across nine time zones while writing the QualScore manuscript;

Ruedi Aebersold, for three weeks of full-time proteomics courses, for inviting me to the Institute of Systems Biology in Seattle and for the contagiously enthusiastic and visionary approach to his research field;

Stefan, Thomas and Christoph for their friendship during and beyond the statistics course;

Last but not least, em Mark, für sini unbertroffeni Loyalität, sin Optimismus und sini Geduld i guete und au schlächte Ziite.
Contents

1 Introduction .................................................. 1
  1.1 Theoretical Background .................................. 3
    1.1.1 Biological Basics ................................ 3
    1.1.2 Protein Identification by Tandem Mass Spectrometry 6
    1.1.3 Measuring the Quality of Separation ............... 12
  1.2 Summary of Results .................................... 15

2 PepSplice: Detecting Unannotated Splice-Sites .......... 21
  2.1 Introduction ........................................... 21
    2.1.1 Previous Work ................................... 22
    2.1.2 Our Contributions ................................ 24
  2.2 Strategic Considerations ............................... 25
    2.2.1 Assessment of The Number of Potential Peptides 25
    2.2.2 Gapped Alignment Algorithms ................... 28
    2.2.3 Scoring Strategies ............................... 28
  2.3 The I/O model ........................................... 30
    2.3.1 Introduction .................................... 30
    2.3.2 Basic Problem Definition ......................... 31
    2.3.3 Notation for Asymptotic Analysis ................. 32
    2.3.4 Definition of the I/O Model .................... 32
    2.3.5 General Lower Bound ............................ 33
  2.4 Algorithms without Parent Mass Constraint .......... 34
    2.4.1 Naive Row-Wise Sweep ............................. 34
    2.4.2 Stripwise Full Rectangle ......................... 36
    2.4.3 Recursive Full Rectangle ......................... 37
  2.5 Algorithms with Parent Mass Constraint .............. 40
    2.5.1 Tolerance Band .................................. 41
    2.5.2 Assumption of Sortedness ......................... 42
    2.5.3 Sorting in External Memory ...................... 43
2.5.4 Naive Algorithm for the Tolerance Band 44
2.5.5 Stripwise Tolerance Band 46
2.5.6 Recursive Tolerance Band 48

2.6 Implementation 51
2.6.1 "Experimental Lower Bound" 51
2.6.2 Generating Spliced Theoretical Spectra from the DNA 53

2.7 Statistical Analysis 59
2.7.1 Hypergeometric Score 59
2.7.2 Spectrum Preprocessing 61
2.7.3 Numerical Treatment of the Hypergeometric Score 64
2.7.4 Reversed Database to Evaluate Scores 66
2.7.5 Inaccessible Mass Bins 69
2.7.6 Ensuring the Exclusivity of the Assignment 71
2.7.7 Identification Performance Analysis 72

2.8 Biological Results 76
2.8.1 Experimental Setup 76
2.8.2 Parent Mass Deviation 77
2.8.3 Whole Genome Search 79
2.8.4 Splice Site Search 83
2.8.5 Proof of Principle: ORF At1g64790 85

2.9 Conclusion 86

3 QualScore: Dynamic Spectrum Quality Assessment and Iterative Computational Analysis of Shotgun Proteomic Data 93
3.1 Introduction 93
3.2 Experimental Procedures and Datasets 97
3.2.1 Experimental Datasets 97
3.2.2 Search Parameters 99

3.3 Computational Method 100
3.3.1 Separation of Spectra into Peak List Subsets 100
3.3.2 Spectrum Features 102
3.3.3 General Spectrum Features 102
3.3.4 Sequence Tags 103
3.3.5 Complementary Fragment Ions and Neutral Losses 106
3.3.6 Neutral Losses 108
3.3.7 Discriminant Analysis 108

3.4 Results 110
3.4.1 Overview of the Quality Assessment Method 110
3.4.2 Tuning and Testing of the Method 111
3.4.3 Evaluation of the Method 116
3.4.4 Applying the Method to a Human Lipid Rafts Dataset 118
Chapter 1

Introduction

Bioinformatics is a rapidly growing field at the interface between computer science and biology. While technological advances in DNA sequencing technology and the ensuing data explosion were responsible for most of the initial momentum, the field has undergone a lot of diversification in the meantime.

Joyce and Palsson describe this as follows in a review in Nature Molecular Cell Biology: "The completion and publication of the Haemophilus influenzae genome sequence in 1995 marked a significant phase transition in the history of biological research. The advent of whole genome sequencing and other high-throughput experimental technologies transformed biological research from a relatively data-poor discipline into one that is data rich. An important challenge that is faced by investigators today lies in interpreting these large-scale data sets and thereby deriving fundamental and applied biological information about whole systems." [Joyce and Palsson, 2006]. In Nature Biotechnology, Patterson puts it more bluntly: "[...] our ability to generate data now outstrips our ability to analyze them." [Patterson, 2003].

Many cellular components are amenable to high-throughput analysis nowadays, and for most of them, their own ‘-ome’ and ‘-omics’ term has been coined. The most established expressions are probably genome/genomics, proteome/proteomics and transcriptome/transcriptomics (the latter dealing with mRNA).

There is a great need for computational frameworks to acquire, store and process this wealth of information and to turn it into valuable knowledge that can be communicated effectively. The methods to cope with the data flood are not just waiting out there to be applied - many of them still
need to be developed. In the preface of the book “An Introduction to Bioinformatics Algorithms”, Pevzner and Jones write: “[...] Bioinformatics has affected more than just biology - it has also had a profound impact on the computational sciences. Biology has rapidly become a large source of new algorithmic and statistical problems, and has arguably been the target for more algorithms than any of the other fundamental sciences.” [Jones and Pevzner, 2004].

Biology is stretching the frontiers so fast that charting of the unknown territory has become a topic of its own. In an article in Nature Biotechnology in 2000, Aebersold, Hood and Watts use the expression ‘discovery science’: “The human genome project has catalyzed a new research method we term discovery science. Discovery science, exemplified by genome sequencing projects, enumerates the elements of a system irrespective of any hypotheses on how the system functions. As illustrated by the revolution in yeast biology and genetics as a result of knowing the complete genomic sequence, discovery science is fundamentally changing how hypothesis-driven science can be conducted. [...] As it generates data on scales of complexity and volume unprecedented in biological sciences, defying analysis by normal means of interpretation, presentation, and publication, discovery science depends on the integration of computational tools to store, model, and disseminate these exploding cascades of information. Discovery science has given rise to, and is a critical part of, a new approach we term systems biology, which involves the studying of all the elements in a biological system both before and after chemical or genetic perturbation. Ultimately, systems biology aims to establish computational models that are predictive of the behavior of the system or its emergent properties in response to any given perturbation.” [Aebersold et al., 2000].

Proteomics can be viewed as one crucial contributor to the systems biology approach, alongside with other ‘-omics’ approaches. The goal of proteomics is the systematic identification and characterization of all proteins expressed in a cell. Proteins are responsible for many cell functions, but it is not possible to predict the dynamic process of protein expression from the static code of the genome alone. To find out which proteins are expressed at what point in time, in what quantity, and in what form, proteomics experiments are needed.

The field of proteomics has benefited a lot from genome sequencing, technological improvements in mass spectrometry and peptide and protein separation techniques [Yates, 2004; Domon and Aebersold, 2006]. In shotgun proteomics, hundreds of thousands of spectra are acquired in a single experiment. Peptide identification is then performed by searching tandem mass spectra against protein sequence databases, which are
1.1 Theoretical Background

1.1.1 Biological Basics

This is a very succinct introduction to most of the molecular biology needed to read this thesis. An excellent 25 page primer on biology for computer scientists can be found in the book “Introduction to Bioinformatics Algorithms” [Jones and Pevzner, 2004].

DNA is the blueprint of an organism and is based on an alphabet of four building blocks, the nucleotides A, C, G, T. They form long strings (millions of nucleotides long) that are called chromosomes. DNA is chemically very stable because it consists of a double-helix. The longevity is important since at least one copy of the DNA has to survive throughout the whole life cycle of an organism and to the next generation. The two DNA strands are complementary to each other: A pairs with T and G pairs with C. Thus, if we know one strand, we know the other as well.

Genes are coding stretches on the DNA that can occur in the forward direction or the reverse direction on the DNA. Since the information that codes for proteins is stored in nucleotide triplets, there are also three possible reading frames in each direction.

Transcription is the process in which the messenger RNAs (mRNAs) are generated (copied) from the genes and later used as templates for protein production. In transcription, both the source and the target have an alphabet of size four. mRNAs are chemically almost identical to DNA, except that the nucleotide T is substituted by U and that they do not readily form double-helices, which makes them less stable than DNA.

Proteins are workhorses which fulfill a variety of functions in the cell. They are based on an alphabet of 20 amino acids. The amino acids in a pro-
1.1 Theoretical Background

Figure 1.1: Splicing edits mRNA and increases protein variety as compared to the classical “one gene - one mRNA - one protein” scheme. Exons (shaded) are protein coding blocks, introns (diagonally hatched) are non-coding blocks in between.

Protein are all connected in the same way via so-called peptide bonds, which form the backbone of a protein. However, each amino acid has its particular side-chain. These side-chains have very diverse chemical properties, which enables proteins to fulfill a plethora of different functions, depending on their amino acid sequence and their three-dimensional structure. A few examples of proteins in humans: hemoglobin is a carrier of oxygen in red blood cells, insulin a hormone that regulates the glucose level in the blood, rhodopsin a light sensor in the eye, trypsin an enzyme that digests proteins in the small intestine, etc...

Translation is the process of protein production. Messenger RNA (mRNA) is used as a template to produce a protein. In translation, one triplet of nucleotides codes for one amino acid. There are $4^3 = 64$ triplets but only 20 amino acids, so there is some redundancy in this scheme. In fact, several triplets code for the same amino acids. A nucleotide sequence always uniquely determines a protein sequence, but due to the redundancy, a protein sequence does usually not determine a unique nucleotide sequence. This is of little significance in nature since the flow of information is usually from DNA to mRNA to protein, but it is crucial if we want to trace back a protein to the DNA. Three out of the 64 triplets code for stop codons, namely TAA, TAG, TGA. They terminate the translation.

DNA, RNA and proteins are synthesized and read in one particular direction, never in the opposite one. The naming conventions are based on the fact that all of these molecules are polymers consisting of coupled chemical building blocks. At the start and at the end there is a chemically characteristic “coupling site” that gives the name. For proteins, the start is called N-terminus and the end C-terminus.
1.1 Theoretical Background

Splicing is a mechanism in eukaryotes (higher organisms) that contributes to the diversity of the proteome (Figure 1.1). Unlike DNA, mRNAs are editable, and in eukaryotes, sequence stretches are often cut out before mRNAs go into the protein production process. The resulting protein versions may display different behaviours and several versions may coexist. Genes consist of introns (non-coding stretches) and exons (coding stretches). Usually, introns get cut out and the remaining exons code for the protein. However, the mechanism is more flexible than that. Splice donor sites are sequence patterns in the mRNA where a gap can begin and splice acceptor sites those where a gap can end. Instead of a splice donor connecting to the next splice acceptor, it can also connect to subsequent ones, thereby potentially dropping exons as well. In the human genome sequencing project, researchers were quite baffled to find only about 30000 genes coding for something as complex as the human body. Splicing is a possible explanation how this complexity is achieved [Brett et al., 2002].

Posttranslational modifications (PTMs) also increase protein diversity. After the protein has been produced, chemical groups are attached to certain amino acid side-chains, or to the start or the end of the protein backbone. These additional molecules add mass to the protein, and they may change its structure and/or its function.

Upregulation and downregulation of genes is necessary because depending on their life situation, cells need more or less of a certain protein. Regulation mechanisms are known on many levels. mRNAs and proteins are continuously produced and degraded at different rates, depending on their sequence and on other factors. Cells can influence the rate at which mRNAs are produced from DNA, the rate at which proteins are produced from mRNA, the rate at which mRNA is degraded, and the rate at which proteins are degraded. Proteomics gives a snapshot of the protein population at a given moment, but the production and degradation rate cannot be detected in this way. The same is true for transcriptomics (gene chips, microarrays) where a snapshot of the current mRNA population is taken. Both techniques give an inventory of cell components while the turnover remains unknown.

Arabidopsis thaliana is a small flowering plant (Figure 1.2) which is an important genetic model organism. It was the first plant for which the complete genome sequence was published and it provided the biological samples for most of the datasets used in this thesis. The size of its genome is about 125 Mbp (million base pairs) and it contains more than 26000 annotated, protein-coding genes. There are five chromosomes in Arabidopsis, plus a tiny plasmid chromosome in each mitochondrion and chloroplast, two types of specialized cell compartments.
1.1 Theoretical Background

Chromatography is a chemical approach to separate a mixture of molecules such that identical molecules are grouped closely together afterwards. In high-pressure liquid chromatography, the molecules are dissolved in a solvent and then pumped through a column, which is a tube full of a solid separation material. The material is chosen such that not all molecules advance at the same speed. Some are attracted more by the solid material than by the solvent and therefore move slowly, others will be more attracted by the solvent and move faster. In the ideal case, well-separated groups of identical molecules arrive one after the other at the end of the column and can be processed further.

1.1.2 Protein Identification by Tandem Mass Spectrometry

Protein identification is a crucial step in biological experiments, and one identification technology of growing importance is mass spectrometry. The size of a protein is typically a few hundred amino acids, but the extremes range from a few dozen to several thousand amino acids. The amino acids in a protein are linked by so-called peptide bonds.

Mass spectrometry is a means to determine the mass of chemical compounds. Thanks to advances in ionization technology, large biological molecules can now also be directly analyzed. (These advances were honored with the Nobel price in chemistry in 2002.)
1.1 Theoretical Background

Tandem mass spectrometry involves multiple steps of mass selection or analysis, usually separated by some form of fragmentation. First, the masses of a mixture of intact molecules are measured. Then, a mass window is selected, the corresponding molecules retained and fragmented, and the fragments are analyzed. This fragment pattern gives information from which the identity of the molecules in the selected mass window may be derived.

In high-throughput experiments, we usually measure a complex sample of many proteins, for example a cell extract. The proteins are usually cut into smaller pieces, usually by the enzyme trypsin. These short amino acid chains are called peptides. Trypsin cuts after the amino acids lysine (K) and arginine (R), which in most cases carry a positive charge, especially in an acidic environment. The charge enables the molecule to be accelerated in the electric field of the mass spectrometer. There are far too many peptides in a complex biological sample to inject all of them at once into the mass spectrometer. Current instruments can only measure a few spectra per second, while there may be tens of thousands of different peptides in the sample. Hence a liquid chromatography column is often used to feed the mass spectrometer continuously with peptides that are separated by biochemical properties. The solvent is usually fairly acidic, which means that the proton concentration is high and most of the peptides are charged. Thus, they can be accelerated in the electric field of the mass spectrometer.

In electrospray ionization (ESI), a fine stream of droplets evaporates at the tip of the chromatography column and, due to the force of the electric field, the charged molecules accelerate and enter the instrument. After the injection, the mass/charge ratio of the intact molecules is first determined. Then a small mass/charge window is automatically chosen and only the molecules that correspond to that window are retained. Then they are fragmented and the fragments measured again. In a typical experiment, several such measurements take place every second over many hours.

The fragmentation is typically carried out by collision induced dissociation (CID), where the peptides are bombarded with non-reactive atoms, usually noble gases such as helium or argon. The weakest links in the peptide backbone are the peptide bonds, and the fragmentation is reasonably predictable. The most straightforward case to identify is a doubly positively charged peptide which breaks up in two singly charged ions (Figure 1.3). The mass of the intact uncharged molecule is denoted as $M$. If two protons are added, the mass of the molecule increases by 2 Dalton and it is doubly charged. Thus, the mass/charge ratio of the intact doubly charged peptide is $\frac{M+2}{2}$. The intensity of the bombardment is chosen such that most of the peptides suffer one single break in the backbone and are split into
Figure 1.3: Mass and charge repartition between b- and y-ions. When a doubly charged peptide is fragmented during collision induced dissociation (CID), it usually breaks at the peptide bond and yields a b-ion and a y-ion. Both of the ions carry one charge and can thus be accelerated and detected in the mass spectrometer. The protons are repartitioned unequally though. The y-ion gets two protons and one electron, while the b-ion loses one electron and is left with one delocalized positive charge. To obtain the b-ion mass, we have to add one hydrogen mass (1 Da) to the residue masses of the amino acids. To obtain the y-ion mass, we have to add the mass of $H_3O^+$ (19 Da) to the residue masses of the amino acids. The chemical processes that take place during peptide fragmentation are described in more detail by [James, 2001].
1.1 Theoretical Background

Figure 1.4: A peptide of length \( n \) yields \( n-1 \) b-ions and \( n-1 \) y-ions, \( 2(n-1) \) ions in total. The ions are complementary and add up to the parent mass.

Figure 1.5: If the \( 2(n-1) \) ions are sorted by their mass, the theoretical spectrum can easily be deduced. Due to the complementarity of the ion pairs, the spectrum has a symmetry axis. In experimental spectra, the b- and y-ions cannot be distinguished a priori.
1.1 Theoretical Background

### Table 1.1: Amino acid codes and masses.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alanine</td>
<td>Ala</td>
<td>A</td>
<td>71.03711</td>
<td>71.0788</td>
</tr>
<tr>
<td>Arginine</td>
<td>Arg</td>
<td>R</td>
<td>156.10111</td>
<td>156.1876</td>
</tr>
<tr>
<td>Asparagine</td>
<td>Asn</td>
<td>N</td>
<td>114.04293</td>
<td>114.1039</td>
</tr>
<tr>
<td>Aspartic Acid</td>
<td>Asp</td>
<td>D</td>
<td>115.02694</td>
<td>115.0886</td>
</tr>
<tr>
<td>Cysteine</td>
<td>Cys</td>
<td>C</td>
<td>103.00919</td>
<td>103.1448</td>
</tr>
<tr>
<td>Glutamic Acid</td>
<td>Glu</td>
<td>E</td>
<td>129.04259</td>
<td>129.1155</td>
</tr>
<tr>
<td>Glutamine</td>
<td>Gln</td>
<td>Q</td>
<td>128.05858</td>
<td>128.1308</td>
</tr>
<tr>
<td>Glycine</td>
<td>Gly</td>
<td>G</td>
<td>57.02146</td>
<td>57.0520</td>
</tr>
<tr>
<td>Histidine</td>
<td>His</td>
<td>H</td>
<td>137.05891</td>
<td>137.1412</td>
</tr>
<tr>
<td>Isoleucine</td>
<td>Ile</td>
<td>I</td>
<td>113.08406</td>
<td>113.1595</td>
</tr>
<tr>
<td>Leucine</td>
<td>Leu</td>
<td>L</td>
<td>113.08406</td>
<td>113.1595</td>
</tr>
<tr>
<td>Lysine</td>
<td>Lys</td>
<td>K</td>
<td>128.09496</td>
<td>128.1742</td>
</tr>
<tr>
<td>Methionine</td>
<td>Met</td>
<td>M</td>
<td>131.04049</td>
<td>131.1986</td>
</tr>
<tr>
<td>Phenylalanine</td>
<td>Phe</td>
<td>F</td>
<td>147.06841</td>
<td>147.1766</td>
</tr>
<tr>
<td>Proline</td>
<td>Pro</td>
<td>P</td>
<td>97.05276</td>
<td>97.1167</td>
</tr>
<tr>
<td>Serine</td>
<td>Ser</td>
<td>S</td>
<td>87.03203</td>
<td>87.0782</td>
</tr>
<tr>
<td>Threonine</td>
<td>Thr</td>
<td>T</td>
<td>101.04768</td>
<td>101.1051</td>
</tr>
<tr>
<td>Tryptophan</td>
<td>Trp</td>
<td>W</td>
<td>186.07931</td>
<td>186.2133</td>
</tr>
<tr>
<td>Tyrosine</td>
<td>Tyr</td>
<td>Y</td>
<td>163.06333</td>
<td>163.1760</td>
</tr>
<tr>
<td>Valine</td>
<td>Val</td>
<td>V</td>
<td>99.06841</td>
<td>99.1326</td>
</tr>
</tbody>
</table>

Two fragments (Figure 1.4). Ideally, the charge is evenly distributed between the prefix (N-terminus of the peptide) and the suffix (C-terminus of the peptide) after the split. However, since the bombardment is a stochastic process, some peptides will not be fragmented while others will undergo multiple breaks.

**Tandem Mass Spectrum**

A spectrum shows the frequency and mass/charge ratios of the detected fragments. The **peak height (peak intensity)** is proportional to the fragment count at the corresponding m/z-value. Spectra are usually depicted as a graph where the x-axis represents the mass/charge ratio m/z and the y-axis the peak intensity. Each peak has an m/z-value and an intensity. A typical high-quality spectrum is shown in Figure 1.6.

In peptide tandem mass spectra, a ladder-like pattern is usually observed (Figure 1.5). This consists of a prefix ion series and a suffix ion...
Figure 1.6: A high quality tandem mass spectrum in which the positions of the expected theoretical peaks are indicated by squares (b-ions) and diamonds (y-ions). As often seen in LTQ ion trap spectra, the signal in the low and high mass region is much weaker than in the intermediate mass region. There are also many more measured peaks than expected theoretical peaks. Some of them are due to noise, but there are also a variety of chemical effects which may lead to additional peaks.
series which are mirror images of each other, apart from a small offset due to the chemical difference between the N-terminus and the C-terminus. If the amino acid sequence is $n$ amino acids long, we expect $n - 1$ prefixes and the same number of suffixes. The peak distances within two neighboring rungs of the "ladder" reflects the mass of the amino acid that is in between. The masses of the amino acids are shown in Table 1.1. The fragmentation patterns are a bit different for each mass spectrometer, depending on the ionization technology, the fragmentation technology and fragmentation parameters, the detector and the instrument software (peak picking algorithms).

**Database Searches, Theoretical Peptides and Scoring Schemes** In database searches, experimental spectra are matched to theoretical spectra. The theoretical spectra are predicted based on peptide sequences, which are in turn predicted from genomes, therefore it is perhaps more appropriate to speak of theoretical peptides. A scoring scheme matches a theoretical peptide or rather its corresponding theoretical spectrum against an experimental spectrum. The better the match, the better the score. This matching is difficult because of the many chemical effects and the noise in spectra, but there are a number of methods that are being used quite successfully (Section 2.7).

### 1.1.3 Measuring the Quality of Separation

In spectrum identification approaches, false positive identifications are a major problem. In many of the current mass spectrometers, spectra are generated all the time, even when no peptide is fed into the mass spectrometer, which happens quite often. In database searches, there will always be a best-matching peptide, even if the match is mediocre.

On the other hand, false negative identifications are also a problem, mainly because standard peptide searches do not cover posttranslational modifications (PTMs), splicing and other mechanisms that are responsible for protein variants. We address this problem in Chapter 3, where we present a method to recover high quality spectra that were not identified in a standard database search but which deserve further identification efforts nevertheless.

In proteome research, false negative identifications are much less dangerous than false positive identifications. The occurrence of a peptide implies the occurrence of the corresponding protein, a false positive assignment thus means a wrong result. In contrast, a false negative assignment is not considered proof of the absence of a protein in the sample. However,
the identification of protein variants that were hitherto not identified is also
much more biologically interesting than the identification of well-known
proteins, so it is worthwhile to investigate false negatives too.

Developing scoring schemes to detect peptide sequences in mass spec-
tra is challenging, and the currently available scores usually need re-evaluation
to derive a statistical confidence for a match. In general, we want as many
identifications as possible, but they should also be as reliable as possible.
The question is how to find a good tradeoff between the two conflicting
goals. There are two standard approaches to analyze this, the ROC (re-
ceiver operator characteristic) and the recall-precision curve. Both of them
have a common basis.

Classification problems arise in many applications ranging from med-
ical tests (where healthy and diseased patients must be distinguished) to
spam filters (where spam mail must be distinguished from regular mail).
Here, we focus on the case of two classes where one discrimination score
is used to distinguish them. Ideally, one would want a score to separate
the two classes without overlap such that a threshold (cutoff) can be deter-
mined: all data points above belong to one class and all below to the other
class. In reality, a discrimination score is not always able to separate the
classes perfectly and we observe an overlap which means that misclassifi-
cations occur (Figure 1.7).

In the case of such an overlap, we may still determine a cutoff, but the
choice of the cutoff depends on the application:

Example: Say a hospital uses an HIV test that is based on a color
reaction where dark blue means infected (positive) and white
means healthy (negative). Shades of light blue are more or less
inconclusive, they fall into the overlap. If we screen blood do-
nations, any donations that produce the slightest shade of blue
will be discarded because we cannot afford the risk of infect-
ing a blood recipient. We use a conservative cutoff because we
primarily want to deliver clean blood to the health care system.
The perspective is different for a doctor whose patient has just
taken a HIV test and wants to know the result. An inconclusive
result must be investigated carefully because a false positive
test result would cause the patient a lot of anxiety, and he may
get unnecessary medication with heavy side-effects. In case of
a false negative result, he may not get the necessary treatment
and possibly pass the infection on to other people.

If we have a training dataset where we know to which class each data
point belongs, we can characterize the separation performance. For any
Figure 1.7: Depicted are the score histograms of two classes where we know for each data point to which class it belongs. The right curve represents the class of positives, the left curve the class of negatives. The discrimination score separates them fairly well, but there is some overlap nevertheless. We show an arbitrary discrimination score cutoff (vertical line) to illustrate how the four classes are derived. Most datapoints are correctly classified (TP, TN), but some are also misclassified (FP, FN). The actual choice of a cutoff strongly depends on the application.

Given cutoff, we can determine the number of correct and incorrect classifications below and above the cutoff (Figure 1.7). This gives us the following four classes: true positives (TP), false negatives (FN), true negatives (TN) and false positives (FP). These four classes sum up to the total number of data points. We can repeat this evaluation for several cutoffs and for each of them, we obtain the four values again.

Since four dimensions are impractical to visualize, one usually derives two ratios from them and plots the ratios against each other. There are three types of ratios which are commonly used, and each focuses on differ-
1.2 Summary of Results

The focus of this thesis are computational methods that allow peptide identification via tandem mass spectra. While current methods are able to routinely identify regular peptides and their corresponding proteins, they are less successful at identifying protein variants due to splicing, posttranslational modifications, chemical modifications, DNA sequencing errors and other phenomena that are sometimes difficult to predict from the genome.
1.2 Summary of Results

Figure 1.8: The receiver operator characteristic (ROC) of the data depicted in Figure 1.7. The ROC describes the separation with equal focus on both classes. The sensitivity describes the fraction of the positive class that is above the cutoff, the specificity the fraction of the negative class that is below the cutoff. An ideal separation would give a dot at sensitivity=1 and 1-specificity=0, in the upper left corner.

Figure 1.9: The recall precision curve of the data depicted in Figure 1.7. This approach focuses on the positive class: The recall (=sensitivity) describes the fraction of the positive class that is above the cutoff, while the precision describes the fraction of true positives among all the data points above the cutoff. An ideal separation would give a dot at recall=1 and precision=1, in the upper right corner.
We investigate such variants, especially peptides spanning across a splice site, which pose a computational challenge. We approach the problem from various angles and blend theoretical analysis with algorithm design, statistics and biology.

In Chapter 2, we study the problem of detecting splice sites in a large spectrum dataset. Many of the above-mentioned protein variants can be optionally searched using existing search tools, but splice site searches are a notable exception. Previous theoretical work on splice site searches has been mainly concerned with the identification of one single spectrum on a spliced genome. This approach does not transfer well to large datasets. In a more biologically oriented publication, splice site searches on large datasets of human proteins have been investigated. The authors use a commercial search tool that they developed in-house and they do not provide any computational analysis [Colinge et al., 2004].

Upon analyzing the size of the search space for spliced peptides on a genome, we find that the number of peptides is linear in the size of the genome and linear in the gap size of the spliced peptide. A naive search for this would be computationally expensive.

We tackle this problem by developing algorithms that are optimized with respect to the number of required input/output operations. In the I/O model, data transfer is considered as the bottleneck that needs to be optimized. As opposed to the widely used RAM model, the I/O model is based on the following three components: a CPU, a fast memory $M$ of limited size and a slow disk of unlimited size. (The memory takes the role of a cache in this context.) Data transfer occurs block-wise, at a cost of one I/O per block of $B$ elements. In this model, it is advantageous to order accesses such that data in the memory can be reused, instead of reloading them at great expense from the disk. Furthermore, it is more efficient to use as many elements as possible in a loaded block, not only a few among them.

We present a lower bound for the problem of scoring a spectrum dataset against a peptide database using the minimum number of I/O operations. Furthermore, we present several algorithms that match the lower bound. The algorithms are designed such as to allow for arbitrary spectrum scoring schemes. The first class of algorithms compares all elements of one dataset against all elements of the other dataset. The second class takes into account the parent mass constraint, meaning that a match between a theoretical peptide and an experimental spectra is only possible if they have (almost) the same parent mass. We further distinguish three types of algorithms: naive, cache-aware and cache-oblivious. The naive algorithms do not attempt to optimize the memory (cache) usage. The cache-aware
algorithms use a blocked calculation strategy that optimally exploits the memory, but they require knowledge about the size of the memory. The cache-oblivious algorithms usually have a slightly reduced efficiency with respect to fully optimized cache-aware algorithms, but they do not need any knowledge about the memory size and are asymptotically optimal nevertheless.

In a timing experiment, we demonstrate that with increasing sizes of spectrum datasets, the running time of the implementation asymptotically approaches the speed limitation given by the CPU clock rate and the processor cache latency. This means that our I/O optimization efforts have been successful both in theory and in practice.

To apply our algorithms to spectra, we need a scoring scheme. We introduce a hypergeometric model by Sadygov and Yates [2003] and adapt it for our purposes. The performance of the scoring scheme depends on how well it captures the essence of the signal. In order to measure this performance and to verify the search results, we do not only search the regular peptide database but simultaneously also a database of the same peptides with their sequences reversed. We assess to what extent the scoring scheme gets misled by the reversed sequences. We find that at high scores, the best hit for a spectrum is virtually always a forward sequence while at low scores, the best scoring hits have a ratio of almost exactly 1:1 between forward and reversed sequences. Since the ratio of forward to reverse sequences in the database is also 1:1, this indicates that the reverse sequence approach works. We apply a precision-recall analysis to estimate the overall identification performance and the precision at a given score cutoff. Our approach does not make assumptions about the distribution of scores in a typical spectrum dataset, unlike other approaches such as PeptideProphet [Keller et al., 2002]. These assumptions may be undermined if for example only a subset of high scoring spectra is analyzed, which is indeed the case in our work. Our approach is therefore more robust in such a scenario.

The algorithms, the scoring and the statistical evaluation are available in a C++ prototype that compiles and runs on Linux, Windows and Mac OS X without any modifications. We reach a search speed of around 30 spectra per second for a tryptic Arabidopsis whole genome search on a mass market dual-core CPU computer. This rate is much higher than we expected, and it exceeds the acquisition speed of current instruments by an order of magnitude.

Using the implementation, we analyze a dataset of 1.4 million tandem mass spectra. In three successive searches, we increase the search space step by step. First, we search the standard Arabidopsis protein database
1.2 Summary of Results

provided by arabidopsis.org, including commonly known contaminant proteins. Thereafter, we search the unspliced whole genome to see whether we obtain additional identifications, indicating that the protein prediction from the genome may be slightly erroneous in some cases. Indeed, we identify 736 spectra that cannot be matched to the protein database, but only to the DNA. In order to do a splice site search, we extract a subset of 98437 spectra using a quality scoring tool, which is described in Chapter 3. Among these, we successfully identify 399 spectra of spliced peptides that contradict the gene prediction underlying the standard Arabidopsis protein database. As a proof of principle, we analyze one particularly striking cluster of identifications in the gene At1g64790 in detail and find that it contradicts current gene predictions in a convincing and consistent way.

We conclude that splice site searches allow to find unexpected protein variants and can thereby provide valuable contributions to the refinement of genome annotation. Our I/O-optimized algorithms can not only be used for splice site searches, but for any other type of peptide database search, such as protein and whole genome searches. Posttranslational modifications could be easily included too.

In Chapter 3, we present a spectrum quality scoring method. Our research originally started with a peptide de novo sequencing project. After testing a heuristic scoring coupled with a dynamic programming approach [Grossmann et al., 2005], we shifted to a probabilistic scoring approach based on a Hidden Markov model [Fischer et al., 2004, 2005]. Realizing that de novo sequencing heavily depends on high quality data, we set out to assess spectrum quality on large datasets.

Typically, less than half of all spectra can be identified in database searches, depending on the technology used to feed tandem mass spectrometers. There are two main reasons for this: First, spectra are usually acquired throughout the measurement process, even when the peptide concentration is very low, thus generating spectra that cannot be identified due to a lack of signal. Second, datasets often contain high quality spectra that cannot be identified because the peptide is not in the database. Regular database searches usually do not take into account posttranslational modifications, rare splice variants and other biological phenomena because searching for them is computationally demanding. We investigate if the identification rate can be increased at a reasonable cost, namely by concentrating on the subset of unassigned high quality spectra. We learn what constitutes a high quality spectrum by using the regular database search results as a training set. We assume that unidentified spectra whose features resemble more those of identified spectra potentially contain untapped in-
formation of biological interest. We therefore extract them from the dataset for further analysis. The size of the extracted dataset is typically an order of magnitude smaller than the original dataset, which saves search time in the subsequent computationally more demanding analyses. We show that by extracting such spectra and re-analyzing them, we can increase the total number of peptide identifications by more than 10% and, most importantly, identify peptide variants that otherwise cannot be found by regular database searches.
Chapter 2

PepSplice: Detecting Unannotated Splice-Sites

2.1 Introduction

In the last few years, tandem mass spectrometry has become a standard tool for high-throughput protein identification. Peptide tandem mass spectra are usually identified by searching them against protein databases, which in turn are mostly predicted from the DNA. Genes in eukaryotes (higher organisms) usually consist of intermittent stretches of protein coding DNA (exons) and DNA that is not translated into protein (introns). The introns are removed at the mRNA level during splicing. Usually, introns and exons are first predicted from the DNA when the genome is available and later refined by additional experimental information, mostly from the mRNA level. In practice, this means that if the genome of an organism is sequenced, a standard protein database search is able to identify most tandem mass spectra that contain enough signal to be identifiable (compare Section 2.7.4 for more information about challenges in spectrum identification). However, since gene prediction is not always perfect, direct searches on the DNA are also used as complementary approaches, for example in the PeptideAtlas project [Desiere et al., 2005, 2006]. In such whole genome searches, the DNA is translated to protein using all six possible reading frames. The results of whole genome searches may be used to refine the genome annotation, which is based on gene prediction models. Even though many genomes are now available, some for years, their annotation is still evolving because gene prediction is a challenging task.

A broad range of gene prediction methods are reviewed in [Mathe et al.].
2.1 Introduction

One well-known probabilistic gene prediction algorithm for the human genome is GeneScan [Burge and Karlin, 1997], which uses descriptions of the basic transcriptional, translational and splicing signals, as well as length distributions and compositional features of exons, introns and intergenic regions. Eugene is a gene prediction algorithm for Arabidopsis that uses similarity searches within and between genomes [Foissac et al., 2003]. Different gene prediction tools may yield contradictory results, which calls for experimental data to determine the correct prediction. Sometimes, several seemingly conflicting predictions are correct. In the case of alternative splicing, various gene products may stem from one and the same gene and it may be neither possible nor sensible to disprove a prediction by experimental data.

The number of unspliced theoretical peptides in a database is linear in the number of amino acids or nucleotides in the database (Section 2.2.1). This is true for both protein and DNA databases. However, DNA databases of higher organisms are usually a lot larger than their corresponding protein database because of so-called non-coding DNA that lies in between genes, and which is not contained in protein databases. If the DNA database is used to generate spliced peptides, the number of peptides that can be derived from the DNA becomes much larger than in the unspliced case. The number of peptides is then linear both in the size of the database and in the length of the gap (more detailed calculation in Section 2.2).

Splicing is a mechanism that is ubiquitous in higher organisms (eukaryotes) and is a contributor to protein diversity, alongside with posttranslational modifications. In splicing, stretches of mRNA between splice signal sites are cut out in various ways, depending on cell regulation processes. Several versions of proteins coded by the same gene may coexist in the proteome. One gene may also contain several splice sites, as illustrated in Figure 2.1. Splicing has been studied extensively at the mRNA level, but has received surprisingly little attention at the level of peptide tandem mass spectrometry. Since the throughput of proteomics experiments has increased rapidly due to technological advances and automatization, proteomics data may become a valuable source for genome annotation. In fact, spectrum datasets that are collected for other purposes might well be used additionally for genome annotation, without requiring separate experiments to be done.

2.1.1 Previous Work

For the case of spliced peptide searches, theoretical calculations on search complexity and statistical issues have been published by [Chen 2001], es-
2.1 Introduction

Figure 2.1: An illustration of gene prediction including splicing from the Arabidopsis chromosome 2. A tiny window of 6 kilobasepairs out of approximately 20 Megabasepairs is shown. Three genes are shown, one on the forward and two on the reverse strand. Filled arrows represent exons, the connecting lines introns (gaps). The figure was generated using the FlagDB tool.

... especially on the risk of random matches as a function of peptide length and genome size. The authors provide algorithms to find spliced peptides on the genome, but these are not optimized for real life data. In their simulations, they only use a tiny genome fragment of 123 kilobasepairs whereas the Arabidopsis genome has 125 Megabasepairs, and they do not give any indications about the search time of their approach. Moreover, they focus on the case where one experimental spectrum is given and then search all matching spliced peptides. In practice, a dataset contains tens of thousands or even hundreds of thousands of experimental spectra that can be searched simultaneously.

Another group of authors worked with real data, but they used their in-house commercial search tool OLAV/Phenyx [Colinge et al., 2004] and give barely any algorithmic information nor do they indicate how much search time was actually needed. They work on tandem mass spectra of human proteins, use first-order Markov chains to predict splice donor and acceptor sites and limit the search area to regions around whole genome hits, probably because of the enormous size of the human genome.

During the last decade, tandem mass spectrometry has seen tremendous technological progress. Current instruments are faster and more sensitive and precise than ever before. A typical ion trap instrument can acquire hundreds of thousands of spectra per day, at a detection threshold of a few dozen molecules per compound. Computer hardware and search algorithms have only partially kept up with this pace of development. In many research facilities, common database searches using commercial tools such as SEQUEST [Yates et al., 1996] are often so slow that they represent a major bottleneck even for normal protein searches or genome searches. Other tools such as X!Tandem [Craig and Beavis, 2004] and MASCOT [Perkins et al., 1999] are faster, but according to a recent study, the combination of SEQUEST and PeptideProphet still seems to provide the highest identification performance [Kapp et al., 2005]. (Their comparison is based on a ROC...
2.1 Introduction

For most tools, some information about the scoring is available, but in the respective publications, little or no information is given about the algorithmic strategies that they use to search the data.

2.1.2 Our Contributions

We present database search algorithms to detect splice sites in large spectrum datasets. The algorithms also work for classical database searches, namely ordinary protein database searches and DNA database searches, including posttranslational modifications. We pay special attention to general hardware properties, mainly the fact that there are several storage levels of various speeds and sizes, such as the processor cache, the RAM, the hard disk and possibly others.

Algorithms that take into account several storage hierarchies are called cache-aware, or cache-oblivious if they do not require parameters about the properties of the storage levels. Our algorithms focus on the optimal order in which the scorings should take place to exploit the hardware capabilities as much as possible.

For the scoring, we use a hypergeometric method that was published recently and which has a favorable sensitivity-precision tradeoff [Sadygov and Yates, 2003]. To obtain confidence values for the scores, we do all searches both on the standard database and also on reversed sequences for control purposes. The scores of the reverted database are used as a reference background score distribution against which the standard database hits are compared. We use a precision-recall curve to extract a set of high confidence identifications.

We implemented all algorithms, including the scoring and the precision-recall analysis, in a C++ software prototype that compiles and runs on Linux, Windows and Mac OS X without any modifications.

First, we applied the software to 1.4 million tandem mass spectra that were labeled as doubly charged by the instrument software. The underlying biological sample originated from an Arabidopsis plant cell culture. We searched them against an Arabidopsis protein database and against the entire unspliced Arabidopsis genome. Both searches together took half a day on an Intel dual-core CPU. This corresponds to a turnover of more than 100,000 spectra per hour. For comparison: The Thermo Finnigan LTQ mass spectrometer produces tandem mass spectra at a rate of less than 1

\footnote{The resolution of ion trap tandem mass spectra currently does not allow to determine the charge state of multiply charged parent ions. Therefore, the instrument software saves two versions of the same spectrum, one assuming a charge state of two, the other assuming a charge state of three. We use only the charge two version because the pattern is more regular, the fragments are mostly singly charged.}
2.2 Strategic Considerations

10,000 per hour. This means that using our algorithm, a single desktop computer is able to do Arabidopsis whole genome searches for more than one mass spectrometer. In the unspliced DNA database search, we identified 736 spectra that were not previously found in the protein database search and which therefore contradict the gene prediction model underlying the protein database.

By applying our in-house spectrum quality scoring tool (Chapter 3) to the 1.4 million spectra, we extracted nearly 100,000 unassigned high quality spectra. We applied our splice site search algorithm to these and searched the whole Arabidopsis genome. We used dinucleotide constraints for donor/acceptor sites and searched the entire genome, not only regions around whole genome identifications. We found 399 spectra with splice sites that were not detected in the protein database search and which thus contradict the gene prediction that underlies the protein database.

By this integrated approach, we demonstrate that shotgun proteomics experiments can indeed be used to refine gene models. Moreover, the implementation does not require more than a single mass market desktop computer thanks to careful algorithm design.

2.2 Strategic Considerations

In our search for spliced peptides, we are given a set of unidentified peptide tandem mass spectra and assume that some of them are best explained by a gapped DNA stretch on the genome (i.e., the corresponding peptide) under some scoring scheme. The best matches per spectrum must be found and reported. Theoretically, several gaps in one peptide are possible, but we restrict ourselves to one gap since peptides are relatively short and the probability that two occur within such a short distance is not very high (based on exon length distributions plotted in [Deutsch and Long, 1999; Sakharkar et al., 2002]). A spliced peptide consists of a prefix, a gap and a suffix (Figure 2.2). Allowing for a gap in the spectrum greatly increases the number of allowed matches as compared to classic whole genome searches.

2.2.1 Assessment of The Number of Potential Peptides

With a view to choosing computational strategies, we make a preliminary assessment of the number of potential peptides in a search. We do this for three different search spaces: protein databases, whole genomes and spliced whole genomes. Our aim is to identify the most relevant tendencies.
### 2.2 Strategic Considerations

**Protein Databases** The number of unspliced peptides in a protein database is equal to the number of tryptic cleavage sites (plus the number of protein ends), assuming that no cleavage site is missed in the enzymatic digestion process.

Let the total number of tryptic peptides be $t$. We can crudely estimate $t$ by assuming that $\frac{2}{20} = \frac{1}{10}$ of all amino acids in the database are tryptic cleavage sites (this number depends on the amino acid frequencies of K and R in the organism of interest). The number of tryptic cleavage sites usually far exceeds the number of protein ends, so we do not count the protein ends. (Cutting a protein $t$ times yields $t + 1$ pieces, but we approximate this by $t$.) Let $a$ be the number of amino acids in a protein database. Then, the number of tryptic peptides is $t = \frac{a}{10}$, so the number of peptides is linear in the length of the database.

**DNA Databases** For DNA databases, three nucleotides are needed per amino acid, but then there are three reading frames too, in two directions. Let $n$ be the number of nucleotides in a database. If the genome has a length of $n$ nucleotides $\{A, C, G, T\}$, $a = 2 \cdot 3^\frac{n}{3} = 2n$, which means that $t = \frac{a}{10} = \frac{n}{5}$. These are obviously hypothetical peptides that only need to be searched if the gene structures are not known with sufficient certainty.

**Spliced DNA Databases** If splice sites are allowed, this adds another dimension. There is a prefix, a suffix and a gap in each spliced peptide (Figure 2.2). We use four constraints as described in Section 2.6.2. The prefix start and the suffix end are tryptic. The gap start must be a dinucleotide GT, which occurs approximately every $4^2 = 16$ nucleotides, and the gap end must be a dinucleotide AG which occurs also roughly every $4^2 = 16$ nucleotides. As an approximation, we can assume that the frequency of prefixes and tryptic peptides is the same, so there are $t$ prefixes in a database. The only difference is that the prefix end is not the next tryptic cleavage site but the next dinucleotide GT. The prefixes and suffixes have the same constraints. This means that their frequency is the same and there are approximately $t$ suffixes too.

Prefixes and suffixes can be combined between any of the three reading frames, but only within the same reading direction, and the suffix start must be located after the prefix end. Furthermore, the combined length must be a multiple of three. Starting from a prefix and going along the DNA, we can expect to find a suffix to that prefix about every 10 nucleotides (based on simplified tryptic cleavage and only in one reading direction), but only every third of these will give a multiple of 3 in the total number of nucleotides. We thus expect a valid suffix every 30 nu-
2.2 Strategic Considerations

Figure 2.2: A complete spliced peptide consists of three components: a prefix, a gap and a suffix. These components are delimited by four points: the prefix start (PS), the prefix end (PE), the suffix start (SS) and the suffix end (SE). We used the following constraints: tryptic prefix start and suffix end, and gaps must start with GT and end with AG.

cleotides. The number of combinations per prefix depends on the gap length: if the gap length is \( g \), the number of prefix-suffix combinations is roughly \( x = \frac{5}{10} t = \frac{2}{30} n = \frac{9}{105} \).

**Additional Effects** There are at least three more effects that have an influence on the number of potential peptides. The first two effects increase the number of peptides in all three types of searches. The last one decreases the number of peptides in genome based searches.

1. Missed tryptic cleavages increase the number of peptides in all three search types. In protein database searches, the number of missed cleavages is usually restricted: [Kapp et al., 2005] for example allow only two per tryptic peptide. Currently, we do not restrict the number of missed cleavages in our implementation at all. By doing so, we could reduce the number of peptides that are searched and thereby increase the search speed, but at the risk of missing peptide variants with missed cleavage sites.

2. In the searches, we always allow for a very frequent amino acid modification (methionine oxidation, M16). This means that methionine can occur in two different masses, and peptides containing \( m \) methionines occur thus in \( 2^m \) versions. However, methionine is relatively rare, its frequency is usually below 2%.

3. In the whole genome search and the splice site search, the occurrence of stop codons which terminate proteins reduces the number of peptides. Peptides must not contain stop codons, and the longer a the-
2.2 Strategic Considerations

2.2.2 Gapped Alignment Algorithms

Algorithms for gapped alignments between nucleotide and protein sequences are long established. One of the early pioneers in biological sequence alignments in general was the Needleman-Wunsch algorithm [Needleman and Wunsch, 1970] that provided an ungapped, weighted sequence alignment of nucleotides or proteins in $O(mn)$ with respect to space and running time ($m$ and $n$ are the lengths of the sequences). A decade later, this algorithm was extended by Smith and Waterman [Smith and Waterman, 1981] to include gaps. Another decade later, the Needleman-Wunsch algorithm was approximated by the BLAST algorithm (Basic Local Alignment Search Tool), which uses a heuristic that emphasizes speed over sensitivity and makes large database searches feasible in practice [Altschul et al., 1990, 1997].

However, the peak information in peptide tandem mass spectra differs markedly from the classical sequence information. Noise peaks usually allow for a variety of sequence interpretations. Moreover, there is the additional constraint that the parent mass of a spectrum and the parent mass of the corresponding peptide must be equal except for some mass tolerance. In view of these differences, the algorithms used for sequence alignments cannot be transferred easily to alignments between peptide tandem mass spectra and protein or nucleotide sequences.

2.2.3 Scoring Strategies

In the following, we discuss computational strategies to match a set of tandem mass spectra against a set of peptides. While deciding on a brute force approach, we optimize the performance in terms of input/output operations.

Peptides and experimental spectra need only be scored if their parent mass is equal, allowing for some measurement error. Consequently, we can sort both the peptides and the experimental spectra by parent mass and then score them in a merge-sort like fashion (a bit like a zipper). However, the sheer number of potential spliced peptides makes this approach impractical: For the Arabidopsis genome and a gap length of 3000 nucleotides, one would need to store and sort roughly one terabyte of data.

\[^{3}\text{We counted } 3.75 \cdot 10^{10} \text{ spliced peptides in the Arabidopsis genome, and if we store a prefix start, a prefix end, a suffix start and a suffix end for each peptide as pointers on the}\]
which would be challenging on current computers. Other genomes are even larger and would consequently require more space.

There are two alternative strategies: In the spectrum-centered approach, we focus on one spectrum at a time and search all peptide database entries that match it, but without making the peptides explicit since this needs too much storage space. In the peptide-centered approach, we focus on one peptide in the database and search all spectra that match it. In both cases we can pre-sort the entries and thus speed up the retrieval:

**Spectrum-centered approach**  Instead of making all spliced peptides explicit, we could index (or just sort) prefixes and suffixes separately by mass and then retrieve combinations that match the parent mass. This needs a lot less space than explicitly enumerating all possible prefix-suffix combinations. Unfortunately, while the number of nucleotides of a peptide is a multiple of 3 and its mass can be calculated very easily, the splice site can cut right across a triplet, and the mass of a prefix or a suffix cannot be determined until the remainder of the split triplet is known. Since there are only 64 triplets, one could build sub-indexes corresponding to all cases of triplet splits. But the gap size is also limited because the prefix and the suffix must be located near each other on the DNA. An index would thus need to take into account the parent mass, the split triplet and the neighborhood on the DNA.

**Peptide-centered approach**  On the other hand, sorting tandem mass spectra by parent mass is straightforward. For large datasets and large mass tolerances it turns out that for any parent mass of a spliced peptide, there are virtually always spectra to be scored. Considering this, we can enumerate spliced peptides on the fly and for each peptide score all spectra whose parent mass lies within the parent mass tolerance interval.

**Scoring**  The scoring may also contain algorithmic optimization potential. Several approaches with sequence tag-based filtering have been described

---

3 In the case of the popular ion trap mass spectrometer, the parent masses range approximately from 1000 – 3000 Dalton (interval length 2000 Dalton) while the tolerance interval is in the order of 1 – 10 Dalton. This means that on average, the number of spectra per parent mass is approximately \( \frac{2000}{10} \) to \( \frac{2000}{2} \). A typical experiment yields tens of thousands or even hundreds of thousands of spectra, the average number of spectra per parent mass is thus in the order of 50 to 500.
2.3 The I/O model

2.3.1 Introduction

In current computers, the speed of the CPU is often so high that the bottleneck are not the calculations themselves but the transfer of data between the storage and the CPU. The storage is not uniform but consists of several levels whose size and speed are usually inversely correlated. The following storage levels are typical for current hardware: the CPU registers, the CPU cache (of level 1, 2 and sometimes even 3), the memory/RAM, and
2.3 The I/O model

Table 2.1: The size and speed of storage levels is inversely correlated. The figures are approximate values for current mass market desktop computers.

<table>
<thead>
<tr>
<th></th>
<th>size [bytes]</th>
<th>access time [clock cycles]</th>
</tr>
</thead>
<tbody>
<tr>
<td>registers</td>
<td>$10^{1.5}$</td>
<td>$10^0$</td>
</tr>
<tr>
<td>processor cache</td>
<td>$10^6$</td>
<td>$10^{0.5}$</td>
</tr>
<tr>
<td>RAM</td>
<td>$10^8$</td>
<td>$10^{1.5}$</td>
</tr>
<tr>
<td>hard disk</td>
<td>$10^{11}$</td>
<td>$10^7$</td>
</tr>
</tbody>
</table>

the hard disk. The typical relative speed of current mass-market components is summarized in Table 2.1.

All storage levels allow accesses to arbitrary positions, so-called random accesses, but strongly fragmented access patterns are usually not very efficient. This effect is especially pronounced for the hard disk because there, a physical adjustment of the reading head is necessary, and once arrived, it has to wait until the disk has rotated to the position where the reading process starts. The average hard disk currently works at a speed of 7200rpm, so one full rotation of the disk takes 8ms.

In the design of most algorithms, storage hierarchies are not routinely taken into account. However, when calculations on large datasets need to be carried out, the order of the calculations has a great impact on the efficiency of the data delivery. If elements can be reused within not too many operations, they generally remain in the cache and the delivery is efficient. If many other operations happen in between the reuses, it may be impossible to keep in a cache of limited size all elements that will be reused. However, reloading elements from slower storage negatively affects the overall performance because the CPU is often idle, waiting for data to arrive. By cleverly managing the order of calculations, unnecessary reloading operations can be avoided and the overall efficiency thereby improved.

2.3.2 Basic Problem Definition

We look at the problem where two datasets $X$ and $Y$ are to be scored against each other, either all against all elements or some against some. The dataset $X$ has $N_X$ elements and the dataset $Y$ has $N_Y$ elements. Let $k$ be the number of scores. If all against all elements are scored, $k = N_X N_Y$. If only a subset of scores is calculated, $k \leq N_X N_Y$. In the realm of tandem mass spectra, $X$ represents experimental spectra, and $Y$ represents theoretical peptides.
2.3 The I/O model

2.3.3 Notation for Asymptotic Analysis

The classical $O$, $\Omega$- and $\Theta$-notations are slightly modified in this chapter to encompass multi-parameter expressions. For $g := X \rightarrow \mathbb{R}^+$:

$O(g) := \{ f : X \rightarrow \mathbb{R}^+ | \exists c \in \mathbb{R}^+: \forall x \in X : 0 < f(x) < c \cdot g(x) \}$

$\Omega(g) := \{ f : X \rightarrow \mathbb{R}^+ | \exists c \in \mathbb{R}^+: \forall x \in X : c \cdot g(x) < f(x) \}$

$\Theta(g) := O(g) \cap \Omega(g)$

Instead of using the notation for one single variable, we use it for several variables at once.

2.3.4 Definition of the I/O Model

The I/O model deals with the input/output complexity of algorithms in settings where the transfer of data is the bottleneck.

In the definition of the I/O model by [Aggarwal and Vitter, 1988], it is based on three components: a CPU, a fast memory of limited size and a slow disk that is infinitely large. The memory has a capacity of $M$ elements and the disk is organized in blocks having a capacity of $B$ elements each. On the disk resides a dataset with $N$ elements where $B \leq M \leq N$. ‘Memory’ and ‘disk’ are used as generic terms that can be applied to any two adjacent storage levels (processor cache/RAM, RAM/hard disk, ...). The CPU has only access to data in the memory. If they are not present in the memory, a page fault occurs and the data need to be loaded from the disk into the memory. Transfer of data between disk and memory takes place in entire blocks. One such transfer operation is called an I/O, and the number of I/Os is used as the performance measure. We further assume that $M$ is a multiple of $B$, which avoids rounding problems for expressions such as $\frac{M}{B}$. Contrary to the definition by [Aggarwal and Vitter, 1988], we do not assume that $N \geq M$, so $N_X$ and $N_Y$ may both fit into the memory.

A decade later, [Frigo et al., 1999] introduce the concept of cache-oblivious algorithms. The advantage of those algorithms is that no hardware dependent variables need to be tuned to achieve asymptotic optimality and that they can therefore be used on a wide variety of machines with little or no adaptation. In the I/O-model, these hardware dependent variables are $B$ and $M$. [Frigo et al., 1999] further introduce an ideal-cache model to study
2.3 The I/O model

the cache complexity of algorithms, especially lower bounds: “The ideal cache uses the optimal off-line strategy of replacing the cache line whose next access is furthest in the future, and thus it exploits temporal locality perfectly.” The difference between the ideal-cache model and cache replacement policies such as the least recently used strategy (LRU) is that the latter need to work online. In this context, Panagiotou and Souza [2006] discuss how to measure the performance of paging.

Available Memory

As we will see later, the memory is always occupied by members of both datasets in our algorithms since scorings are only meaningful between one element of each. For cache-oblivious algorithms there are many results that require a tall-cache assumption such as \( M > B^2 \). This assumption is unnecessarily strong for our results. Still, we do need a constant number of blocks in main memory, so we assume that \( M \geq 7B \). (In practice, \( M \gg B \), i.e., the memory is usually considerably larger than one block.) At the same time, we define the so-called available memory as

\[
M' = M - 5B
\]

(2.1)

The available memory is almost as large as the full memory, but we allow one block to do calculations and two further blocks at the margins of the dataset \( X \) and \( Y \). The blocks at the margins might be only partly filled because the elements in \( X \) and \( Y \) do not necessarily respect the block boundaries. This also implies that if \( \frac{N_X}{B} \) and \( \frac{N_Y}{B} \) are used to express the cost in terms of I/Os, they must actually always be rounded up to \( \lceil \frac{N_X}{B} \rceil + 1 \) and \( \lceil \frac{N_Y}{B} \rceil + 1 \). However, for the sake of readability, we do not write the ceilings in the asymptotic analyses since \( \Theta(\lceil \frac{N_X}{B} \rceil + 1) = \Theta(\frac{N_X}{B}) \) and \( \Theta(\lceil \frac{N_Y}{B} \rceil + 1) = \Theta(\frac{N_Y}{B}) \). Furthermore, we find that for the asymptotic analyses

\[
\Theta(M') = \Theta(M)
\]

(2.2)

because \( M \geq 7B \) and \( M' = M - 5B \), hence \( M' \geq 2B \), from which follows that \( \frac{M}{M'} = \frac{M}{M - 5B} \leq \frac{7}{2} \).

2.3.5 General Lower Bound

Before we present algorithms to score spectra against peptides, we present a general lower bound for scoring two datasets \( X \) and \( Y \) against each other.

We assume that exactly one score is calculated per pair \( x_i \in X \) versus \( y_j \in Y \). The total number of scores calculated is \( k \). It turns out that \( \Omega(\frac{1}{MD}) \)
I/Os per score are a lower bound on the I/O performance.

**Theorem 2.3.1.** Assume an algorithm computes $k$ scores of elements $x_i \in X$ and $y_j \in Y$ in $l$ I/Os. Then, $l \geq k \cdot \frac{BM}{M}$. 

**Proof:** Let us assume that we are in the process of scoring elements in $X$ against elements in $Y$ and have the memory full with $M - B$ elements. Let us further assume that scores are not stored and that scores between elements are calculated immediately after a block has been loaded and the elements are simultaneously present in the memory. If one additional block $B$ is loaded, this block can be scored against all the objects already present in the memory. For one I/O, at most $BM$ scores are thus calculated or $\Omega(\frac{1}{BM})$ I/Os per score. Multiplication by $k$ scores gives the overall lower bound of $\Omega(\frac{k}{BM})$. 

Several of our algorithms match this lower bound in many situations, namely the stripwise approaches which are cache-aware and the recursive approaches which are cache-oblivious.

### 2.4 Algorithms without Parent Mass Constraint

In the following, we analyze algorithms that score all peptides against all spectra, without the constraint that the parent mass of experimental spectra and theoretical peptides should be approximately equal. The experimental spectra are represented by the dataset $X$ and the theoretical peptides by the dataset $Y$.

#### 2.4.1 Naive Row-Wise Sweep

The most naive algorithm to score all elements of $X$ against all elements of $Y$ would be to use two nested loops. We call this algorithm ROWWISE-RECTANGLE (Figure 2.3 and Algorithm 1). Throughout this thesis, we assume that arrays are indexed starting from 1.

There are two main scenarios under which the efficiency should be analyzed. Either the dataset $X$ fits into the available memory, then the approach is fairly efficient. If not, the approach is fairly inefficient. We first analyze the case where it fits into the available memory.

**Theorem 2.4.1.** If $N_X \leq M'$ (the dataset of the inner loop fits into the available memory), the algorithm ROWWISE-RECTANGLE computes all scores in $\Theta(\frac{N_X + N_Y}{B})$ I/Os.
2.4 Algorithms without Parent Mass Constraint

Algorithm 1: Row-wise sweep of the rectangle by the algorithm RowWiseRectangle.

\begin{verbatim}
for i = 1 to NY do
  for j = 1 to NX do
    score(x[j], y[i])
  end
end
\end{verbatim}

**Proof:** If the dataset \( X \) fits into the available memory \( M' \), we initially pay \( \lceil \frac{NX}{M'} \rceil \) I/Os for loading the dataset \( X \). It remains in the memory until the algorithm terminates. No reloadings are necessary. The dataset \( Y \) is loaded gradually at a cost of \( \lceil \frac{NY}{M'} \rceil \). Hence, the total cost of the algorithm is \( \Theta \left( \frac{NX}{M'} + \frac{NY}{M} \right) \), the cost of loading both datasets once.

The inner and outer loop of RowWiseRectangle are interchangeable as far as the asymptotic performance is concerned, provided that the dataset of the inner loop fits into the available memory.

If the dataset \( X \) exceeds the memory, the efficiency of the algorithm is worse because reloading operations are inevitable. The upper bound is consequently higher, while the lower bound previously shown is still valid, though it is not tight anymore. For \( NX \geq 2M \), we provide tight bounds, whereas in the remaining region of \( M' < NX < 2M \), the upper and lower bound are not tight. However, since we will present algorithms below that are optimal for all \( NX \), we do not further elaborate on this remaining region and proceed directly to the case where \( NX \geq 2M \):

**Theorem 2.4.2.** If \( NX \geq 2M \), the algorithm RowWiseRectangle computes all scores in \( \Theta \left( \frac{NX NY}{M} \right) \) I/Os.

**Proof:** The memory can hold at most \( M \) elements of \( X \) while the remaining \( NX - M \) elements exceed the memory size limit. To execute the inner loop once, at least \( NX - M \geq M \) elements need to be reloaded because there could not have remained more than \( M \) elements in the memory from the previous inner loop execution. The inner loop is executed \( NY \) times. Loading the exceeding elements once costs \( \frac{NX - M}{M} \). Since \( NX \geq 2M \) such that \( M \geq \frac{1}{2} NX \), the cost of the exceeding elements alone is already \( \Omega \left( NY \frac{NX - M}{M} \right) = \Omega \left( \frac{NX NY}{M} \right) \). The upper bound can be derived from the scanning bound, calculating \( NX NY \) scores consumes \( O \left( \frac{NX NY}{M} \right) \) I/Os, i.e. \( O \left( \frac{1}{M} \right) \) per score. \( \square \)
2.4 Algorithms without Parent Mass Constraint

Figure 2.3: Row-wise sweep of the rectangle using two “for-loops”. The rectangle width is $N_X$ and the rectangle height is $N_Y$. Each dot represents one score calculation between a theoretical peptide and an experimental spectrum. The algorithm has been interrupted in the middle of a row, less than halfway through the entire calculation.

2.4.2 Stripwise Full Rectangle

We have seen that if $X$ fits into the available memory, both datasets are loaded one single time and no reloading operations are necessary. We can exploit this observation to improve the order of calculations if $X$ does not fit into the available memory. We can prevent many reloading operations by restricting the number of elements which the innermost loop is allowed to load. We restrict it to $M'$ elements, the size of the available memory. To make sure all calculations are done nevertheless, we add an additional outer loop to repeat the process until the full dataset $X$ is done. Graphically, this strategy can be viewed as sweeping the rectangle $X \times Y$ in a stripwise fashion (Figure 2.4 and Algorithm 4). We call this the STRIPWISERECTANGLE algorithm. It costs $O(\frac{N_X N_Y}{M'})$ I/O operations, a factor of $M'$ less than ROWWISERECTANGLE for $N_X \geq 2M$, as shown in the following.

Theorem 2.4.3. If $N_X \geq B$ and $N_Y \geq B$, the STRIPWISE_RECTANGLE algorithm computes all scores in $O(\frac{N_X + N_Y}{B} + \frac{N_X N_Y}{MB'})$ I/Os.
2.4 Algorithms without Parent Mass Constraint

![Figure 2.4: Stripwise sweep of the rectangle by the STRIPWISERECTANGLE algorithm. The strip width is $M'$, the rectangle height is $N_Y$ and the rectangle width $N_X$. In this strategy, the elements of $X$ are loaded only once. The elements of $Y$ are loaded as many times as there are strips, $\lceil N_X/M' \rceil$ times.](image)

Proof: Each value of $h$ corresponds to one strip. Within each strip, the corresponding $M'$ elements out of the dataset $X$ are used throughout the entire strip. They are loaded once at a cost of $\lceil M'/B \rceil$ when $i = 1$ and then reused until $i = N_Y$. Every element in the dataset $X$ is thus loaded exactly once altogether. On the other hand, all elements in $Y$ are reloaded for each strip. The cost per strip with respect to $N_Y$ is thus $\lceil N_Y/B \rceil$. Since there are $\lceil N_X/M' \rceil$ strips, the total asymptotic cost for loading data out of $Y$ is $O(\lceil N_X/M' \rceil \lceil N_Y/B \rceil) = O(\frac{N_X N_Y}{M'B})$. Asymptotically, this term dominates the cost $\lceil N_X/B \rceil$ caused by loading the entire dataset $X$ once, except if $N_X \leq M'$: In that case, the number of strips is $\lceil N_X/M' \rceil = 1$ and the total cost reduces to $N_X + N_Y$.

2.4.3 Recursive Full Rectangle

The stripwise approach requires prior knowledge about the size of $M'$. Cache-oblivious algorithms do not use prior knowledge about $M'$ and thus work under a wide variety of parameters [Frigo et al., 1999, Vitter, 2001].

To solve the problem at hand, we halve the entire rectangle recursively into ever smaller units down to singletons. We call this cache-oblivious algorithm RECURSIVERECTANGLE (Algorithm 3, Figure 2.5). Always the
Algorithm 2: Stripwise sweep of the rectangle (\textsc{StripWise-Rectangle}).

\begin{verbatim}
// number of strips
for h = 1 to $\lceil \frac{N_X}{M'} \rceil$ do
  // height of strip
  for i = 1 to $N_Y$ do
    // width of strip
    for j = 1 to $M'$ do
      score(x[(h-1)*M' + j], y[i])
    end
  end
end
\end{verbatim}

longer side of a rectangle is halved, which eventually results in alternation between the two sides once the two sides are roughly equal.

\textbf{Theorem 2.4.4.} If $N_X \geq B$ and $N_Y \geq B$, the \textsc{RecursiveRectangle} algorithm computes all scores in $O\left(\frac{N_X + N_Y}{M'} + \frac{N_X N_Y}{M'N} \right)$ I/Os.

\textbf{Proof:}

To analyze the cost of the recursive approach, we focus on the biggest sub-rectangle $R_f$ that still fits into the available memory. In such a rectangle, the area corresponds to the number of scores that we obtain in return. We know that each $R_f$ is fully computed before other rectangles at the same or at a higher level are computed. This is due to the hierarchical nature of the recursion. To calculate $R_f$, it is sufficient to load the corresponding data one single time at most (some may still be present from previous scorings). Based on the ideal-cache model (optimal cache replacement policy, Section 2.3.4), the order in which the computation takes place within $R_f$ does not matter.

The exact memory requirement of $R_f$ may be a bit below $M'$ since by halving side-lengths, we may never step exactly on the memory size. We can estimate this effect by looking at two rectangles in neighboring recursion levels, $R_f$ and $R_n$. At the upper level, the rectangle $R_n$ does not fit into the available memory while one level lower, the rectangle $R_f$ does fit.

To be more precise, $R_n$ has side lengths $a$ and $b$ while $R_f$ has side lengths $a/2$ and $b$ where

$$\frac{a}{2} < b \leq a \quad (2.3)$$

The area of $R_f$ is $A = \frac{a}{2}b$ and corresponds to the number of scores that
2.4 Algorithms without Parent Mass Constraint

Algorithm 3: The recursive algorithm RECURRECTANGLE or recrec\((x_1, x_2, y_1, y_2)\) halves rectangles down to singletons.

\[
\begin{align*}
\text{xlen} &= x_2 - x_1 + 1 \\
\text{ylen} &= y_2 - y_1 + 1 \\
\text{if} \ xlen = 1 \text{AND} \ ylen = 1 \text{ then} \\
& \hspace{1em} \text{// terminate recursion and calculate score} \\
& \hspace{2em} \text{score}(x_1, y_1) \\
\text{else if} \ xlen > ylen \text{ then} \\
& \hspace{1em} \text{// halve xlen} \\
& \hspace{2em} \text{xpivot} = x_1 + \lfloor \text{xlen}/2 \rfloor \\
& \hspace{2em} \text{recrec}(x_1, \text{xpivot}, y_1, y_2) \\
& \hspace{2em} \text{recrec}(\text{xpivot} + 1, x_2, y_1, y_2) \\
\text{else} \\
& \hspace{1em} \text{// halve ylen} \\
& \hspace{2em} \text{ypivot} = y_1 + \lfloor \text{ylen}/2 \rfloor \\
& \hspace{2em} \text{recrec}(x_1, x_2, y_1, \text{ypivot}) \\
& \hspace{2em} \text{recrec}(x_1, x_2, \text{ypivot} + 1, y_2)
\end{align*}
\]

the rectangle yields. We want to know how many scores we obtain for how many I/O operations. Geometrically speaking, the minimum area of \(R_f\) allowed by the previous restrictions represents the minimum number of scores while its side-length represents the memory requirement. To make it comparable to the results of the previous algorithms, we want to express the number of scores in terms of \(M'\). To do this, we introduce an additional constant \(c\) that we determine later to obtain a relative lower bound on the number of scores compared to the cache-aware approach.

\[
A = \frac{a}{2} b \geq \frac{M'}{c} \quad (2.4)
\]

We know that \(R_f\) fits into the memory and \(R_n\) does not. This can be expressed as

\[
\frac{a}{2} + b \leq M' < a + b \quad (2.5)
\]

To obtain expressions in which \(a\) and \(b\) are not mixed, we use the equations 2.3 and 2.5. We estimate \(a + b\) by \(a + a\) and by \(2b + b\) and insert these in equation 2.5. For the variable \(a\), we find that

\[
M' < a + b \leq a + a \quad (2.6)
\]
2.5 Algorithms with Parent Mass Constraint

The algorithms above are based on the assumption that all elements in $X$ need to be scored against all elements in $Y$. In the realm of tandem mass spectra however, scoring only makes sense if the parent masses of...
the theoretical and the measured spectrum are identical, except for some
error tolerance. Otherwise the theoretical and the experimental spectrum
cannot stem from the same peptide. The tolerance depends on the mass
accuracy and resolution of the mass spectrometer used. For the data used
in this thesis, we use a tolerance interval of -1 to +5 Dalton around the
monoisotopic theoretical parent mass. Since our mass spectra have a par-
ten mass in the order of 1000-3000 Da, this means that much fewer scores
need to be calculated than for the “full rectangle", approximately by a fac-
tor of \( \frac{3000 - 1000}{5 - (-1)} = \frac{2000}{6} \approx 333 \) for the measurements used in this thesis. For
a detailed discussion of experimentally observed mass deviations, please
refer to section 2.8.2.

2.5.1 Tolerance Band

In the algorithms that we present below, a peptide (theoretical spectrum)
and an experimental spectrum are scored against each other if their parent
mass is equal up to a mass tolerance. We call this the parent mass con-
straint. Let \( I = [\delta_-, \delta_+] \) be a parent mass tolerance interval. The parent
mass constraint then requires that

\[
\delta_- < pm_{exp} - pm_{th} < \delta_+ \tag{2.8}
\]

where the theoretical parent mass is the monoisotopic parent mass as
calculated from a peptide sequence\(^4\). The number of scorings needed is
mainly determined by:

1. The number \( N_X \) of experimental spectra and the number \( N_Y \) of the-
toretical spectra
2. The parent mass tolerance allowed between an experimental and a
theoretical spectrum
3. The distribution of experimental and theoretical spectrum parent masses

The general goal is to carry out all the necessary scorings between the
experimental and the theoretical spectra under the parent mass restriction.
To do this, we sort both datasets \( X \) and \( Y \) by the parent mass before the

\(^4\)The theoretical parent masses are based on the scientific literature, the measured parent
masses on instrument output. Theoretically speaking, the real masses of the molecules might
be better represented by real numbers, but in practice, real numbers can be approximated
by rational numbers at arbitrary precision. In the PepSplice implementation, theoretical and
measured parent masses are both assumed to be rational numbers and are represented by
floating point numbers.
scoring process begins. After the sorting, the scoring pattern has the shape of a tolerance band as depicted in Figure 2.6.

We define a local tolerance band width $W_y$ at a given $y$ for fixed $X$, $Y$, and $I$ as the spectra that fall into the tolerance interval $I$:

$$W_y = \{ x \in X \mid pm(x) - pm(y) \in I \}$$

The total number of scorings that are contained in the tolerance band is denoted by $k$ which is defined as follows:

$$k = \sum_{y \in Y} |W_y|$$

This is a useful measure to compare the running time between algorithms that do the scorings in the band.

### 2.5.2 Assumption of Sortedness

In the case of the tolerance band, only a subset of scores needs to be calculated, in contrast to scoring all against all elements. For the design and analysis of the following algorithms, we assume that the peptides and spectra are sorted already. This makes it easier to load peptides and spectra in ways that minimize the number of I/Os.
Sorting and scoring are fairly independent steps, and many different sorting algorithms exist. Therefore, we consider this separation adequate. However, it should be kept in mind when reading the analyses below that especially for very small $N_X$ (small spectrum datasets), the sorting cost for $Y$ (peptides in a database) may exceed the benefits that it ultimately provides. In the worst case, there would be no spectrum at all to score and the sorting would thus be utterly pointless. However, our focus is more on algorithms for relatively large experimental spectrum datasets (large $N_X$). Spectrum-centered algorithms which do a search for one spectrum already exist, but in reality, dataset sizes are increasing rapidly.

The sorting step is also justified by a theoretical argument. Consider the special case where $N_X = N_Y$ and for each peptide, there is exactly one spectrum and vice versa. One can show a lower bound for scoring unsorted datasets that is asymptotically equal to the lower bound of sorting/permuting in the I/O model [Jacob, 2006]. Therefore, if an algorithm for sorted input is asymptotically optimal, the combined cost of such an algorithm plus optimal sorting is also asymptotically optimal.

### 2.5.3 Sorting in External Memory

We illustrate the running time of sorting in external memory by using the example of merge sort. Merge sort is an algorithm that is well suited for external memory sorting. For a $k$-way variant of merge sort, [Aggarwal and Vitter, 1988] find a running time of $O\left(\frac{N}{B} \log_{M/B} \frac{N}{M}\right)$ in external memory for $k = \frac{M}{B} - 1$. (Please note that the classical running time of merge sort in internal memory is $O(N \log N)$.) We briefly summarize the $k$-way variant of merge sort in the following paragraph. For a more thorough description of merge sort, please refer to [Cormen, 2001] or equivalent references.

The input is a set of $N$ items with a key each. The output is the same set of items, but sorted by their keys. In the case of spectra and peptides, this key is the parent mass. $k$-way merge sort works in two phases: the “run formation phase” and the “merge phase”. In the “run formation phase”, the unsorted $N$ items are loaded from the disk into the memory in chunks of size $M$. Each chunk is sorted within the memory, yielding a so-called “run” of the $M$ elements in sorted order. The run is then written to the disk, and the content of the memory is replaced by $M$ more unsorted items (or less than $M$ if there are not enough left). These are sorted again and the runs written to the disk, until at the end of the “run formation phase”, the entire set is organized in $\left\lceil \frac{N}{M} \right\rceil$ sorted runs. In the “merge phase”, up to $k$ runs are merged at a time, in a similar fashion as cars would merge from several lanes onto one lane. The difference is that the items are admitted to
the one “lane” in sorted order only. Each such merge step yields one sorted run whose length is the combined length of the incoming runs. [Aggarwal and Vitter, 1988] set \( k = \frac{M}{B} - 1 \), reserving one block in the memory for calculations and for the outgoing “lane” while assigning each of the \( k \) other blocks to one incoming “lane”. The runs are processed recursively until all of them have been merged into a single sorted run of size \( N \). At this point, the algorithm terminates.

In the following, we assess the cost of the algorithm. Loading and writing \( N \) items in the run formation phase costs \( O(\frac{N}{B}) \). In the merge phase, the entire \( N \) are processed \( O \left( \log_{M/B} \frac{N}{M/B} \right) \) times at a cost of \( O(\frac{N}{B}) \). Since the initial number of \( \lceil \frac{N}{M/B} \rceil \) runs is reduced by a factor of approximately \( M/B \) at each merge step, the number of runs decreases exponentially, which results in the logarithmic expression. We see that the larger \( M \), the fewer runs there are at the end of the “run formation phase”, and the fewer I/Os are needed to arrive at the end result. \( M/B \) indicates how many runs can be merged simultaneously: the larger \( M \), the more.

2.5.4 Naive Algorithm for the Tolerance Band

We assume that the width of the tolerance band (a subset of \( X \)) at any \( y \) fits into the available memory while the entire dataset \( N_X \) does not (\( |W_y| \leq M' < N_X \) for any \( y \)). Two nested loops are then basically sufficient to obtain optimal performance, in analogy to the full rectangle with \( N_X \leq M' \). We call this naive strategy the ROWWISEBAND algorithm.

```
Algorithm 4: Row-wise tolerance band (ROWWISEBAND).

// number of rows
for i = 1 to N_Y do
  // length of row
  for j = 1 to N_X do
    if withintolerance(x[j], pm, y[i], pm) == true then
      // calculate score
      score(x[j], y[i])
  end
end
```

A crucial detail for the ROWWISEBAND algorithm is whether we can test the ‘within condition’ without paying an I/O. This could be done for example by keeping an array of the parent masses in the memory. If that is possible, this algorithm has running-time \( \Theta(\frac{N_X + N_Y}{B}) \). If not, we pay
as much for the naive testing alone as for the full naive rectangle (Section 2.4.1).

**Theorem 2.5.1.** If testing the ‘within condition’ does not cost any I/Os and the band width always fits into the available memory, the ROWWISEBAND algorithm costs $O(N_X + N_Y)$, the cost of loading the two datasets once.

**Proof:**

The elements in $Y$ are sorted by parent mass and gradually loaded by the “for $i$” loop. As the elements of $Y$ are loaded, the tolerance interval of $X$ gradually shifts with the ascending parent masses in $Y$. Throughout this shifting process, elements in $X$ enter the tolerance interval once and leave it once. While they fall into the tolerance interval, they are never evicted because the interval width always fits into the memory. Each element in $X$ is thus loaded exactly once.

If testing the ‘within condition’ is not possible without retrieving the spectrum, we can still exploit the monotonicity of the tolerance band border which is guaranteed by the double sorting. We call this the ROWWISE-BORDERBAND algorithm (Algorithm 5).

**Algorithm 5:** Border-bound row-wise tolerance band (ROWWISE-BORDERBAND).

```
// number of rows
for $i = 1$ to $N_Y$ do
    // go to start of interval
    while withintolerance(x[j].pm, y[i].pm)==false do
        intervalstart = intervalstart + 1
    end
    $j = intervalstart$
    // score until end of interval
    while withintolerance(x[j].pm, y[i].pm)==true do
        score(x[j], y[i])
        $j = j + 1$
    end
end
```

There are two inner loops on the same level. The first finds the interval start based on the previous start (which is usually fairly close due to the

---

5Assuming a least recently used (LRU) cache policy.
monotonicity of the border). The second inner loop walks through the interval and stops once the interval end is reached.

**Theorem 2.5.2.** If the tolerance band width fits into the memory, the ROWWISE-BORDERBAND algorithm costs \( \Theta\left( \frac{N_x + N_y}{B} \right) \), the cost of loading the two datasets once.

**Proof:** In analogy to the ROWWISERECTANGLE algorithm, elements never need to be reloaded here. Since the band borders are monotonous, the dataset \( X \) and the dataset \( Y \) are loaded one single time. The tolerance interval for the elements in \( X \) gradually shifts as the sorted elements in \( Y \) are loaded in ascending order. \( \square \)

### 2.5.5 Stripwise Tolerance Band

If the width of the tolerance band exceeds the memory, the memory cannot hold all elements of a row. Elements in \( X \) must therefore be evicted before all calculations for them are finished. To prevent this, we can again organize the computation in a stripwise fashion that we call STRIPWISE-BORDERBAND algorithm (Algorithm 6).

The cost is more complicated to estimate than for the full rectangle, as illustrated in Figure 2.7. Unlike for the full rectangle, we only need to score a fraction of each strip thanks to the parent mass restriction. This fraction is highly dependent on the width of the tolerance band.

**Theorem 2.5.3.** On a problem instance with size parameters \( N_X \) and \( N_Y \), the algorithm STRIPWISEBORDERBAND computes all \( k \) scores within the tolerance band in \( \Theta\left( \frac{N_x + N_y}{B} + \frac{k}{M'} \right) \) I/Os.

**Proof:** The dataset \( X \) is loaded exactly once. The dataset \( Y \) is loaded once at least, depending on the band width. Together, this gives a cost of at least \( \frac{N_x + N_y}{B} \), corresponding to the scanning bound for both datasets. In the stripwise computation, a subset of the dataset \( Y \) is loaded for each strip. The subset consists of the following components: First, there are two border areas with semi-full rectangles. These never overlap, so each border area has a cumulated height of at most \( N_Y \). Together, they generate thus a cost of at most \( 2 \frac{N_y}{B} \). Second, there are full rectangles (sandwiched between the semi-full rectangles) that have a cumulated height of at most \( \frac{k}{M} \). The cost to load the corresponding elements of \( X \) and \( Y \) is at most \( \frac{k}{M} \). Unlike the semi-full rectangles, adjacent full rectangles may overlap with respect to \( Y \). Third, the spectra may not respect the block boundaries in the border regions. There are four such border regions, two...
2.5 Algorithms with Parent Mass Constraint

Algorithm 6: Stripwise sweep of the tolerance band (STRIPWISE-BORDERBAND algorithm).

\begin{verbatim}
xstart = 1;
ystart = 1;
// enumerate strips
for h=1 to \[N_X/M'] do
  i = ystart;
  // find lowest row by checking parent masses of leftmost elements in strip
  while pm(x[(h-1) * M' + 1]) - pm(y[i]) < \delta_-
    i = i + 1
  end
  ystart = i;
  // do until upper border while checking parent masses of rightmost elements in strip
  while pm(x[h * M']) - pm(y[i]) < \delta_+
    // find row start
    j = xstart
    while pm(x[h * M' + 1]) - pm(y[i]) < \delta_-
      j = j + 1
    end
    xstart = j
    // score row
    while pm(x[h * M' + 1]) - pm(y[i]) < \delta_+
      score(x|h * M' + j, y[i]); j = j + 1;
    end
  end
  i = i + 1;
end
\end{verbatim}
2.5 Algorithms with Parent Mass Constraint

Figure 2.7: The calculation cost for the \textsc{StripWiseBorderBand} algorithm on the tolerance band consists of several components. The dataset $X$ is loaded exactly once while the dataset $Y$ may be loaded more than once. There are full areas (blue or dark grey) that are seamed by semi-full lower and upper border areas (yellow or light grey). The vertical side-length of both the lower and upper area are less or equal the height of the entire rectangle. There is no such vertical bound for the full areas, they may overlap, especially if the tolerance band is large (vertically).

per semi-full rectangle. We take them into account by adding four slim margins at a cost of 4 I/Os per strip, in total $4\lceil \frac{N_X}{M} \rceil$. Since $M' > B$, this cost is dominated by $\frac{N_X + N_Y}{B}$ in the asymptotic analysis. The total cost is $\Theta \left( \frac{N_X + N_Y}{B} + 2 \frac{N_Y}{B} + \frac{k}{M'} + 4 \lceil \frac{N_X}{M} \rceil \right)$, which yields $\Theta \left( \frac{N_X + N_Y}{B} + \frac{k}{M'} \right)$ as stated in the theorem.

2.5.6 Recursive Tolerance Band

In analogy to the recursive approach on the full rectangle, a recursive, cache-oblivious approach can also be applied to the tolerance band. We
2.5 Algorithms with Parent Mass Constraint

start with the full rectangle and halve this down to singletons. However, we require rectangles to intersect with the tolerance band. Rectangles that do not intersect are treated as dead-ends and not further pursued (Figure 2.8, Algorithm 7). We call this cache-oblivious strategy the **RECURSIVE** BAND algorithm. The following theorem shows that the algorithm **RECURSIVE** BAND has the same asymptotic running time as the **STRIPWISE** BORDER-BAND algorithm.

**Theorem 2.5.4.** On a problem instance with size parameters $N_X$ and $N_Y$, the algorithm **RECURSIVE** BAND computes all $k$ scores within the tolerance band in $\Theta\left(\frac{N_X + N_Y}{B} + \frac{k}{M/B}\right)$ I/Os.

**Proof:** The proof idea is similar to the one in the analysis of the **STRIPWISE** BORDER-BAND algorithm. There are full rectangles in the middle of the tolerance band which are tightly seamed by semi-full rectangles at the border. However, all rectangles have the same size, which is assumed to be such that their corresponding data fit into the memory, as in the proof for the **RECURSIVE** RECTANGLE algorithm (Section 2.4.3). Their side lengths

Figure 2.8: The recursive assignment of experimental spectra to peptides on the tolerance band requires rectangles to intersect with the tolerance band. Rectangles that do not intersect are treated as dead-ends and not further pursued. We call this the **RECURSIVE** BAND algorithm.
Algorithm 7: The recursive algorithm \texttt{RECURSIVEBAND} or \texttt{recpec}(x_1, x_2, y_1, y_2) halves rectangles down to singletons. Only rectangles that intersect with the tolerance band contain scores, the others are treated as dead ends and not further pursued.

\begin{verbatim}
// test intersection with tolerance band
if \( pm(x_1) - pm(y_2) < \delta_+ \) AND \( pm(x_2) - pm(y_1) > \delta_- \) then
  \( xlen = x_2 - x_1 + 1 \)
  \( ylen = y_2 - y_1 + 1 \)
  if \( xlen = 1 \) AND \( ylen = 1 \) then
    // terminate recursion and calculate score
    \( \text{score}(x_1, y_1) \)
  else if \( xlen > ylen \) then
    // halve xlen
    \( x_{\text{pivot}} = x_1 + \lfloor xlen/2 \rfloor \)
    \text{recpec}(x_1, x_{\text{pivot}}, y_1, y_2)
    \text{recpec}(x_{\text{pivot}} + 1, x_2, y_1, y_2)
  else if \( xlen \leq ylen \) then
    // halve ylen
    \( y_{\text{pivot}} = y_1 + \lfloor ylen/2 \rfloor \)
    \text{recpec}(x_1, x_2, y_1, y_{\text{pivot}})
    \text{recpec}(x_1, x_2, y_{\text{pivot}} + 1, y_2)
\end{verbatim}
are $\frac{a}{2}$ and $b$ where $\frac{a}{2} \leq b$.

The cost of the two border areas together is $\leq 2 \frac{N_x + N_y}{b} \cdot (\text{twice the scanning bound of both datasets})$ since the borders are strictly monotonous functions (from the lower left corner to the upper right corner in the Figure 2.8).

In general, the larger the band width, the more full rectangles fit within the two borderlines. The full rectangles are computed at a cost of at most $\frac{12k}{B^2}$ per score as shown in the proof for the RecursiveRectangle algorithm, which yields a cost of at most $\frac{12k}{B^2}$ for all scores in full rectangles.

A small additional cost is generated for testing whether the rectangles intersect with the tolerance band. The cost is at most 4 I/Os per rectangle, assuming that each intersection test loads two peptides and two experimental spectra per rectangle to check the parent mass constraint. The test only causes I/Os at recursion levels with rectangles that are larger than $R_f$. The number of rectangles $R_f$ along one border of the band is at most $\lceil \frac{N_x}{a/2} \rceil + \lceil \frac{N_y}{a/2} \rceil$, and the same for the second border. For the next higher level, the number of rectangles is half as high, and so on for each higher level. This is a geometric series where the total number of rectangles of size larger than $R_f$ is the same as the maximum number of rectangles $R_f$ along the border. The testing thus costs at most $4 \cdot 2 \cdot \left( \lceil \frac{N_x}{a/2} \rceil + \lceil \frac{N_y}{a/2} \rceil \right)$, $a$ is derived from $M'$ and is therefore not a constant. However, since $B \leq M'$, as defined earlier, this testing cost is dominated by $\frac{N_x + N_y}{B^2}$.

Ultimately, the running time for the full rectangles, the semi-full rectangles and the intersection testing is just $\Theta \left( \frac{N_x + N_y}{B^2} + \frac{k}{B^2} \right)$ because all constants disappear in the asymptotic notation.

## 2.6 Implementation

The above algorithms and a scoring scheme including statistical analysis (Section 2.7) were implemented in a C++ prototype which compiles and runs on Linux, Windows and Mac OS X without modifications.

### 2.6.1 “Experimental Lower Bound”

We tested the actual search speed of the implementation to obtain an estimate on the CPU time needed per score. To measure this, each score was calculated not only once, but the calculation was then repeated immediately after the first calculation. We assume that right after the score has
2.6 Implementation

Figure 2.9: The actual speed of the implementation was tested. An ‘experimental lower bound’ was measured for different dataset sizes and for different numbers of scoring repetitions. The rectangles and diamonds represent the computing times for 1, 2 or 3 identical scores while the circles indicate the averaged extra time that a score repetition costs. This extra time for a repetition is assumed to represent an ‘experimental lower bound’ since right after having calculated a score, all data should be in the L1 cache and a repetition should thus reflect the inevitable cost that is caused by the CPU and L1 cache.
been calculated, the data needed for the score still reside in the fastest possible cache. A spectrum will not fit into the CPU registers, but it will easily fit into the L1 cache of current computers. Thus, the measured CPU calculation times also include the L1 cache latency. Since even the best I/O optimized algorithms are based on using a cache or memory, we consider this combined measurement of CPU running time and L1 cache latency as the most relevant benchmark. We tested the approach on an Intel Xeon 3.0 GHz processor using different numbers of scoring repetitions and different dataset sizes. We obtained an ‘experimental lower bound’ of approximately 250 CPU cycles per score (wall clock time) and a real overall speed of approximately 450 CPU cycles per score for a dataset comprising 152838 spectra (Figure 2.9). This means that using our I/O management, we succeed in reducing memory transfer time to below CPU time. We cannot fully exclude that the compiler realizes some savings when the scoring is repeated. However, eliminating such savings would mean that the “experimental lower bound” is higher and therefore closer to the measured time. In that case, our optimization would be even more successful than we already state. The overall speed corresponds to approximately 15 CPU cycles per theoretical peak if the average peptide consists of 30 theoretical peaks. At a clock rate of 3 GHz in a single core CPU, this translates to a turnover of about 200 million theoretical peaks per second.

2.6.2 Generating Spliced Theoretical Spectra from the DNA

In the following, we describe how we generate spliced peptides in our implementation. Since we are using the peptide-centered approach, the spliced peptides are generated from the DNA sequence on the fly. Each spliced peptide consists of three components: a prefix, a gap and a suffix. These can be characterized by four values: the positions of the prefix start (PS), the prefix end (PE), the suffix start (SS) and the suffix end (SE) on the DNA (Figure 2.2). If the peptide is on the forward strand, the position numbers are in ascending order from the prefix start to the suffix end, and in descending order if the peptide is on the reverse strand. Peptides cannot contain stop codons (TAA, TAG, TGA). As soon as a stop codon is encountered in the sequence, the peptide is invalid.

We use search constraints for all four positions in order to reduce both the search time and the number of false positives. The prefix start and the suffix end are assumed to be tryptic. This means that the suffix end must be either K or R, and that the amino acid residue preceding the prefix start must be K or R as well. We use the same gap constraints as [Chen, 2001] used in their simulations: gaps must start with the dinucleotide GT and
end with the dinucleotide AG [Chen, 2001]. These constraints reduce the search space approximately by a factor of $10^2$ for the two tryptic constraints and by a factor of $4^4$ for the two dinucleotide constraints.

Our goal is the following: given a DNA string (chromosome), enumerate all spliced peptides that fulfill the above-mentioned constraints. We approach the problem as follows:

We enumerate all spliced peptides by walking over the DNA and simultaneously screening for the relevant patterns (PS, PE, SS, SE, stop codons). The screening is linear in the length of the nucleotide sequence (chromosome). The nucleotide sequence is first converted from \{A, C, G, T\} to \{0, 1, 2, 3\} by means of a conversion array. Subsequently, these nucleotide integers are converted to the amino acid sequence of the peptide to screen for tryptic prefix starts and suffix ends and to calculate peptide and fragment masses. This is also done via a conversion array. There are $4^3 = 64$ nucleotide triplets and 20 resulting amino acids (Table 2.2). We first initialize an array $x[4][4][4]$ using 64 integer nucleotide triplets and their corresponding amino acids (or stop codons '-') and 64 integer nucleotide triplets and their corresponding amino acids (or stop codons '-'). After the initialization, an integer triplet provides the three indices that are necessary to retrieve the corresponding amino acid in a single operation.

Apart from the amino acid sequence, additional information is needed to enumerate and score peptides. Among them are the reading frame of each nucleotide and also pointers to the closest patterns to assist the enumeration process. Since genomes are potentially huge and the additional information is only needed in the region where we are enumerating peptides, we make it explicit only locally, by sliding a so-called enumeration window along the DNA. We continuously enumerate all valid spliced peptides in this window. At the entrance of the window, we incrementally update the additional information needed and incrementally discard it at the exit of the window, when it is not needed anymore. The size of the window must cover the maximum length of a spliced peptide (prefix, gap, and suffix combined). Instead of continuously shifting the sequence and the additional information through the window (i.e., through arrays or arrays in the implementation) we use the modulo function to translate global genome positions to local window positions and thus smoothly roll the window over the DNA, instead of shifting it. (If the window size is 1000 and the window start is located at the position 12345666 on the genome, the equivalent position in the window is 666.) This approach uses very little RAM since the same RAM region is reused over and over. (The typical window length is $< 10000$, and the memory needed is less than 1 MB with all additional information included.) One could also allocate a full-length array for each additional type of information, with the same length as the
2.6 Implementation

Figure 2.10: For every nucleotide that enters the enumeration window, arrays containing additional information are updated incrementally. These arrays contain the original sequence information (nucleotides), but also derived information such as integer nucleotides, corresponding amino acids and reading frames (left half of figure). Furthermore, there are arrays containing relative pointers which indicate the distance to the last pattern parsed (stop codon, prefix start, prefix end, suffix start or suffix end). During the enumeration process, the pointers are used to jump from any position to the last pattern of interest. The reading frame dependent columns are highlighted in grey shades.

<table>
<thead>
<tr>
<th>Nucleotide</th>
<th>Integer Nucleotide</th>
<th>Amino Acid</th>
<th>Reading Frame</th>
<th>Since Last Stop Codon</th>
<th>Since Last Prefix Start</th>
<th>Since Last Prefix End</th>
<th>Since Last Suffix Start</th>
<th>Since Last Suffix End</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>3</td>
<td>F</td>
<td>0</td>
<td>90 36</td>
<td>31 29</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>F</td>
<td>1</td>
<td>171 30</td>
<td>32 30</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>F</td>
<td>2</td>
<td>48 9</td>
<td>33 31</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>F</td>
<td>0</td>
<td>93 39</td>
<td>34 32</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>F</td>
<td>1</td>
<td>174 33</td>
<td>35 33</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td>L</td>
<td>2</td>
<td>51 12</td>
<td>36 34</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0 42</td>
<td>37 35</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>E</td>
<td>1</td>
<td>177 36</td>
<td>38 36</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td>K</td>
<td>2</td>
<td>54 15</td>
<td>39 37</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>R</td>
<td>0</td>
<td>3 0</td>
<td>40 0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>D</td>
<td>1</td>
<td>180 0</td>
<td>41 1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>I</td>
<td>2</td>
<td>57 18</td>
<td>42 2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>S</td>
<td>0</td>
<td>6 3</td>
<td>43 3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td>P</td>
<td>1</td>
<td>183 3</td>
<td>44 4</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>R</td>
<td>2</td>
<td>60 21</td>
<td>45 5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>A</td>
<td>0</td>
<td>9 0</td>
<td>0 6</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td>L</td>
<td>1</td>
<td>186 6</td>
<td>1 7</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>C</td>
<td>2</td>
<td>63 24</td>
<td>2 8</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...
<table>
<thead>
<tr>
<th>Amino Acid</th>
<th>Codons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ala</td>
<td>GCU, GCC, GCA, GCG</td>
</tr>
<tr>
<td>Cys</td>
<td>UGU, UGC</td>
</tr>
<tr>
<td>Asp</td>
<td>GAU, GAC</td>
</tr>
<tr>
<td>Glu</td>
<td>GAA, GAG</td>
</tr>
<tr>
<td>Phe</td>
<td>UUU, UUC</td>
</tr>
<tr>
<td>Gly</td>
<td>GGU, GGC, GGA, GGG</td>
</tr>
<tr>
<td>His</td>
<td>CAU, CAC</td>
</tr>
<tr>
<td>Ile</td>
<td>AUU, AUC, AUA</td>
</tr>
<tr>
<td>Lys</td>
<td>AAA, AAG</td>
</tr>
<tr>
<td>Met</td>
<td>AUG</td>
</tr>
<tr>
<td>Asn</td>
<td>AAA, AAC</td>
</tr>
<tr>
<td>Pro</td>
<td>CCA, CCC, CCA, CGG</td>
</tr>
<tr>
<td>Gln</td>
<td>CAA, CAG</td>
</tr>
<tr>
<td>Arg</td>
<td>CGU, CGC, CGA, CGG, AGA, AGG</td>
</tr>
<tr>
<td>Ser</td>
<td>UCU, UCC, UCA, UCG, AGU, AGC</td>
</tr>
<tr>
<td>Thr</td>
<td>ACU, ACC, ACA, ACG</td>
</tr>
<tr>
<td>Val</td>
<td>GUU, GUC, GUA, GUG</td>
</tr>
<tr>
<td>Trp</td>
<td>UGG</td>
</tr>
<tr>
<td>Tyr</td>
<td>UAU, UAC</td>
</tr>
<tr>
<td>Stop</td>
<td>UAG, UGA, UAA</td>
</tr>
</tbody>
</table>

Table 2.2: 64 nucleotide triplets code for 20 amino acids and stop codons. The T in DNA is substituted by U in mRNA. Most amino acids are coded by several triplets. A nucleotide sequence therefore uniquely determines a protein sequence, but not vice versa.
2.6 Implementation

DNA itself and leave the rest to the cache management. This would probably be equally fast as the rolling window approach but it would keep huge amounts of RAM occupied with data that are obsolete.

As the window moves along the DNA, the nucleotides are screened for the dinucleotide patterns GT and AG and for stop codons, and the amino acids for tryptic patterns. During the screening, the arrays of pointers to the next pattern of each category are continuously updated. The corresponding array is set to zero if a pattern is found at the very same position, otherwise it is incremented. Numbers at any position in the array can thus be used as pointers that allow a direct jump to the previous sequence pattern (Figure 2.10, columns starting with “since last ...”). This property is exploited during the enumeration of the spliced peptides. Most pointers are incremented by 1 every nucleotide, but two of the pointer arrays are incremented by reading frame only (+3 every third nucleotide): the stop codons (TAA, TAG, TGA) and the prefix starts. The stop codons are only effective in their own reading frame while the reading frame of the prefix start is determined if the prefix end, suffix start and suffix end are given because the total length of the spliced peptide must be a multiple of 3. More detailed explanations about the window and the additional information that it contains are found in Figure 2.10.

The enumeration of the spliced peptides is done in three nested loops. When a suffix end enters the peptide enumeration window, the enumeration window is updated and the algorithm ENUMERATE_SPLICED_PEPTIDES (Algorithm 8) is triggered. It enumerates all spliced peptides that correspond to the freshly entered suffix end. The search for the patterns is done backwards from the suffix end because the previous nucleotides have been parsed into memory already and are randomly accessible in the enumeration window. Each spliced peptide must have a valid combination of suffix start, prefix end and prefix start, as specified in the algorithm.

The mass of the suffix and the prefix cannot be determined independently, since the mass is determined by amino acid masses, not by nucleotide masses, and the splice gap may start in any of 3 reading frames and produce a split triplet that is distributed between the prefix and the suffix. The entire nucleotide length of the spliced peptides is a multiple of 3, and once the suffix length is determined, the triplet split between the prefix and the suffix and their reading frames are determined. The minimum and maximum length of the prefixes, suffixes and gaps are user-defined parameters that can be chosen according to the biological context.

The presence of stop codons can be ruled out by using the array ‘since last stop codon’ which indicates how far back the last codon is located in the reading frame of the prefix or the suffix.
Algorithm 8: Given a suffix end, the algorithm `ENUMERATESPICED-PEPTIDES` enumerates all corresponding spliced peptides with their prefix start, prefix end and suffix start. Three nested loops are used that screen the whole window length $l$. They terminate whenever a condition is not met. At the start of each loop, the algorithm jumps to the next pattern by looking up its relative position in a pattern pointer array (Figure 2.11). The outer loop determines the suffix start and the suffix length of the spliced peptide, the middle loop the prefix end and the gap length, and the inner loop the prefix start and the prefix length. The maximum length of the prefix, gap and suffix are tested each time and must not be exceeded.

```plaintext
for i = l to 1 do
    look up previous suffix start in pointer array;
    if(suffixlength > suffixmax) break;
    if suffix contains stop codon, break;
    determine triplet split, this constrains prefix start;
    for j = l to 1 do
        look up previous prefix end in pointer array;
        if(gaplength > gapmax) break;
        test position and length, break if necessary;
        for k = l to 1 do
            look up previous prefix start in pointer array;
            if(prefixlength > prefixmax) break;
            if prefix contains stop codon, break;
            if split triplet is stop codon, break;
            if total peptide mass too large, break;
            if peptide mass not too small then
                forward peptide to peptide buffer;
            end
        end
    end
end
```
In summary, the enumeration window slides over the DNA and produces a steady stream of spliced peptides, which are then scored against the experimental spectra that fall within the parent mass tolerance. The major challenge in enumerating spliced peptides is to translate a plethora of biological constraints into code that produces the biologically correct output. Moreover, the strategy should be reasonably efficient with regard to the overall search time.

The cost of peptide generation is distributed over the spectra. The relative cost of peptide generation is therefore inversely correlated to the size of the spectrum dataset. A simple profiling experiment on a Linux Pentium Xeon 3.0 GHz machine showed that in a search with 23 spectra, the peptide generation turnover was about five times higher than with 25000 spectra. If we assume that with 23 spectra, the peptide generation is dominant and consumes close to 100% of the computation time, this suggests that with 25000 spectra, more than four fifth of the overall computation time is consumed for scoring and less than one fifth for the peptide generation. With 150000 spectra, this figure fell to about 3%.

2.7 Statistical Analysis

When matching a theoretical spectrum against an experimental spectrum, we need to judge if there is significant similarity between the two. A variety of scores exist to measure the similarity between two spectra: heuristic approaches such as the shared peak count (dot product), the cross-correlation [Eng et al., 1994], semi-probabilistic approaches such as the MOWSE score in MASCOT [Perkins et al., 1999], or fully probabilistic approaches such as OLAV/Phenyx [Colinge et al., 2003], the hypergeometric model proposed by [Sadygov and Yates, 2003] or NovoHMM [Fischer et al., 2005].

2.7.1 Hypergeometric Score

In our work, we use the hypergeometric score, which builds up on the shared peak count. In addition to the shared peak count, it takes also into account the length of the spectrum, the number of theoretical peaks that are to be expected and how many peaks are present in the experimental spectrum. It has first been suggested for use in spectrum scoring by [Sadygov and Yates, 2003]. The hypergeometric model is a discrete probability distribution. It describes the number of $k$ successes in a sequence of $n$ draws from a finite population of size $N$, without replacement. The population is
divided in two types of elements, $K$ of one type and $N - K$ of the other type. $k$ is the random variable, and the sum of the probabilities as $k$ runs through the range of possible values is 1. The null hypothesis is that all elements in the population have a uniform probability of being drawn. The null hypothesis is rejected if an excessive number of successes is observed.

The hypergeometric model can be represented as an urn containing $N$ balls of which $K$ are black and $N - K$ are white. In total, we draw $n$ times and get $k$ black balls and $n - k$ white balls. The question is now if we have drawn exceptionally many black balls. In the spectrum scoring context, the urn corresponds to a discretized spectrum with $N$ mass bins instead of $N$ balls. We discretize the spectra in steps of 1.0005 Daltons. This is a slightly rounded discretization constant based on two values described in the literature: 1.00048 [Perkins et al., 1999] and 1.00045 [Fischer et al., 2005]. There are $N = P/1.0005$ mass bins in a discretized experimental spectrum of parent mass $P$. Among these, there are $K$ bins containing peaks (black balls) and $N - K$ bins that are empty (white balls). The $n$ theoretical peaks determine how many times and where we draw peaks in the experimental spectrum. For a peptide of length $l$, we obtain $n = 2(l - 1)$ theoretical peaks of b- and y-ions. For each of them, we draw once at the corresponding position in the experimental spectrum. In $k$ out of $n$ draws, the theoretical peak will point to a mass bin which contains an experimental peak, so $k$ is equivalent to the shared peak count.

The hypergeometric probability density function is calculated as follows:

$$hgPDF_{N,K,n}(k) = \binom{K}{k} \binom{N-K}{n-k} \binom{N}{n}$$

This probability density function (pdf) indicates how probable it is to get a shared peak count of exactly $k$ given the variables $N$, $K$ and $n$.

Example: Given a peptide with parent mass $P = 2001$, we have $N = 2001/1.0005 = 2000$ mass bins. The peptide is assumed to be 18 amino acids long, we have thus $n = 2(18 - 1) = 34$ theoretical peaks, b- and y-ions confounded. The experimental spectrum is assumed to contain $K = 200$ peaks. If 26 of 34 theoretical peaks have a counterpart in the experimental spectrum, $k$ is 26. The probability density for this combination is $e^{-45.35} = 2.0 \cdot 10^{-20}$, which is an extremely small value and thus exception-

\footnote{This positional information is not inherent in the hypergeometric model. It slightly undermines the assumption that the bins in the spectrum have all uniform probability of being drawn, which would be the case for blindly drawing balls in an urn.}
2.7 Statistical Analysis

ally good indeed. The distribution for this example is shown in Figure 2.11.

However, we are more interested in the p-value than in the local probability at a given \( k \). The p-value is the cumulated probability for all bins from \( k \) through \( n \). In our experiments, the probability density decreases by orders of magnitude from one bin to the next and hence the differences between the \( - \log_{10}(p\text{value}) \) and the \( - \log_{10}(pdf) \) were so small (Figure 2.12) in the relevant score range that we decided to use the pdf throughout because it is computationally cheaper than the p-value. The corresponding p-value for each identification is still automatically computed and output at the end of each search. Should the deviations unexpectedly be more pronounced, this could be easily detected.

We define the score henceforth as the negative decimal logarithm of the hypergeometric probability density function:

\[
\text{score} = -\log_{10} \left(\frac{\binom{K}{k} \binom{N-K}{n-k}}{\binom{N}{n}}\right)
\] (2.10)

2.7.2 Spectrum Preprocessing

In experimental spectra from LTQ ion trap instruments, we observed that approximately one half of all bins contained peaks, most of them only noise. We obtained better identification performance when only the most intense peaks were used for scoring.

In ion trap tandem mass spectra, the peak intensity in the central region of the spectrum is a lot higher than in lower and higher region. Selecting the most intense peaks globally introduces a bias towards peaks in the central region. To prevent this, the peak intensity was normalized using an approach proposed by [Pletscher et al., 2006]. There, the peak intensity is learned over all spectra in the dataset. A histogram with 200 bins is initialized, and the peak intensity (Section 1.1.2) of each peak is added to the corresponding bin (Figure 2.13 upper plot). The bin number \( b \) is calculated based on the \( m/z \)-value \( m \) of the peak and the parent mass \( p \) of the spectrum: \( b = \left\lceil \frac{m}{p} \right\rceil \). The resulting distribution is normalized to 1 and inverted (Figure 2.13 lower plot). To avoid divisions by zero, a regularization constant of 0.0005 is added to all bins of the learned distribution. If a spectrum is multiplied with the inverse of the learned histogram, we obtain a relatively flat spectrum, i.e., the intensities are distributed much more evenly across all mass regions. Since the number of expected peaks in a spectrum is approximately proportional to the parent mass, we did
2.7 Statistical Analysis

Figure 2.11: The figure shows hypergeometric distributions for $N = 2000$, $K = 200$ and $n = 18$, $n = 26$ and $n = 34$. The probability density is on the y-axis, but transformed by the negative logarithm. The higher the curve, the lower the probability density and the more exceptional and reliable an identification. High shared peak counts mean high reliability and low probability of a random event. For very low $k$, this trend is reversed, meaning that it is also less likely to obtain an excessively small peak count, say zero when the theoretical and the experimental spectrum are both normally populated with peaks.
Figure 2.12: Correlation between p-value and probability density function for 23349 spectra in a splice site database search. The diagonal line is $x = y$, identity. The 23349 dots are all on the diagonal, barely visible, the correlation is very high. A small deviation is only visible for small values, but these are the bad scores anyway. For those, the $-\log_{10}(pdf)$ looks a little bit too good compared to the $-\log_{10}(pvalue)$. 

not select a fixed number of peaks from each spectrum. (Generally speaking, a short peptide gives a small parent mass and few signal peaks, a long peptide gives a high parent mass and many signal peaks.) We retained the highest \( x \) peaks per 100 Dalton for each spectrum, where we set \( x = 10 \) in all our searches. This means that for a spectrum with parent mass 1500, we retain 150 peaks. All spectra are binarized for the scoring, the intensity of the peaks is not used anymore after the preprocessing.

### 2.7.3 Numerical Treatment of the Hypergeometric Score

To evaluate the hypergeometric probability density function, three binomial coefficients need to be calculated, i.e., nine factorials.

\[
\text{hgPDF}_{N,K,n}(k) = \binom{K}{k} \binom{N-K}{n-k} \binom{N}{n} \tag{2.11}
\]

The highest factorial that can be represented by the standard 64-bit floating point number (double) is \( 171! = 10^{306} \). Since all factorials in the formula are either multiplied or divided by each other, it is possible to log-transform the entire formula.

\[
\log \text{hgPDF}_{N,K,n}(k) = \log \binom{K}{k} + \log \binom{N-K}{n-k} - \log \binom{N}{n} \tag{2.12}
\]

Log factorials can easily be represented by floating-point numbers, and instead of multiplications and divisions we need only additions and subtractions. (The GNU scientific library also uses log factorials to calculate the hypergeometric formula.)

\[
\log \frac{K!}{k!(K-k)!} + \log \frac{(N-K)!}{(n-k)!(N-K-(n-k))!} - \log \frac{N!}{n!(N-n)!} \tag{2.13}
\]

Since the formula is computed for each score, efficiency is crucial. In the model outlined above for discretized spectra with mass bins of approximately one Dalton, the largest factorial ever needed is given by the highest spectrum parent mass. This is well below 5000 Dalton in our datasets. It is therefore sufficient to precompute a few thousand log factorials at the beginning, which are then available for lookups. The lookup table fits easily into the L2 processor cache. The GNU scientific library offers the function `gsl_sf_lnfact()` to compute log factorials. To avoid dependency on external libraries, we used a slightly less precise but much simpler approach where
Figure 2.13: Learned peak intensity histogram and inverted histogram over 23349 ion trap tandem mass spectra. The relative mass/charge ratio is discretized into 200 bins.
we sequentially compute all log(i) and continuously sum them up, which provides log(i!). The rounding imprecision in the last digits of the floating-point number cumulates in this approach, but we tested the deviations up to log(10^{55}) using 64-bit doubles and found that the largest absolute deviation compared to the GNU gsl_sf_lngamma() function was below 10^{-8}. This precision is more than sufficient for our scoring.

**Detailed Calculation of the Above Example**

We repeat the calculation for the previously used example with N=2000, K=200, n=34 and k=26 in detail. We substitute \( \ln(x) \) by \( lfx \), whose values are stored in a one-dimensional array of precomputed floating-point numbers.

\[
\frac{\ln K!}{k!(K-k)!} + \ln \frac{(N-K)!}{(n-k)!(N-K-(n-k))!} - \ln \frac{N!}{n!(N-n)!} \tag{2.14}
\]

\[
\]

**2.7.4 Reversed Database to Evaluate Scores**

In spectrum scoring, the highest scoring match does not necessarily point to the correct sequence. In this section, we explain the reasons for this and present the reversed database approach as a way to deal with the problem. The simplest reason why the highest scoring assignment might fail
to identify the spectrum is that the peptide database might not contain the spectrum sequence at all:

1. The spectrum might stem not from a peptide but just from chemicals and solvent present in the sample.
2. The experimental spectrum might stem from contaminations.
3. The peptide might be posttranslationally modified or spliced in a way that is not specified in the search parameters.

Another reason could be that the information content of a spectrum is not sufficient to pinpoint its source with certainty even though it stems from a peptide in the sample and its sequence is present in the database. This may be caused by weak signal or by an excess of alternative explanations in the database:

1. Databases often have high degrees of homology (internal sequence similarity), which results in several competing, high scoring matches for one and the same spectrum.
2. The enzymatic digestion, the ionization and the collision induced fragmentation may not take place as expected in theory, due to a variety of chemical effects and interactions. Peaks may thus be missing, they may be shifted, may be more or less intense than expected, or additional peaks may occur where none are expected.
3. The concentration of the peptide in the sample may be too low and the signal therefore too weak.

Conflicting explanations for a spectrum are worth much less than exclusive assignments if they are to be used to confirm the presence of a particular protein or protein variant. This is even more true for the detection of splice sites where the evidence should correspond to one single position on the DNA, otherwise it is not sufficient as a basis for genome annotation. If the spectrum can be explained by an alternative database entry, its value as a splice site indicator is very limited, even if the peptide sequence is entirely correct.

The hypergeometric score measures the similarity between a theoretical and an experimental spectrum and tells us how exceptional this similarity is under the assumptions of the hypergeometric model. Some of these assumptions are also implicitly used by other scores, but they do not necessarily hold in the context of spectrum scoring. There are a number of effects that may interfere:
1. The prefix ion series and the suffix ion series are assumed to be independent of each other, although a prefix ion and its corresponding suffix ion originate in one single dissociation event.

2. Some mass bins in a spectrum are totally inaccessible since peptide fragments are sums of discrete amino acid masses. Hence the probability of a theoretical peak falling within them is zero. This is especially the case for mass bins near zero or near the parent mass. The effect of inaccessible mass bins is taken into account by Sadygov and Yates [2003] and we slightly adapt their approach (Section 2.7.5). However, the remaining bins do not have a uniform probability of containing peaks either (Figure 2.14).

3. Our LTQ spectra contain empty regions in the lowest mass region due to technical limitations of the instrument. This is a different kind of inaccessibility which is very instrument specific.

4. Scoring one spectrum against many peptide entries means multiple testing. In a tryptic search pass through the Arabidopsis protein database, allowing for methionine oxidation, the parent mass of each spectrum matched on average 25500 monoisotopic peptide parent masses within a mass window of -1 Da to +5 Da mass tolerance. This means that the hypergeometric test was carried out 25500 times and only the highest p-values were retained. If under these circumstances we find a maximum score of 4 in a spectrum, approximately corresponding to a p-value of $10^{-4}$, this is not significant. (A p-value of $10^{-5}$ or lower occurs by chance every $10^5$ times.)

To estimate the probability that the highest-scoring sequence assigned to a spectrum is indeed correct, all the above effects must be taken into account. One possible approach is to determine a background random score distribution which can be used as a reference. The best score of each spectrum is then compared to the distribution of best random scores.

Random scores can be generated actively and labeled as such, for example by using a reversed database in addition to the forward database [Kapp et al., 2005, Pletscher et al., 2006]. (This has nothing to do with the forward and the reverse reading direction on the DNA. Here, we consider reversed peptide sequences, which give rise to different fragmentation patterns than the original sequence.) Theoretically, there is a small risk that a reversed sequence is identical to a correct sequence. However, this drawback is

---

7In general, false negative identifications are less problematic than false positive identifications. The occurrence of a peptide implies the occurrence of the corresponding protein, a
more than outweighed by the fact that the reversed database as a reference cancels out the other bothersome effects. The size of the database, the distribution of the parent masses and the amino acid composition are equal in the forward and the reversed database, the internal degree of homology is the same, and so is the effect of spectrum and scoring particularities.

To accommodate peptides from heterogeneous sources, such as protein databases, DNA databases and artificially spliced DNA, we do the reversion step at the peptide level. If the peptide sequence were reverted naively, the lysine or arginine that is usually at the C-terminus of tryptically digested peptides would be at the wrong end, and the reverted peptides would be slightly less likely than the tryptic forward peptides to reach equally high random scores. The last residue is thus kept while the other positions are reverted (ACDEFGHIK gives IHGFEDCAK). After completion of the search, the forward and the reverse sequence results are assessed by a slightly adapted precision-recall analysis (Section 2.7.7).

### 2.7.5 Inaccessible Mass Bins

The assumption of the hypergeometric model that each mass bin has a uniform probability of containing a peak is a simplification. Since peptide fragment masses are sums of discrete amino acid masses, some mass bins are combinatorially inaccessible and the probability of a theoretical peak falling within them is zero. This is especially true for mass bins near zero or near the parent mass.

Sadygov proposes to determine inaccessible mass bins empirically by doing a first search pass exclusively for that purpose. We tested the empirical approach using the Arabidopsis database (Figure 2.14). The bins from 1 to 57 are all combinatorially inaccessible because glycine with a mass of 57 Dalton is the lightest amino acid. However, if there are not so many theoretical spectra in a search, even for the combinatorially accessible bins, no theoretical peak may be encountered during the search. On the Arabidopsis protein database, this was not an issue and the results were so alike for all experimental spectra that we did not learn the parameters for each spectrum empirically. Instead, we determined the number of combinatorially accessible and inaccessible bins globally at the outset.

We observed a length dependence of the random scores in the hypergeometric model. Spectra of low parent mass were assigned excessively high random scores more often than spectra of higher parent masses. The frequency of these “low parent mass - high random score” spectra was mod-

false positive assignment thus means a wrong result. In contrast, a false negative assignment is not considered proof of the absence of a protein in the sample.
Figure 2.14: For some mass bins, no theoretical peak is ever observed. We analyzed this empirically with 225 spectra in an Arabidopsis protein database search. The x-axis represents the mass bins from 0 to 350 Da. Each spectrum is represented as a horizontal intermittent line. In the upper figure, black means that at least one theoretical peak was scored against the mass bin. The lower figure gives more detailed information about the frequency of theoretical peaks, which is not uniform. We find that in all spectra, the frequencies are very similar, in spite of their different parent masses.
2.7 Statistical Analysis

est, but they contaminated the high-confidence non-random identifications at higher parent masses. We suspected that the correction for inaccessible bins was not strong enough because barely accessible bins are not taken into account in that approach. We tested how a further reduction of $N$ by a constant number of bins influences the performance. We found that after elimination of the combinatorially inaccessible bins, a further reduction by 200-300 bins gave a significantly higher performance. This might be explained both by empty border regions or by the non-uniform probabilities that are shown in Figure 2.14. In both cases, the estimation of $N$ would need to be adjusted downwards. The figure shows clearly that in the hypergeometric model presented by Sadygov, there is room for improvement with respect to the non-uniform frequency of theoretical peaks. A pragmatic but effective solution might be to adjust the parameters $N$ and $K$ in a more sophisticated way, based on learned frequencies, while still using the hypergeometric model for the score calculation itself. It would be interesting to investigate this more thoroughly, but it was beyond the scope of this work where we focus on algorithms for splice site identification.

It should also be mentioned that in spite of all the criticisms, the hypergeometric model has a high identification performance. It is a major improvement over the shared peak count approach and currently used heuristics, and this at very little computational cost.

2.7.6 Ensuring the Exclusivity of the Assignment

The best matches per spectrum are always collected and ranked according to their score. Multiple matches with equal sequences are given the same rank. As soon as there is at least one amino acid difference, the next lower rank is assigned.

A good hypergeometric score has a very low p-value and is thus especially exceptional. Due to homologies in the database, this does not necessarily mean that it is exclusive. For the same spectrum, a second match may have an equal score. To detect splice sites, only unambiguous scores can be accepted. The exclusiveness of the best score must thus be taken into account, for example by comparing it to the second-best score. Exploiting the relationship between the best and second best scores as discrimination feature was first proposed by [Eng et al., 1994] and is still used in SEQUEST and PeptideProphet [Keller et al., 2002], where it is known as dCn. We subtracted the second-best score from the best score, where the score is the negative decimal logarithm of the probability density function.
This gave the discrimination feature $d_2$.

\[ d_2 = -\log(pdf_1) - (-\log(pdf_2)) \]  \hspace{1cm} (2.15)

We observed that $d_2$ was more discriminative than the original score. Especially for stringent precision cutoffs, the number of identifications was significantly higher. We observed that the earlier-mentioned spectra with excessively high random scores and low parent masses seemed to have a less detrimental effect. If their best scores were excessively high, so were their second-best scores. Moreover, $d_2$ was much more robust towards adjustments of $N$ than the original score. This might be explained by the fact that the $d_2$ completely cancels out the three terms in the hypergeometric formula which contain only $K$ and $N$. As mentioned above, $N$ and $K$ are also the variables which probably need adjustment due to non-uniform probabilities, and using $d_2$ reduces the impact of both variables.

It might be worthwhile to take into account the distribution of the highest matches more systematically to estimate the quality of the single best match, but this was beyond the scope of this work. Using the feature $d_2$, we obtained up to 20 percent more identifications at the same precision cutoff than using the non-normalized scores, the hypergeometric pdf or the p-value.

### 2.7.7 Identification Performance Analysis

As mentioned, we not only search the regular peptide database but simultaneously also a database of the same peptides with their sequences reversed. We assess how often the scoring gets misled by the reversed sequences and observe that the forward sequences are extremely overrepresented at high scores, which is a good sign. This is visible in the modified precision-recall plot in Figure 2.15 where the curves start at 1. At low scores, the ratio between forward and reverse sequences is almost exactly 1:1 between forward and reversed sequences. Since the ratio of forward to reverse sequences in the database is also 1:1, this indicates that the reverse sequence approach works. To show this 1:1 ratio, the Figure 2.15 is not ideal because the datapoints are cumulated in descending order from the highest scores. Therefore, we also show another figure where the datapoints are cumulated in ascending order from the lowest score (Figure 2.16).

The identification performance of the hypergeometric model is very satisfactory (Figure 2.15). The plot shows the results for one large dataset of 152838 spectra. We also searched other datasets and found a compara-
Figure 2.15: Precision-recall curves of three different search results are shown: the normal protein database search has the highest performance, followed by a combined protein-chromosome database search. The pure DNA search without protein database fares less well. The precision is shown as a function of the fraction of all spectra: 152838 doubly charged spectra from the dataset ‘Osmo10-g’. The spectra are sorted by identification quality from left to right. The plot must be interpreted a bit differently than usual, as described in the text. The lower figure shows the same data in a different zoom, focusing on high precision identifications. At a precision of 0.99 (0.98), 18 percent of the spectra labeled as doubly charged were identified in the protein database search.
Figure 2.16: The figure shows to what degree the scoring is misled by reverse sequences. Two cumulated sums are shown: the upper curve counts the number of spectra where the best hits are forward sequences, and the lower curve counts the spectra where the best hit are reverse sequences. The spectra are sorted in ascending order by the score of the best hit. At low scores, the two curves grow at almost the same rate, which means that the ratio between forward and reverse sequences is almost 1:1 in this region, i.e., the identification process is not able to distinguish between forward and reverse sequences based on the signal in the spectrum. At intermediate scores, the scoring seems to be able to somewhat distinguish between forward and reverse sequences, so the curves grow apart. At very high scores, almost no additional reverse sequences are found, only forward sequences. This means that the identification process is so reliable that it is virtually never misled by the reverse sequences. We used a dataset of 46473 spectra that were searched against the TAIR Arabidopsis protein database.
ble performance, though the number of identifications strongly depends on the experimental settings, such as the density and complexity of the samples. The more material is loaded into the mass spectrometer, the more spectra of high quality usually result. The results are in good agreement with the results of SEQUEST/PeptideProphet, which is currently one of the best performing tandem mass spectrum identification approaches \cite{Kapp2005} for standard protein database searches.

The evaluation of the performance is a bit challenging because the standard precision-recall plot does not capture well some properties of the reversed database search. First, the number of all positives that constitutes the denominator of the recall \footnote{recall=TP/(TP+FN)} can only be estimated. Second, the database consists of equally many forward peptides and random peptides which means that the precision \footnote{precision=TP/(TP+FP)} must be corrected. In a combined database of equally many forward and reversed peptides, spectra can be assigned false positive identifications in the forward part too, at almost equal probability as in the reversed database. If spectra were searched against a nonsense database with forward and reversed peptides, we would expect the plot to give a horizontal line at precision=0.5, as shown in Figure 2.15. The real precision is thus twice as far away from the ‘perfect 1’ than the precision shown in the figure (a seeming precision of 0.99 corresponds to a real precision of 0.98, a seeming precision of 0.5 to 0.0).

This reversed database search approach combined with a precision-recall analysis is more powerful than other approaches. Statistical evaluation tools of database scores, such as PeptideProphet \cite{Keller2002}, make assumptions about the score distribution of database searches. In contrast, the reversed database approach in conjunction with a precision analysis does not assume any distributions whatsoever. The only parameter needed is the proportion between the number of forward and reversed peptides that constitute the database. This is a very useful property for ‘cascading’ database searches. There, standard protein and DNA database searches are followed by searches with a larger search space, allowing for post-translational modifications, semi-tryptic or non-tryptic peptides or, in our case, splice sites. Spectrum quality scoring approaches can be used to extract promising subsets from the original data. This saves search time and reduces the risk of false positive identifications, but it may affect the score distribution and thus the performance of PeptideProphet and similar tools. In contrast, the reversed database approach should be resistant against such changes.
2.8 Biological Results

2.8.1 Experimental Setup

An Arabidopsis cell culture was grown at room temperature in growth medium containing 0.3% sucrose. The cell culture consisted of 4 aliquots of 500ml, each in a 2l-Erlenmeyer flask. After an adaptation phase of 2 days in the dark, 2 aliquots were shifted to the light whereas the other 2 aliquots were left in the dark. The experiment was continued in this setting for 5 days. Then, the cells were centrifuged and frozen in liquid nitrogen. The sample was serially fractionated as described in [Kleffmann et al., 2004], yielding the following fractions:

1. soluble proteins (aqueous buffer)
2. peripheral membrane associated proteins (8 molar urea)
3. membrane proteins (detergents, CHAPS and BRIJE35)
4. remainder (SDS)

Of each fraction, 500-1000 micrograms of protein mixture were separated on a 12% 1D-SDS gel. The gel was sliced horizontally such that proteins of similar molecular weight were in the same slice. The proteins were digested overnight in-gel using trypsin, in accordance with [Shevchenko et al., 1996]. The digested protein sample was measured on a Thermo-Finnigan LTQ ion trap mass spectrometer. The sample was loaded on a reverse-phase C18 column with an inner diameter of 75 micrometer. The column was directly coupled to the mass spectrometer and the elution was carried out using a solvent gradient between an organic solvent and an aqueous solution. 3 data-dependent tandem mass spectra were generated per full scan. In total, we acquired 490 LC-MS runs which yielded 1.4 million spectra labeled as doubly charged. The data were searched using SEQUEST against the TAIR protein database (version 20040228 plus common contaminants). The search was constrained to tryptic peptides, allowing for one missed cleavage at most. Dynamic methionine oxidation (M16) and static cysteine carboxamidomethylation (C57) were used as modifications. For all PepSplice searches, we used a parent mass tolerance of ±1/±5 Dalton around the monoisotopic theoretical parent mass. We also used M16 and C57 as modifications, as in the SEQUEST search, but we did not restrict the number of missed cleavages (thereby increasing the search space, see Section 2.2.1). For extracting unidentified high quality spectra with QualScore, we chose a score cutoff of 0 (roughly 75% of the SEQUEST/PeptideProphet identifications were above this cutoff).
2.8 Biological Results

2.8.2 Parent Mass Deviation

The values measured by the tandem mass spectrometer are subject to small measurement errors. In measurements of any kind, we usually expect deviations to consist of the following two error components:

1. A random error (resolution dependent) that can often be approximated by a Gaussian distribution.

2. A systematic error (due to miscalibration) that can often be approximated by a linear shift.

Apart from the measurement error, there is another effect in mass spectrometry that can cause deviations. The chemical elements that constitute proteins consist of more than one kind of isotope per element. This effect translates into an isotope pattern, where a small cluster of peaks with a spacing of 1 Dalton is observed (or a fraction of 1 Da if the ions carry more than one charge). Peptide parent masses are sometimes indicated as the average mass value of this distribution, or as the lowest mass-value in the distribution, also called the monoisotopic peak.

Both for the measurement process and the database search afterwards, parent mass tolerances need to be chosen, either based on monoisotopic values or average values. We compared the measured parent masses once with the monoisotopic parent masses and once with the average parent masses. We found that in both cases, the distribution of the parent mass deviations of the Thermo Finnigan LTQ instrument does not look like a Gaussian whose mean may be a little bit shifted, but that it has a very clearly resolved isotope pattern instead (Figures 2.17 and 2.18).

We believe that the instrument can actually measure individual peaks at a precision that is significantly better than half a Dalton (the spacing between isotopes of doubly charged ions) but instead of detecting parent ion isotope patterns and choosing the monoisotopic peak, it indiscriminately picks the highest peak. That would mean that the largest error stems from “mispicking” isotopes and causes deviations of several Dalton in the parent mass of doubly charged ions, while the random error and the calibration error are far below one Dalton. Parent mass adjustment methods at the tandem mass spectrum level might yield a higher parent mass precision.

The difference between the monoisotopic and the average parent mass deviation histogram becomes more understandable once the parent mass dimension is also added (Figure 2.19). We find that the isotope pattern is detectable with both the monoisotopic and the average masses. However, the parent mass deviations of average parent masses depend on the parent
2.8 Biological Results

Figure 2.17: Histogram of monoisotopic parent mass deviations. It is based on 15600 spectra identified at precision above 0.995 (0.99) in a whole genome search. The deviation is calculated by subtracting the monoisotopic parent mass from the measured parent mass. Instead of a shifted Gaussian distribution, we observe a very clearly resolved isotope pattern.

Figure 2.18: Histogram of average (not monoisotopic) parent mass deviations. The resolution of the isotope pattern is clearly worse than in Figure 2.17. It seems that the measured parent masses are better reflected by monoisotopic theoretical parent masses.
mass while the deviations of the monoisotopic parent approximately are
the same for all parent masses, they show no shift.

Based on these observations, we use monoisotopic parent masses for
all our searches, not average masses. (Monoisotopic masses have the addi-
tional advantage that they are totally independent of the isotope composi-
tion of the elements. Elements such as carbon undergo radioactive decay
processes, which are also used for C14-dating in archeology.)

The search time is approximately linear in the width of the tolerance
window. A tailor-made search interval minimizes any waste of CPU time
and maximizes the number of identifications. SEQUEST for example does
not allow to determine the deviation window asymmetrically, but in our
opinion, that is a very simple and effective feature. It was thus imple-
mented in the PepSplice software, and a parent mass deviation histogram
is automatically calculated for every search.

2.8.3 Whole Genome Search

As a preliminary stage to splice site finding, we implemented a tryptic
whole genome search function and tested it on our data. We searched
1.4 million spectra using the $d_2$-score on the entire genome of Arabidop-
sis and obtained 98337 spectrum identifications above a precision cutoff
of 0.995 (0.990, error rate 1%). The results are depicted by their location
on the chromosomes in Fig 2.20. One clearly sees that the chromosomes
are densely covered by identifications, except for a few regions which are
nearly empty, the centromere regions. Of these, 736 spectra were found
exclusively in the whole genome search and not in the protein database
search. The results of a preliminary analysis by Jonas Grossmann are shown
in Table 2.3. The data correspond to the data shown in the chromosome
view in Figure 2.20, where each diamond represents one of the 736 spectra
that were exclusively found in the whole genome search.

The speed of the whole genome search was much higher than expected,
the 1.4 million spectra were searched against the Arabidopsis genome within
12 hours on a dual-core Pentium Xeon 3GHz CPU running under Linux.
This is equivalent to a turnover of approximately 30 spectra per second,
roughly ten times faster than the mass spectrometer can acquire the spec-
tra. The parent mass tolerance window was 6 Dalton, which is generous
and computationally more intensive than a smaller window.

These spectra were acquired by Jonas Grossmann. Jonas Grossmann will present more
extended biological results based on these measurements in his PhD thesis at the department
of biology at ETH Zurich.
Figure 2.19: Parent mass deviations for monoisotopic and average parent masses were compared. On the y-axis, the figure shows the same information as the histograms in Figures 2.17 and 2.18 but now the parent mass dimension is added on the x-axis (bins of 100 Da). Intense gray shades indicate high counts. One can see that the dark streaks are horizontal in the monoisotopic case and diagonal in the average case. In the monoisotopic/horizontal case, the isotope pattern (spacing 1 Dalton) is much better resolved than in the average/diagonal case. There, the isotopes are clearly resolved in the two-dimensional representation but melt together if projected onto the single dimension at the right border of the frame.
Figure 2.20: Chromosome view of 98337 spectra identified in a whole genome search on 1.4 million spectra. The precision is 0.995 (0.990, error rate 1%). Each of the five Arabidopsis chromosomes is densely covered by identifications, except for one near-empty region per chromosome, probably the location of the centromere. Forward reading frames are to the left of the chromosome, reverse reading frames to the right. The vertical position is precise, the horizontal position is slightly jittered to ensure that multiple identifications at the same site are visible. (Please refer to the PDF document for a colored, zoomable version of the figure.)
Figure 2.21: 736 spectra that were identified exclusively in the whole genome search, not in the protein database search, are added here to the Figure 2.20. The precision level is 0.995 again, and the additional spectra are depicted as diamonds.
2.8 Biological Results

Table 2.3: The table shows the results of a preliminary analysis by Jonas Grossmann of the whole genome search results. The number of unique identified peptides is 75, categorized by 6 categories indicating the reason why the spectra could not be identified in the regular protein database search. These data may give hints how genome annotation can be improved.

<table>
<thead>
<tr>
<th>Category</th>
<th>Spectra</th>
<th>Peptides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wrong stop</td>
<td>71</td>
<td>11</td>
</tr>
<tr>
<td>Absent from protein DB, but AGI number exists</td>
<td>412</td>
<td>17</td>
</tr>
<tr>
<td>Wrong start</td>
<td>115</td>
<td>18</td>
</tr>
<tr>
<td>Intergenic</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Splice site not in protein DB, found elsewhere</td>
<td>71</td>
<td>18</td>
</tr>
<tr>
<td>Gene absent from protein DB, present in EuGene</td>
<td>45</td>
<td>9</td>
</tr>
</tbody>
</table>

2.8.4 Splice Site Search

After the whole genome search, we ran a splice site search. From the 1.4 million spectra that had been searched using SEQUEST and PeptideProphet, we extracted 98437 unidentified high quality spectra using QualScore (Chapter 3). Using PepSplice, we searched these spectra against the protein database a second time, allowing for semi- and non-tryptic peptides. We then searched them against the whole genome without splicing and finally against the genome with splicing. (This was done in one and the same run. The number of equal-ranking hits for proteins, DNA and spliced DNA is shown separately in the search results). We searched for tryptic spliced peptides with a maximum gap of 3000 nucleotides and a minimum prefix and suffix length of 6 nucleotides. (The search space for these parameters is approximately 2000 times larger than for a standard Arabidopsis protein database search and 300 times larger than for a whole genome search, see Section 2.2.1).

Whenever we found a spliced peptide for which a peptide with identical score existed in the protein or the whole genome search, the identification was eliminated. Spectra pointing to more than one spliced peptide with identical scores were also eliminated. After these elimination steps, 144 spliced spectra above a precision cutoff of 0.995 (0.990) and 255 spliced spectra above a precision cutoff of 0.95 (0.90) remained. These are depicted in Figure 2.22. Most of them occur in small clusters, which is a good sign. Random false positive hits would be more evenly distributed, assuming that they would not have any positional preference. In the optimal case, splice site hits cluster together with whole genome hits. Whole genome
Figure 2.22: 144 spliced peptide spectra of high confidence (large pluses) and 255 of intermediate confidence (small pluses) are added to Figure 2.21. The precision level is 0.995 (0.990) for the 144 spectra and 0.95 (0.90) percent for the 255 spectra. Please refer to the PDF document to zoom in and explore this map in more detail.
hits in Arabidopsis can be determined with fairly high confidence and have been found to be consistent with other biological observations in the analyses so far. The number of splice sites matches is somewhat lower and they are more difficult to verify. Since the search space for splice sites is huge, there is also some danger that they match to similar but not equal peptides, i.e., if a spectrum stems from a posttranslationally modified peptide, there may be no counterpart in the database, but a spliced peptide may be similar enough to give a fairly good hit. This possibility deserves to be further investigated, e.g., by equipping PepSplice with a feature for the detection of posttranslational modifications. However, since we found a number of whole genome hits which contradict a predicted splice site, this strongly indicated that splice site prediction is not perfect yet. Thus, we expect to find a comparable number of spectra that justifiably point to sites that are not predicted in the protein database. We find several examples where newly predicted splice sites occur within the same open reading frame as whole genome hits, which indicates that the splice site prediction may be indeed erroneous locally, missing the splice site by a small margin. The most striking case of this kind is located in the open reading frame (gene) At1g64790.

2.8.5 Proof of Principle: ORF At1g64790

One cluster, located in the open reading frame (ORF) At1g64790, is especially striking: we find two exclusive whole genome hits (no hit in the protein database) and two splice site hits, supported by 17 spectra (Table 2.4). A further 115 spectra with unspliced whole genome hits (and corresponding protein database hit) fall also into the same open reading frame, at a spectrum identification precision of 0.995 (0.990). Normal protein database search hits are not included in these numbers. All spectra are located between 24069677 and 24085133 on the reverse strand of the Arabidopsis chromosome one (Figures 2.23, 2.25, 2.26). The ORF spans 16677 nucleotides and contains 58 predicted exons. Several competing gene predictions contradict each other around the position 24083000. Especially the Eugene prediction seems to be different from the other two AGI predictions (Figure 2.23), but MIPS and TIGR also diverge (Figure 2.24). TAIR (www.arabidopsis.org) describes At1g64790 as “translational activator family protein, similar to HsGCN1 (Homo sapiens)”. According to TAIR, the protein is 2441 amino acids long and has a molecular weight of 265340 Dalton. A cDNA with length 7618 exists under the accession

11 An open reading frame or ORF is any sequence of DNA or RNA that can be translated into a protein. It is usually equivalent to a gene.
number 1009023556 in TAIR (2005-11-05). The protein shares 63% identity with a rice homologue and 42% identity with a human homologue. In alignments between the Arabidopsis protein, the rice protein, mammalian proteins and several other species, it turns out that three of the four hits are located across or in close vicinity of gaps (Figures 2.29 and 2.28). The fact that several spliced peptides and exclusive whole genome hits are found in the same ORF strongly suggests that the gene structure prediction in that particular region is probably erroneous indeed. In NCBI, we found one alternative Arabidopsis protein entry, gi5042415, which is 2698 amino acids long instead of 2441. We aligned both proteins and added the four novel hits for At1g64790 to the alignment (Figure 2.27). We found that both whole genome identifications and one of the spliced peptide identifications are contained in gi5042415. This effectively confirms three of our four hits and shows that we can reliably detect spliced peptides, in spite of the huge search space. The fourth hit is also close to a gap in the alignment (Figure 2.29) but not explained by other protein entries so far. We conclude that genome annotation might well benefit from adjustment by proteomics data, and that it might be worthwhile to submit spectra more systematically to extended searches, including splice site searches.

2.9 Conclusion

We successfully identified several hundred spectra that contradict current gene predictions for Arabidopsis. We use the gene At1g64790 as a proof of principle and find that the evidence is very consistent on a number of levels: In the location of our corrections, gene models diverge and gaps occur in a multiple protein sequence alignment between a variety of species. Moreover, three of four corrections match an alternative prediction of the protein. If one considers that the search space for spliced peptides on the Arabidopsis genome is about 2000 times larger than for a standard protein database search, we find a respectable number of identifications while keeping the false positive rate and the running time under tight control.
2.9 Conclusion

Table 2.4: The two unpredicted splice sites and the two unpredicted whole genome hits in the gene At1g64790 are supported by 24 spectra in total. Whole genome hits are represented by two coordinates, spliced peptides by four. Coordinates in descending order mean that the identification is located on the reverse strand.

<table>
<thead>
<tr>
<th>PS</th>
<th>PE</th>
<th>SS</th>
<th>SE</th>
<th>spectra</th>
</tr>
</thead>
<tbody>
<tr>
<td>24080376</td>
<td>24080335</td>
<td>24080255</td>
<td>24080244</td>
<td>3</td>
</tr>
<tr>
<td>24083470</td>
<td>24083450</td>
<td>24083368</td>
<td>24083330</td>
<td>4</td>
</tr>
<tr>
<td>24083617</td>
<td>24083579</td>
<td>24083889</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>24083939</td>
<td></td>
<td></td>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>

Figure 2.23: Three competing gene predictions for the gene At1g64790 are shown in FlagDB. The upper arrows represent CDS Eugene, the middle arrows CDS AGI and the lower arrows mRNA AGI. Above the line are the gene (exon) predictions, the very short arrows on the line are the spectra that were found in the whole genome search. (Please refer to the PDF document for a colored, zoomable version of the figure.) The region around 24083000 is contradictory, especially Eugene diverges from the other two predictions.

Figure 2.24: Two competing gene predictions for the gene At1g64790 are shown here, TIGR versus MIPS. They also diverge around 24083000.
Figure 2.25: All whole genome hits of the gene At1g64790 are represented below the line. Above the line are the spectra which were only found in the whole genome search, not in the protein database search. Furthermore, there are two splice sites that were not indicated in the protein database search, supported by 3 and 4 spectra respectively. One of the 4 spectra has a precision of only 0.95 (0.9), all others have a precision of 0.995 (0.99). The fact that novel whole genome hits and novel splice sites co-occur suggests that the gene prediction for this gene may indeed be slightly erroneous and further annotation therefore needed.

Figure 2.26: This is a zoom into Figure 2.25, highlighting the novel hits that could not be detected by the protein database search. These cover the region from 24080244 to 24083939 on chromosome 1 in the reverse direction, approximately 3700 nucleotides.
2.9 Conclusion

Figure 2.27: The four hits in At1g64790 are placed in an alignment between two protein predictions. The query sequence (gi42562949) is identical to At1g64790 in the TAIR protein database - we used the TAIR protein database for the protein search prior to the splice site search. The two whole genome hits (IGMLNAVQELASAP, LSILSAVASWASR) are explained by the second sequence, gi5042415. One of the spliced hits is also explained by the second sequence (NPDTISQISDLLSPLIQLVK), but not the other splice hit (SGPLPVDTFIPFLER).
2.9 Conclusion

<table>
<thead>
<tr>
<th></th>
<th>Arabidopsis1</th>
<th>Arabidopsis2</th>
<th>Arabidopsis3</th>
<th>Rice1</th>
<th>Rice2</th>
<th>Mouse1</th>
<th>Mouse2</th>
<th>Mouse3</th>
<th>Mouse4</th>
<th>Rat1</th>
<th>Rat2</th>
<th>Human1</th>
<th>Human2</th>
<th>Dog1</th>
<th>Fly1</th>
<th>Worm1</th>
<th>Yeast1</th>
<th>Neurospora</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>2</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>3</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>4</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>5</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>6</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>7</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>8</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>9</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>10</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>11</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>12</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>13</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>14</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>15</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
<tr>
<td>16</td>
<td>KRAI----------</td>
<td>LRAI----------</td>
<td>KRAI----------</td>
<td>TGSXG------</td>
<td>TGSXG------</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG-</td>
<td>TGSXG---</td>
<td>TGSXG-</td>
</tr>
</tbody>
</table>

Figure 2.28: Both whole genome hits, IGMLNAVQELASAPEGK and LSIL-SAVASWASR, occur in the sequence Arabidopsis2 but not in Arabidopsis1 (TAIR). Where Arabidopsis1 has a gap, Arabidopsis2 has a sequence part that is fairly conserved throughout various species (shaded amino acids), including humans.
Figure 2.29: The splice site hit NPDTSQDLLSPLIQLVK is also contained in the sequence Arabidopsis2, but not the other; SGPLPVDTFTFIPILER.
2.9 Conclusion

The figure shows predicted exons and introns, where the gap dinucleotides are always gt–ag. The highlighted regions indicate the locations of splice sites.

Figure 2.30: Approximately the same region as in Figure 2.25 is shown here in text format, based on FlagDB [Samson et al., 2004]. Predicted exons are shown in uppercase, introns in lowercase. The first two bold stretches highlight the whole genome hits, the remaining four highlight the two splice sites, where the gap dinucleotides are always gt–ag.
Chapter 3

QualScore: Dynamic Spectrum Quality Assessment and Iterative Computational Analysis of Shotgun Proteomic Data

Authorship

This chapter is shared work with Jonas Grossmann, Alexey I. Nesvizhskii, Sacha Baginsky and other authors. It has been published in “Molecular and Cellular Proteomics” [Nesvizhskii et al., 2006] and is only slightly modified here to better fit into the thesis as a whole. To avoid splitting the work into incoherent fragments, the work is presented in its entirety here. Jonas Grossmann will present the same chapter in his PhD thesis in the department of biology at ETH Zurich within the next few months.

3.1 Introduction

In mass spectrometry-based proteomics, frequently hundreds of thousands of tandem mass (MS/MS) spectra are collected in a single experiment. Of these, a relatively small fraction is confidently assigned to peptide se-
3.1 Introduction

sequences, while the majority of the spectra are not further analyzed. Spectra are not assigned to peptides for diverse reasons. These include deficiencies of the scoring schemes implemented in the database search tools, sequence variations (e.g. single nucleotide polymorphisms) or omissions in the database searched, post-translational or chemical modifications of the peptide analyzed, or the observation of sequences that are not anticipated from the genomic sequence (e.g. splice forms, somatic re-arrangement and processed proteins). To increase the amount of information that can be extracted from proteomic MS/MS datasets we developed a robust method that detects high quality spectra within the fraction of spectra unassigned by conventional sequence database searching and computes a quality score for each spectrum. We also demonstrate that iterative search strategies applied to such detected unassigned high quality spectra significantly increase the number of spectra that can be assigned from datasets and that biologically interesting new insights can be gained from existing data.

Shotgun proteomics creates significant computational challenges [Nesvizhskii and Aebersold, 2004, Patterson, 2003, Johnson et al., 2005, Russell et al., 2004]. Large numbers of MS/MS spectra acquired in each experiment need to be computationally processed to identify peptides that produced them and to infer what proteins were present in the original sample. In most high throughput studies, peptide identification is performed by searching MS/MS spectra against protein sequence databases. A number of automated database search tools have been developed for that purpose, including commercial and open source programs [Eng et al., 1994, Mann and Wilm, 1994, Perkins et al., 1999, Clauser et al., 1999, Field et al., 2002, Zhang et al., 2002, Craig and Beavis, 2004, Geer et al., 2004, Colinge et al., 2003]. These programs correlate the experimental MS/MS spectra with theoretical fragmentation patterns of peptides obtained from a sequence database and use various scoring schemes to find the best matching peptide sequence. This high throughput protein identification process, however, is prone to false positives resulting from incorrect peptide assignments to MS/MS spectra by the database search tools [Nesvizhskii and Aebersold, 2004, Keller et al., 2002, Nesvizhskii et al., 2003, Baldwin, 2004, Carr et al., 2004]. The problem of false positives has received significant attention in recent years. As a result, statistical approaches and computational tools were developed for assigning confidence measures to peptide and protein identifications and for estimating the false identification rates. These tools reduce the need for time consuming manual verification of peptide assignments and allow faster and more consistent analysis of large-scale datasets [Nesvizhskii and Aebersold, 2004].
Despite the progress in developing new database search tools and methods for statistical validation of peptide assignments to MS/MS spectra, the number of spectra that remain ‘unassigned’ (i.e., the sequence of the peptide that produced the spectrum is not known) in any experiment is significant. In fact, of all MS/MS spectra acquired in a typical shotgun proteomics experiment only a relatively small fraction (e.g., typically less than one half in the case of the ion trap instruments) is assigned a peptide with high confidence. The fraction of assigned spectra is usually even far less than one half. Low quality spectra are often generated when instruments keep measuring even though (too) little sample is being injected, but this heavily depends on the complexity, concentration and pre-fractionation of the sample, on the measurement parameters, on the speed and the detection threshold of the mass spectrometer and other factors. In the whole genome search described in Section 2.8.3, the fraction of assigned spectra was about 7% at an error rate of 1%, so there remain plenty of unassigned spectra to be investigated.

The use of higher quality instruments having better mass accuracy and resolution alleviates the problem, but does not eliminate it. Reasons for such a high failure rate include the deficiencies of the scoring schemes used to quantify the degree of similarity between the experimental spectrum and those predicted for database peptides, ambiguities in the determination of the charge state of the peptide ions selected for fragmentation (in low mass accuracy instruments), the presence of spectra arising from non peptidic contaminants, concurrent fragmentation of multiple different precursor ions, and the low quality of many spectra due to excessive noise or incomplete peptide fragmentation [Nesvizhskii and Aebersold, 2004; Johnson et al., 2005; Resing et al., 2004; Chalkley et al., 2005]. However, a significant fraction of high quality, peptide derived spectra also remain unassigned, and the failure of the database search tools to interpret them correctly cannot be explained by the reasons mentioned above.

One source of unassigned high quality spectra are peptides containing a post-translational or chemical modification. To speed up the analysis, MS/MS database searching is often performed in a way that does not anticipate the presence of modifications in the peptides analyzed and the modified peptides are therefore missed. Another source for such spectra are peptides whose sequence is not present in the searched protein sequence database, e.g., peptides corresponding to unanticipated alternative splice forms or sequence variants of known proteins (polymorphisms). Although identification of modified, mutated, or novel1 peptides can be of

1The term ‘novel’ peptide here refers to a peptide identified by searching an unannotated genomic database and whose sequence is not present in any major protein sequence database
3.1 Introduction

high significance, the high quality spectra that remain unassigned after the initial database search pass through the data in most high throughput experiments are not further analyzed. Finding such spectra would normally entail manually sifting through large amounts of low quality data, which is rarely practiced due to the ever increasing number of newly acquired datasets waiting to be processed and interpreted. Thus, there is a clear need to develop computational tools for automated spectrum quality assessment that could be used to detect unassigned high quality spectra and mark them for subsequent re-analysis. Such quality assessment tools can also be used for a different task, e.g., to filter out low quality spectra prior to database searching in order to reduce the computational time, or to assist in the process of discriminating between correct and incorrect peptide assignments.

In this work, we present a dynamic quality scoring approach for finding high quality unassigned spectra in large shotgun proteomics datasets. The basic idea behind the method is that in the first database search pass through the data, high confidence peptide assignments are generally based on high quality MS/MS spectra. Therefore, the notion of what constitutes a high quality spectrum can be learned from the analyzed data itself, i.e., without relying on a training dataset created using spectra from a different experiment [Nesvizhskii et al., 2004]. This way the statistical classifier is automatically developed for each dataset anew, ensuring the robustness of the method toward variations in the MS/MS spectrum properties caused by differences in acquisition methods or instrument to instrument variability. This distinguishes this method from other recently described approaches [Moore et al., 2000; Nesvizhskii, 2001; Bern et al., 2004; Purvine et al., 2004; Xu et al., 2005] that can produce inaccurate results when applied to spectra that differ significantly from those used for training. The statistical spectrum quality classifier is computed using an extended set of spectrum features, including those designed to take into account the knowledge of the peptide fragmentation process.

The accuracy of the method is evaluated using a dataset of MS/MS spectra from a recent experiment on human lipid rafts [Von Haller et al., 2003]. We find that the spectrum quality classifier quickly and automatically detects high quality spectra left unassigned during the initial computational pass through the data. We also demonstrate that by investigating those unassigned high quality spectra more comprehensively using existing protein sequence databases, and by searching against genomic databases, one can significantly increase the number of identified peptides, including peptides containing modifications and sequence polymor-
3.2 Experimental Procedures and Datasets

3.2.1 Experimental Datasets

Three experimental datasets of MS/MS spectra were used to evaluate and optimize the quality scoring method and to investigate the sources of peptides that produced those spectra. All spectra were acquired using electrospray ionization (ESI) ion trap tandem mass spectrometers.

1. The *H. influenzae* dataset consisted of 15 LC-MS/MS runs and was previously used as a test dataset for the development of a statistical model for validation of peptide assignments to MS/MS spectra [Nesvizhskii et al., 2003]. Sample proteins were digested using trypsin and the resulting peptide mixtures were separated using reverse phase and strong cation exchange (SCX) chromatography prior to MS/MS sequencing. The dataset contained more than 16,000 multiply charged MS/MS spectra generated from the membrane fraction of the sample. The spectra were searched using SEQUEST as described previously [Nesvizhskii et al., 2003], and peptide assignments to the spectra were processed using PeptideProphet. Of the 31728 search results (multiply charged spectra were searched twice, assuming 2+ or 3+ charge state, because the exact charge could not be determined), 4229 MS/MS spectra were assigned a peptide with probability above 0.9 (3457 and 730 assignments to spectra of doubly charged and triply charged precursor ions, respectively). This dataset was used to optimize the method for computing spectrum quality scores, and to test the accuracy of the method as a function of the dataset size.

2. The Arabidopsis dataset (unpublished) was acquired from four protein mixtures derived from cultured Arabidopsis thaliana cells. The crude extract was loaded on a 1D-SDS gel and single bands were cut out. Each band was digested with trypsin and peptides were sequenced by LC-MS/MS. In total, 3420 MS/MS spectra were submitted to a SEQUEST database search against a protein database of Arabidopsis and known contaminants allowing for semi-tryptic (tryptic at one terminus only) peptides, with a mass tolerance of 3 Da,
and with specifying carboxyamidomethylation of cysteines and methionine oxidation as variable modifications. This search resulted in 6720 peptide assignments (counting 2+/3+ duplicates). Of those, 924 were peptide assignments to spectra with a probability of being correct greater than 0.90, as computed by PeptideProphet. Additional database searches were performed using this dataset to test the ability of the method to recover high quality unassigned spectra.

3. The human lipid rafts dataset was taken from a large-scale quantitative proteomics experiment on lipid raft plasma membrane domains from human Jurkat T cells [Von Haller et al., 2003]. The full dataset is available from the PeptideAtlas MS/MS data repository [Desiere et al., 2005]. 9 LC/MS/MS runs (flow through sample, SCX fractions 31 through 39) were selected for the analysis in this work. The spectra, 12864 in total, were first searched using SEQUEST against the Human International Protein Index (IPI) database [Kersey et al., 2004], version v2.35, allowing for semi-tryptic peptides, with a mass tolerance of 3 Da, and allowing for methionine oxidation as a variable modification. This resulted in 24197 peptide assignments, counting 2+/3+ duplicates (711 singly charged, 12153 doubly charged and 12044 triply charged). Of those, 4171 were peptide assignments to spectra with a PeptideProphet probability of being correct $\geq 0.9$ (4034 peptide assignments to 2+/3+ spectra and 137 assignments to 1+ spectra). The results were further analyzed using ProteinProphet [Nesvizhskii et al., 2003], resulting in 315 proteins having a ProteinProphet probability above 0.9 identified by 660 unique peptides. The number of proteins was determined by counting the number of entries in the minimal list of proteins sufficient to explain all observed peptides [Nesvizhskii et al., 2003]. In calculating the number of assigned spectra, a spectrum was counted only if it was assigned a peptide corresponding to a protein with ProteinProphet probability greater than 0.9. To eliminate random matches to proteins correctly identified by other peptides, an additional peptide level constraint was applied: a spectrum was counted as assigned only if the peptide was identified (from this or another spectrum in the dataset) with PeptideProphet probability equal or greater than 0.5.

\[\text{www.peptideatlas.org}\]
3.2 Experimental Procedures and Datasets

3.2.2 Search Parameters

In order to determine what types of peptides generated the high quality spectra unassigned in the initial search, a subset of the initially unassigned MS/MS spectra was re-analyzed using a number of additional searches:

1. “Large mass tolerance” search: SEQUEST, semi-tryptic, 5 Da mass tolerance (larger mass tolerance compared with the initial 3 Da search).

2. “4+/5+ charge state” search: SEQUEST search, semi-tryptic, 3 Da mass tolerance, assuming 4+ or 5+ charge state.

3. “Pyro-Glu” search: SEQUEST, semi-tryptic, 3 Da mass tolerance, allowing for conversion of N-terminal glutamine to pyroglutamic acid (loss of 17 Da) as a variable modification. Due to a limitation of SEQUEST, the search was performed allowing for modified residues to be located anywhere in the sequence; peptides that did not contain glutamine at the N-terminus were filtered out at the data validation stage.

4. “Mascot” searches: Mascot, tryptic peptides only (note that Mascot tryptic search allows for removal of the initiating Met), 2 missed cleavages or less, 3 Da mass tolerance, allowing for N-terminal acetylation (+42 Da) or carbamylation (+43 Da) as a variable modification.

5. Miscellaneous searches: X! Tandem [Craig and Beavis, 2004] search allowing for more than one type of modification per peptide; SEQUEST and Mascot searches allowing for modifications not specified in the previous searches (e.g., conversion of N-terminal glutamic acid residues to pyroglutamic acid, phosphorylation, guanidation, etc.).

These searches were performed against the same version of the IPI database as the initial search (version 2.35). The results from different searches were processed using Interact and PeptideProphet and then manually scrutinized. Validated peptide sequences were imported in a relational SQL database, and the total number of assigned spectra, the number of unique peptides, and the minimum number of protein identifications sufficient to explain all identified peptides was calculated. In addition, the spectra were searched against a genomic database as described in the Results section.
3.3 Computational Method

3.3.1 Separation of Spectra into Peak List Subsets

Prior to computing spectrum features, two peak list subsets are extracted from each spectrum. The main reason for this step is a reduction of the number of noise peaks in the spectrum, whose presence can lower the discriminating power of some spectral features. The two subsets of peaks are extracted for each spectrum in parallel using the following approaches:

1. A signal intensity cutoff is applied using a robust percentile-based approach, creating the “high intensity” peak subset. To account for the significant variability of the fragment ion intensities across the spectrum, the spectrum is divided into 5 equally sized m/z sections. Within each section, the peaks are sorted according to their intensity, and the intensity at a given percentile is used as the cutoff value (i.e. the signal intensity at the 50th percentile would be equal to the median intensity of the peaks in a section). All peaks with intensity above the cutoff are assigned to the high intensity peak subset. The peak picking algorithms of some instruments generate an excessive number of (noise) peaks per spectrum. To cope with this, we additionally cap the number of high intensity peaks and allow not more than 400 peaks per 1000 Da into this subset.

2. A second, parallel attempt to extract signal peaks is based on isotope patterns. Most peaks in tandem mass spectra feature isotope patterns, which are evenly spaced peaks that occur because the elements that constitute proteins contain heavy isotopes. The frequencies of isotopes are well known, but there are five different elements, hydrogen, carbon, nitrogen, oxygen and sulfur, which each have their own distributions. One would have to convolute all these distributions to predict the isotope distribution exactly. Moreover, at the stage of deisotoping, the amino acid composition of the peptide is generally unknown. Therefore we choose a Poisson model with only one frequency parameter, confounding all the individual frequencies. The Poisson distribution is \( f(k; \lambda) = \frac{e^{-\lambda} \lambda^k}{k!} \) where \( k \) is the random variable and \( \lambda \) describes the expectation value of the number of heavy isotopes. We estimate a general expectation value by summing up the average masses of the 20 amino acids and dividing them by the sum of the 20 monoisotopic amino acid masses, which gives 1.000676. Monoisotopic means that the molecule exclusively consists of the lightest (most frequently occurring) isotopes. This means that
3.3 Computational Method

Figure 3.1: Poisson distribution modeling isotopes for 500 Da, 1000 Da and 1500 Da. One sees that the first “peak” is the highest for 500 Da and 1000 Da, but not for 1500 Da anymore.

For a monoisotopic mass of 1000 Da, the corresponding average mass is 1000.676 and the expected number of heavy isotopes is 0.676. The general expectation value multiplied by the fragment mass $m$ yields $\lambda$, i.e., $\lambda = m \cdot 1.000676$. The theoretical distribution (Figure 3.1) and the actual patterns in the spectrum are compared, and the chi-square test statistic is calculated to measure the quality of the fit. All peaks with a value below 20 are assigned to the deisotoped peak list subset. This is an experimental choice based on visual inspection of many spectra and the correspondence of the model with their isotope patterns. It is a relatively loose threshold appropriate for low mass resolution data where the observed isotope distributions of heavy ions tend to deviate substantially from the theoretical distributions. This can be explained by selection processes in the instrument where molecules of a particular $m/z$ ratio are retained for the fragmentation step.

Spectrum features such as sequence tags, complementary fragments, and neutral losses (described below) are calculated exclusively on the high intensity peak subset as defined by the percentile cutoff, whereas the general spectrum features are calculated using the full spectrum as well as the two peak list subsets (the high intensity peak subset and the deisotoped peak subset). The percentile cutoff was optimized separately for each spectrum feature as described in the Results section.
3.3 Computational Method

3.3.2 Spectrum Features

Several classes of spectrum features were introduced to measure the quality of MS/MS spectra. These features were designed to characterize the overall distribution of peaks in the spectrum (general spectrum features), as well as to detect certain patterns (sequence tags, complementary fragment ions, and neutral losses) expected to be present in high quality MS/MS spectra but absent in low quality spectra.

3.3.3 General Spectrum Features

Eight general descriptive features were calculated for each peak list subset:

1. Number of peaks, square root transformed
2. Arithmetic mean of the peak intensities, log transformed
3. Standard deviation of the peak intensities, log transformed
4. Smallest m/z-range containing 95% of the total peak intensity
5. Smallest m/z-range containing 50% of the total peak intensity
6. Total ion current (TIC) per m/z (TIC divided by feature 4), log transformed
7. Standard deviation of the consecutive m/z-gaps between all peaks, log transformed
8. Average number of neighbor peaks within a 2 Da interval around any peak

The log- or square root-transformation of some of the spectrum features was applied to obtain a more gaussian shape of the distributions and to minimize the variance across the spectra, and resulted in improved discriminating power of the composite spectrum quality classifier derived using the linear discriminant function approach (described in the next section). For each spectrum, the entire peak list and the two subsets (the high intensity peak subset and the deisotoped peak subset) were extracted and characterized by the 8 descriptive features, resulting in 24 general spectrum features. Clearly, many of these spectrum features are redundant, and there is a particularly high degree of correlation between those spectrum features that are derived by applying the same descriptive feature to the three different peak subsets. Nevertheless, having each spectrum feature computed in triplicate slightly improved the overall performance of the method.
3.3 Computational Method

Discussion of Log-Transformation

A more robust way of estimating the feature 2 above would be to apply the log transformation not to the mean of the peak intensity in a spectrum but rather to the individual peaks. However, we found that the classification behavior was very similar for both approaches. Let $n$ be the number of peaks in the spectrum, and $p_i \in P$ the corresponding peak intensities. Originally, we calculated the feature $f$ as follows:

$$f = \log \sum_{i=1}^{n} \frac{p_i}{n}$$  \hspace{1cm} (3.1)

whereas the alternative way of calculation is the following:

$$f = \frac{\sum_{i=1}^{n} \log p_i}{n}$$  \hspace{1cm} (3.2)

We compare the performance of both approaches on the *H. influenzae* dataset. We plot the distributions of the original log(mean())-feature on all three spectrum versions: all peaks, signal peaks, isotope peaks, and likewise for the mean(log())-feature (Figure 3.2). We find indeed some differences in the distributions, though they are not striking. The distribution of identified spectrum scores for the mean(log()) feature is a bit more compact. All other features are computed as usual, and a ROC curve is drawn for both approaches (Figure 3.3). The ROC curves and therefore the overall discrimination are nearly identical for both approaches.

3.3.4 Sequence Tags

A high quality peptide MS/MS spectrum should contain an extended series of peaks separated by the mass differences corresponding to one of the 20 amino acids. Thus, if a spectrum does not allow extraction of at least a short sequence tag (partial sequence) it is unlikely to be of high quality.

The sequence tag-based spectrum features are calculated using the high intensity peak subset only. The following four spectrum features measuring the presence of amino acid mass differences in the spectrum are computed:

1. The length of the longest sequence tag that can be extracted from the spectrum [Nesvizhskii et al., 2004]. The tags are extracted assuming that all peaks correspond to singly charged fragment ions from the same ion series (e.g., all $y^+$ or $b^+$ ions).

---

3The length of the sequence tag is implemented as an optional spectrum quality filter in SpectrumMill (http://www.chem.agilent.com)
Figure 3.2: The results for the new approach (left) and the old approach (right) are compared. The separation of the distributions of the identified and unidentified spectra of the test set is shown. Additionally, the distribution of all spectra confounded is shown, but it is of no importance here. The old and new distributions differ most for the peak set of “all peaks”, but by and large, they are fairly similar.
3.3 Computational Method

Comparison log(mean()) vs. mean(log())
Using all 40 features on 33926 spectra (haemophilus influenza)

Figure 3.3: The old and the new approach are compared with respect to the overall performance. The two ROC curves are nearly identical, the overall performance remains almost unchanged. The diamonds represent the new approach, mean(log()), the squares the old approach, log(mean()).
2. The average (per peak) length of all extracted sequence tags, calculated by summing up the lengths of the tags involving each peak in the high intensity subset of the spectrum and dividing it by the number of peaks.

3. The number of peak pairs corresponding to an amino acid mass difference (sequence tag of length one).

4. A derived version of the above feature, computed using the peak intensities as weighting factors to account for the increased likelihood of more intense peaks being true fragment ions. For every pair of peaks, the geometric mean of their intensities is calculated, and the spectrum feature is computed as a ratio of the sum of the geometric means of peak pairs corresponding to an amino acid mass, divided by the sum of the geometric means of all peak pairs.

In the case of peptides containing modifications, the sequence tags are bound to be interrupted by mass differences not corresponding to any of the standard amino acid residue masses. Although the masses of typical modifications can be included in the list of amino acid masses, this would lead to an increased number of peak pairs occurring at random because additional combinations between peaks would be admitted for these additional masses. The additional measures counting the isolated amino acid mass differences (features 3 and 4) are thus useful in the case of modified peptides where extraction of long sequence tags using the standard set of 20 amino acids might not be possible.

3.3.5 Complementary Fragment Ions and Neutral Losses

The fragmentation of multiply charged peptide ions via collision induced dissociation results in the charges being distributed among the resulting fragment ions. The doubly charged parent ions most often dissociate into two singly charged fragments, which leads to a high degree of symmetry in the spectrum. In contrast, the triply charged parent ions dissociate into two charged fragments, but the charges are unevenly distributed, with one fragment carrying two charges. When a precursor ion with the mass to charge (m/z) value \( m/z_p \) and charge \( z_p \) breaks down into two ions (in the case of ion trap mass spectrometers, primarily \( b \)-ions and \( y \)-ions), the overall mass and charge are preserved but split up between the two fragment ions, thus meeting the following criteria \cite{Perez2002, Sadygov2002}:

\[
mz_p \cdot z_p = mz_b \cdot z_b + mz_y \cdot z_y
\] (3.3)
3.3 Computational Method

\[ z_p = z_b + z_y \]  
\[(3.4)\]

\[ 0 < z_b < z_p \]  
\[(3.5)\]

\[ 0 < z_y < z_p \]  
\[(3.6)\]

Although the fragmentation process is strongly dependent on the sequence of the peptide and the resulting fragmentation patterns observed in MS/MS spectra can be quite complex, a high quality spectrum should still contain a significant number of complementary peak pairs (fragment ions whose masses add up to the mass of the precursor ion). Three spectrum features are computed (using the high intensity peak subset only) in the following way:

1. The number of complementary peak pairs satisfying the above constraints.

2. The number of complementary peak pairs (feature above) divided by the background count (the number of complementary pairs expected by chance). The background count is estimated by counting the number of peak pairs satisfying the same constraints mentioned above at 10 incorrect m/z values of the precursor ion. These incorrect m/z values are obtained by shifting the correct precursor ion mass by ± 10, 20, 30, 40, 50 Da. For each offset, the number of peak pairs is counted and the counts are then averaged, yielding the background count.

3. Similar to feature b, but with the background count subtracted from the number of complementary peak pairs instead of taking the ratio.

When the charge state of the precursor peptide is not known (e.g., multiply charged ions in the case of low mass resolution data), the number of complementary peaks is computed first assuming the 2+ charge state (in which case all observed fragment ions are expected to be singly charged), and then assuming the 3+ charge state (in which case both singly charged and doubly charged fragments are allowed). Then, the largest of the two counts is selected and used as the number of complementary peak pairs feature (feature 1), and the two derivative features (2 and 3) are computed as described. It should be noted that some of the spectrum features used here for quality assessment purposes, especially the number of complementary fragments and its derivatives, can be used to assist in the deter-
mination of the precursor ion charge state directly from the MS/MS spec-
trum [Perez et al., 2002; Sadygov et al., 2002]. In the case of high resolution
mass spectrometers, where the charge state can be accurately determined,
the entire analysis can be performed separately for spectra with different
precursor ion charge states.

### 3.3.6 Neutral Losses

Similarly, neutral losses such as loss of ammonia (loss of 17 Da), water (18
Da) and carbon monoxide (28 Da) are often found in MS/MS spectra of
peptide ions, see e.g. [Tabb et al., 2003b]. The presence of neutral losses in
the spectrum is useful for distinguishing peptide ion spectra from those of
contaminants or background noise. The original feature counts the num-
ber of peak pairs which meet the required offset. A mass tolerance of 0.5
Dalton is used. Two more features are derived from the original features,
as for the complementarity features. (b) One derived feature counts the
number of background hits to be expected as described for the comple-
mentarity feature background. The ratio of the original feature divided
by the background count is used as the feature. (c) In the other derived
feature, the background count is subtracted from the original feature. The
three approaches multiplied by the three offsets account for 9 features.

### 3.3.7 Discriminant Analysis

Improved discrimination between assigned and unassigned spectra (and
thus, to a large degree between high quality and low quality spectra) can
be achieved by combining all separate spectrum features into a single dis-
criminant score (the spectrum quality score, SQS). We used Fisher’s linear
discriminant analysis to calculate the discriminant score. It is a weighted
combination of the individual scores \( s_i \) described above, computed accord-

\[
SQS(s_1, s_2 \ldots s_n) = c_0 + \sum_{i=1}^{n} c_i s_i \tag{3.7}
\]

The constant \( c_0 \) and the weights \( c_i \) are derived in such a way that the ra-
tio of between-class variation to within-class variation is maximized under

---

4The classifier is trained to discriminate between the assigned and unassigned spectra (given the results of the initial database search). However, since assigned spectra tend to be of high quality, and the majority of unassigned spectra tend to be of low quality, the classifier developed this way can be used to discriminate between high quality and low quality spectra.
the assumption of multivariate normality. This assumption does not nec-
essarily hold for all of the spectrum features, and nonlinear classification
methods should in general allow better discrimination. Nevertheless, the
LDA analysis is a robust multivariate technique that has been proven to be
a valuable chemometrics tool in similar applications, e.g., the elucidation
of chemical structure of organic compounds from mass spectrometry data
[Varmuza and Werther][1996]. The main advantage of the LDA approach
and the rationale for its use in this work over more sophisticated nonlinear
methods was the ease of its implementation as a part of the dynamic
quality scoring algorithm.

Training Scenario

Deriving the discriminant function (classifier) requires a training dataset.
This training is done on the same experimental data that need to be classi-
ified, but for cross-validation purposes, the experimental data are first ran-
domly divided into a training dataset and a test dataset. In our tests, we
used a probability ratio of 1:1.

The positive and the negative labels are generated automatically from
the PeptideProphet database search confidence values which are parsed
together with the spectra. In our tests, identified spectra (PeptideProphet
confidence at least 0.9) were used as positive labels while unidentified
spectra (confidence at most 0.1) were used as negative labels. Other labels,
say scores from other search engines, are obviously an alternative. Indeed,
the approach has been successfully used on labels generated by PepSplice.

Based on the training dataset, the variance-covariance matrix and the
vector of the group differences are calculated, and the coefficients
\( c_i \) are computed using simple matrix algebra. The resulting score computed ac-
cording to the discriminant function is used as a final quality score instead
of the original numerical spectrum features.

The distributions for the individual features are always shown auto-
matically in the GUI version of the software, as well as a distribution of
the overall classification and a ROC curve. Spectra can be viewed in a
browser, and experienced mass spectrometrists are able to tell apart high
quality from low quality spectra. This gives users the opportunity to man-
ually double-check the performance of the method on their data.

Labels of Unidentified High Quality Spectra

The purpose of the approach is mainly to extract unidentified high quality
spectra from spectrum datasets. This poses a challenge in that these very
same target spectra are labeled as unidentified at the time of the training,
3.4 Results

However, by definition, unidentified high quality spectra are not known as such at the training stage. This is usually a minor problem since the number of unidentified spectra generally exceeds the number of identified spectra and a functioning classifier can be developed in spite of this challenge. Nevertheless, we tested how well the method copes with that challenge. We artificially generated known unidentified high quality spectra by sabotaging the database search process such that it could not identify many spectra that it could otherwise identify. We found that the method copes very well with these challenges. Even if half of the identifications were lost due to the sabotage, the method still worked reasonably well. (These tests are described in Section 3.4.3 and Figure 3.6). Moreover, the cross-validation clearly indicated a drop in the performance in agreement with the severity of the sabotage. This means that users of the software would immediately notice a lower than expected performance since the distributions are automatically generated for each run. They could then react to the problem by using a more conservative quality score cutoff and submitting a higher proportion of the spectra to the more resource-hungry searches.

3.4 Results

3.4.1 Overview of the Quality Assessment Method

First, acquired MS/MS spectra are searched against a protein sequence database using any of the currently available database search algorithms. The accuracy of peptide assignments is then evaluated statistically, e.g., using PeptideProphet [Keller et al., 2002]. Probabilities computed by that tool are used to automatically assemble a training dataset of ‘assigned’ spectra (spectra assigned a peptide with PeptideProphet probability of being correct above than 0.9) and ‘unassigned’ spectra (probability below 0.1). If the exact charge state of multiply charged precursor ions cannot be accurately determined (low mass accuracy instruments), the spectra are searched against the sequence database once for each potential charge state. Spectra are then added to the unassigned subset of the training set only if all peptide assignments to that spectrum (assuming all charge state possibilities) fail to pass the 0.1 probability threshold. The training set is created on the fly.

Second, for each MS/MS spectrum, a set of 40 spectral features (see Experimental Procedures for details) is calculated describing different aspects of the spectrum. These features were designed to quantify the overall distribution of peak positions (mass to charge [m/z] values) and peak in-
3.4 Results

tensities, the presence of certain peak pairs (e.g., neutral losses or pairs of y+ and b+ ion summing up to the measured precursor ion mass) expected to be present in high quality spectra, and the presence of sequence tags in the spectrum. Prior to computing the spectrum features, each MS/MS spectrum is separated into low and high intensity signal components (low and high intensity peak subsets). Some spectral features are computed using all peaks in the spectrum, whereas other features are computed using high intensity peaks only.

Third, all spectrum features are combined into a single spectrum quality score (see Experimental Procedures). The weighting factors, measuring the relative contribution of each spectrum feature toward the composite quality score, are determined using the linear discriminant analysis (LDA). Discriminant function coefficients are determined automatically for each analyzed dataset anew (“dynamic” quality scoring). Finally, the statistical classifier is applied to the spectra left unassigned in the initial search to find those that are of high quality. Such spectra are the most interesting candidates for subsequent re-analysis, since they are likely to contain valuable biological information that is worthwhile to be further explored.

3.4.2 Tuning and Testing of the Method

The computational method was first tested and tuned using several datasets. The tuning of the spectrum features was performed using the H. influenzae dataset. Spectrum features such as sequence tags, complementary fragment ions, and neutral losses were found to be sensitive to the peak intensity cutoff. As the peak intensity cutoff increases, an increasingly higher fraction of the noise peaks are eliminated, thus reducing the number of peak pairs and sequence tags occurring by chance. This, in turn, improves the discriminating power of the spectrum features. This trend, however, holds up only to a certain point, after which the performance starts degrading due to exclusion of true fragment ions from the high intensity peak subset. Figure 3.4A illustrates this in more detail for one particular feature, the length of the longest sequence tag that can be extracted from the spectrum. The plot shows the frequency of extracting a tag (the longest tag for each spectrum) of a certain length among assigned and unassigned spectra in the H. influenzae dataset when using all peaks in the spectrum, and after applying an 80% intensity cutoff (i.e., using only the top 20% of most intense peaks). For both assigned and unassigned spectra, the distributions are shifted toward shorter tag lengths when the sequence tags are

---

5Denoising of MS/MS spectra is also known to increase the sensitivity of the database search-based peptide identification process [Gentzel et al., 2003, Rejtar et al., 2004]
extracted from the high intensity peak subsets compared to using all peaks. The average length of the longest tag extracted from the assigned or unassigned spectra was around 15 and 6 amino acids, respectively, when the tags were extracted using all peaks. These numbers dropped to approximately 7 and 2 at the 80% intensity threshold. To find the optimal intensity cutoff, the significance of each spectrum feature was measured using the non-parametric two group comparison test (Mann-Whitney z score). Figure 3.4B shows the statistical significance of several spectrum features that are representative of each group (sequence tags, complementary fragments, and neutral losses) plotted as a function of the intensity cutoff. For each class of spectrum features, the discriminating power was evaluated and the parameters optimized. The plot demonstrates that in the case of sequence tag-based features, the optimal intensity cutoff (the cutoff resulting in maximum discriminating power) was around 80%.

A similar analysis was performed for other classes of spectrum features, and the intensity cutoffs of 70% and 50% for the complementary fragments and neutral losses, respectively, were chosen. In addition, similar trends were investigated for the mass tolerance parameter, and the tolerance of 0.7 Da was selected for computing the sequence tag-based features, 0.9 Da for the complementary fragments, and 0.5 Da for neutral losses. In general, these parameters should be tuned separately for each type of mass spectrometer. Ideally, the optimization process should be carried out using the same on the fly created training dataset used to determine the discriminant function coefficients in Eq. 2, thus making the method even more robust. However, preliminary tests indicated that slight changes in the model parameters did not significantly affect the overall performance of the spectrum quality classifier, and the implementation of the dynamical optimization process was deferred to future work.

The overall performance of the composite spectrum quality classifier was tested on the *H. influenzae* dataset using receiver operator characteristic (ROC) plots. To generate ROC curves, the discriminant function was developed using 50% of the data and applied to the remaining 50%. Figure 3.5A plots the spectrum quality score obtained by combining all spectrum features using the LDF approach, as well as scores computed using spectrum features from a single category only (e.g., using general spectrum features only). The plot demonstrates that by combining multiple different spectrum quality features into a composite score one can achieve better discrimination between assigned and unassigned spectra than by using any class of spectrum features alone. Many spectrum feature groups have high discrimination power individually, but the significance of each group varies. For example, the spectrum features describing the presence
Figure 3.4: Optimization of spectrum quality features. A, distribution of the length of the longest sequence tag among assigned (green) and unassigned (blue) spectra extracted at different peak intensity cutoffs, 80% (solid line) and 0% (using all peaks in the spectrum, dashes). Inset, mean length of the longest sequence tag extracted from the assigned (green) or unassigned (blue) spectra plotted as a function of the peak intensity cutoff. B, discriminating power (Mann-Whitney z score) of several representative spectrum features: the length of the longest sequence tag that can be extracted from the spectrum (blue, solid) the number of complementary fragments (magenta, dash-dot), and the number of peaks corresponding to the loss of water (green, dashes) computed as a function of the peak intensity cutoff.
Figure 3.5: Discriminating power of the spectrum quality classifier. A, ROC curves showing the discriminating power of the spectrum quality classifier computed using all features (black curve, squares) and all except general features (purple, triangles pointing right), as well as using features from each single feature group: general features (green, circles), complementary fragments (magenta, triangles pointing upward), sequence tags (blue, triangles pointing downward), and neutral losses (red, diamonds). B, ROC curves showing the performance of the classifier as a function of the training dataset size: 2029 assigned, 6747 unassigned (black curve, squares); 444 assigned, 1312 unassigned (green, circles); 111 assigned, 362 unassigned (magenta, triangles pointing upward); and 40 assigned, 142 unassigned (blue, triangles pointing downward).
of complementary fragment ions in the spectrum are the most discrimi-
native of all classes of spectrum features in the region of low sensitivity 
(below 0.5). This is the region of interest when the task is to find a rela-
tively small number of unassigned spectra of very high quality. At the 
same time, the spectrum features from the general category allow better 
discrimination in the region of high sensitivity, which is more relevant to 
the problem of filtering out low quality spectra prior to database searching 
without a significant loss in the number of peptide identifications.

Within each class of spectrum features, not all individual features con-
tributed equally to the discrimination. For example, using the length of 
the longest sequence tag extracted from the spectrum as the only sequence 
tag-based feature allowed almost as good overall discrimination as using 
all four features from that category combined. The effect of inclusion of re-
dundant or marginally discriminating features on the accuracy of the clas-
sifier was investigated by performing a step-wise analysis (addition of one 
new feature at each step), and ROC curves were plotted to measure the 
performance of the classifier at each step (data not shown). Importantly, 
addition of redundant features or features with low discriminating power 
did not negatively affect the performance of the classifier, but resulted in 
a slight improvement. Furthermore, the use of multiple related features 
made the method more robust toward unexpected variations in the spec-
trum properties. Based on these considerations, all 40 spectrum features 
were kept in the analysis.

Another important consideration is the minimum size of the dataset 
needed to realize the advantages of dynamic training. This was investi-
gated by applying the method to several *H. influenzae* datasets of decreas-
ing size. The results of this analysis are shown in Figure 3.5B. Overall, the 
plot demonstrates that the method can be used even with datasets of rela-
tively small size. As expected, the best results were achieved with large 
datasets consisting of more than several LC-MS/MS runs. However, even 
for a dataset consisting of a single LC-MS/MS run (with 111 and 362 as-
signed and unassigned spectra, respectively, in the dynamically created 
training dataset), the classification performance was satisfactory and only 
marginally worse than that observed for larger datasets. The accuracy of 
the classifier degraded only when applied to very small datasets (with less 
than 50 spectra assigned in the initial search). Thus, in a typical shot-
gun analysis of complex protein mixtures, in most cases even a single LC-
MS/MS run generated on current mass spectrometers should be sufficient 
to train the classifier because it is unlikely that it would contain fewer than 
several thousand spectra of which at least a few hundred spectra would be 
confidently identified in the initial database search. In case of very small
3.4 Results

datasets, rather than training the classifier dynamically, it would be better to use a larger dataset generated in another (but similar) experiment. The larger size of the training dataset would then compensate for the loss of accuracy due to the differences in the spectrum properties between the dataset of interest and the training dataset.

3.4.3 Evaluation of the Method

The ability of the method to effectively recover unassigned high quality spectra was evaluated using the Arabidopsis dataset. The same dataset was used to investigate the robustness of the method toward the presence of ‘misclassified’ (high quality) spectra in the unassigned subset of the training set, which was created on the fly. The tests were carried out in the following way: First, the spectra were searched against the full database, allowing for Met+16 as a variable modification. This is a standard search and was done for reference. The search parameters were then deliberately modified to prevent certain identifications and the same spectra as in the reference search were searched again. Some spectra could obviously not be identified anymore - we call these ‘masked’ spectra. The results are shown in Figure 3.6A. These flawed searches were performed in three different ways by making the following changes compared to the normal search:

1. Search not allowing for Met+16 modification, resulting in 94 masked spectra.

2. Search against a modified database created by removing sequences of four high abundance proteins, resulting in 122 masked spectra.

3. Search with the mass of serine statically increased by 5 Da, resulting in 402 masked spectra.

The spectrum quality classifier was calculated using the results of the flawed searches only. We expected the classifier to assign a high quality score to spectra whose identification had been prevented by the flawed search parameters. The performance of the method was then evaluated by comparing the quality score distribution of the masked spectra with the distribution of the assigned spectra (see Figure 3.6B, C and D). In the first two tests (Figure 3.6B, C), the quality score distribution of the masked spectra remained similar to the distribution of the assigned spectra. The performance was worse in the third case (the search with incorrect serine mass) where the number of high quality spectra in the unassigned subset of the dynamically created training dataset far exceeded the number of
3.4 Results

Figure 3.6: Evaluation of the ability of the method to detect unassigned high quality spectra. A, correlation between the probabilities of peptide assignments in the “normal” search (allowing for Met+16 as a variable modification) and in the “flawed” search (no modifications allowed). The performance of the classifier was evaluated by its ability to assign high quality scores to the masked spectra, which are located in the lower right corner of the plot. B, the distribution of spectrum quality scores among assigned (green, dash-dot), unassigned (blue, solid), and masked spectra (magenta, short dashes) in the first test: database search allowing for no modifications. C, same as above, second test: database search with the sequences of four high abundance proteins removed from the database. D, same as above, third test: database search with incorrect mass of serine residues.
assigned spectra, a situation that is unlikely to occur in practice (Figure 3.6).

Overall, the tests described above demonstrate that the approach should be able to detect unassigned high quality spectra based on their similarity to successfully identified spectra in any typical experiment. In those cases where it is expected that only a small fraction of all high quality spectra would be successfully assigned in the initial database search (e.g., proteomics studies using organisms with unsequenced or partially sequenced genome), a better approach would be to use a classifier trained on a different, more suitable dataset generated using the same type of mass spectrometer.

3.4.4 Applying the Method to a Human Lipid Rafts Dataset

The tests described in the previous section demonstrate that the method is able to recover high quality spectra left unassigned due to intentionally introduced flaws in the database search parameters or in the sequence database itself. However, for practical applications, it is more important to estimate how many high quality spectra are left unassigned in a typical analysis, to test whether the dynamic quality scoring method is able to recover them, and investigate what types of peptides produced those spectra. Answering these questions is also important for designing more efficient computational strategies for peptide identification.

To address these questions, a dataset of ion trap MS/MS data, a subset of a large human lipid rafts experiment [Von Haller et al., 2003], was chosen for more detailed analysis (see Experimental Procedures). The spectra were initially searched against the human IPI database (version 2.35) using SEQUEST with 3 Da precursor ion tolerance, allowing for tryptic and semi-tryptic peptides, and with variable Met +16 modification. These search parameters are equivalent to those implemented in the PeptideAtlas high throughput data analysis pipeline (31), and are quite typical for data generated using ion-trap mass instruments. In total, the initial database search resulted, after applying ProteinProphet [Nesvizhskii et al., 2003], in 315 proteins having a ProteinProphet probability above 0.9 (the minimal number of proteins sufficient to explain all observed peptides, see References [Nesvizhskii et al., 2003, Nesvizhskii and Aebersold, 2005]) identified by 653 unique peptides from 4172 assigned spectra, see Table 3.1.

Analysis of the database search results confirmed that this dataset was a typical example of a shotgun proteomic dataset of an intermediate size.

---

6Here the term ‘unique peptide’ is used to describe the identification of the same peptide sequence from one or more different MS/MS spectra.
Close to 30% of the proteins were identified by a single (but high probability) unique peptide in one charge state, and another 20% were identified by one unique peptide observed in two or more different charge states (the fraction of single peptide identifications would be smaller if the entire dataset of MS/MS spectra, and not only SCX fractions 31-39, were used in the analysis). Among all spectra assigned a peptide with high probability, approximately 2.5% percent were assignments to semi-tryptic peptides and 2% to peptides containing oxidized methionine, see Table 1. The sequences of approximately 65% of all peptides identified with high probability contained no missed trypsin cleavage sites, and only a small percentage of the peptides contained more than two missed cleavages. It is also worth mentioning here that no protein identifications would be lost if the search was limited to unmodified peptides only, and only 4 proteins would be lost (or statistical significance of the identification drop below the 0.9 ProteinProphet probability threshold) if the search was limited to unmodified tryptic peptides.

The results of this initial database search were used to train the spectrum quality classifier, which was then applied to all multiply charged spectra. Singly charged MS/MS spectra were removed from further analysis due to their relatively small contribution.

The distribution of quality scores for the entire dataset of multiply charged spectra, as well as for the assigned and unassigned subsets, is shown in Figure 3.7. The plot indicates that close to 80% of all unassigned MS/MS spectra are of relatively low quality; these spectra might not be worth additional efforts to reanalyze them. However, a significant number of unassigned spectra are actually of high quality, which manifests itself in the presence of an extended right tail in the quality score distribution for the unassigned spectra. Fitting the quality score distribution observed for the assigned spectra to match the extended right tail of the quality score distribution for the unassigned spectra indicates that it should be possible to extract at least 25% more peptide identifications in addition to those obtained in the initial database search (see Fig. 5).

In order to determine what types of peptides generated the high quality spectra unassigned in the initial search, 632 unassigned spectra having high computed spectrum quality score (SQS above 1), representing less than 5% of the original dataset, were selected for more comprehensive re-analysis. All searches were performed against the same version of the IPI database as the initial search (version 2.35). The overall database search strategy was to minimize the number of redundant searches while attempting to identify as many different peptides as possible (see Experimental Procedures for details). Of all the spectra submitted for re-analysis,
3.4 Results

Figure 3.7: Application of the method to the human lipid raft dataset. Distribution of quality scores plotted for all spectra from the dataset (black curve) and separately for the assigned (dash-dot) and unassigned (dashes) spectra based on the initial database search is shown. The distribution of scores for the unassigned spectra has an extended right tail indicating the presence of high quality spectra among them. The short dashes represent an estimated fraction of the unassigned spectra that are of high quality.
359 were assigned a peptide in one of the additional searches, and 6 were determined to be correctly assigned in the initial search (but with low PeptideProphet probability of less than 0.1). The list of the additional peptide assignments is given in the Supplementary Table S3. Comparing with the 1695 spectra of comparable quality (SQS above 1) that were assigned in the initial search, this represents a 20% increase in the number of assigned spectra. As expected, the higher the quality score of the initially unassigned spectrum was, the more likely it was that the spectrum would eventually be interpreted with high confidence after completion of all additional database searches (see Figure 3.8). For example, among 268 initially unassigned spectra having a quality score between 1 and 1.5, it was possible to identify only 129 spectra (48%), whereas it was possible to identify 22 out of the 28 spectra having a quality score above 3 (79%). This is consistent with the expected rates of spectrum assignments given the distributions of quality scores plotted in Figure 3.7.

The distribution of peptide assignments, obtained in the additional searches, among different categories is shown in Table 1. The search with large mass tolerance added 64 assigned spectra (46 unique peptides) that were missed in the initial search due to inaccurate measurement of the precursor ion mass (parent mass deviation of more than 3 Da). In addition, 20 MS/MS spectra (13 unique peptides) were arising from peptide ions with 4+ or 5+ charge states. Furthermore, 42 unmodified peptides were identified from 64 spectra with the help of two other database search tools, Mascot and X! Tandem, and by performing SEQUEST searches using a different set of search parameters than those used in the initial search (i.e., trypsin-constrained search and not allowing for any modifications).

In addition, a significant number of peptides containing modifications were identified, among which peptides with the N-terminal glutamine converted to pyroglutamic acid (116 spectra, 48 unique peptides) and N-terminal acetylation or carbamylation (68 spectra, 33 unique peptides) were the most common (in addition to the peptides containing oxidized methionine identified in the initial search). Other types of detected modifications included sodiation, tryptophan oxidation, lysine carbamylation, and histidine methylation (15 spectra, 8 unique peptides in total). Figure 3.9A shows one such example, the identification of a tryptic peptide YPHE-GIVTNWDDMEK from Actin, cytoplasmic 1 protein (Swiss-Prot: P02570) containing methylated histidine at position 5 in the peptide sequence (position 73 in the protein sequence), which is a known post-translational modification annotated in the Swiss-Prot database.

Although it is informative to know how many new spectra can be assigned by performing additional, more comprehensive database searches,
3.4 Results

Figure 3.8: Results of reanalysis of 632 high quality (SQS > 1) spectra unassigned in the initial search (human lipid raft dataset). The number of initially unassigned spectra that were eventually interpreted after completion of all additional database searches (black bars) and the number of spectra that remain assigned (white bars) as a function of spectrum quality are shown. Inset, fraction of initially unassigned spectra that got identified in one of the additional searches as a function of spectrum quality.
Figure 3.9: A, tandem mass spectrum supporting the identification of a tryptic peptide, YPIEHGIVTNWDDMEK, from Actin, cytoplasmic 1 protein (Swiss-Prot accession number P02570) containing methylated histidine at position 5 in the peptide sequence and oxidated methionine at position 14. B, tandem mass spectrum supporting the identification of a tryptic peptide, LQGSATAAEAQVGHQTAR, from a computationally predicted splice form of the Lck-interacting transmembrane adaptor 1 protein obtained by searching the unassigned high quality spectra against the EST database.
it is more important to know whether those additional assignments add any new information to the data obtained from the initial search. Although a significant number of unmodified peptides (101 unique peptides) were identified by performing the additional searches (large mass tolerance, 4+/5+ charge state, X! Tandem and Mascot searches), only 9 of those peptides were not identified from other spectra by SEQUEST in the initial search, resulting in 1 new protein identification and 1 identification with increased confidence. Similarly, most modified peptides were found to be arising from a small number of the most abundant sample proteins that were already identified on the basis of tryptic peptides seen in the unmodified form in the initial search. In fact, no new protein identification resulted from any peptides containing modifications other than conversion of N-terminal glutamine to pyroglutamic acid and N-terminal acetylation.

Of the 48 unique peptides (116 MS/MS spectra) identified containing N-terminal pyroglutamic acid, all except one were not previously seen in the unmodified form in this dataset. However, this modification is likely attributable to sample handling, because all of the identified peptides containing this modification were tryptic and not located at the N-terminus of the protein. It is known that pyroglutamic acid readily forms at the N-terminus of peptides beginning with glutamine when peptides are kept in acidic solution for 15 minutes or more, which is often the case in shotgun proteomic analysis. Nevertheless, identification of peptides containing this modification was informative: it resulted in 10 new protein identifications and 4 protein identifications with increased confidence (identified by unmodified peptides in the initial search but with ProteinProphet probability below 0.9). Eight of those proteins could not be identified with high confidence based on the results of the initial SEQUEST search even when the search was extended to include the entire human lipid rafts dataset of Ref. 30. These proteins include several membrane proteins that are likely to be relevant to the analysis of lipid rafts, e.g. KIAA0143 protein (IPI0009217, Swiss-Prot: Q14156). The accession numbers of those proteins and the sequences of their corresponding peptides can be found in the Supplementary Table S4, with one MS/MS spectrum for each of the identified peptides shown in the Supplementary Table S5.

Of the 33 unique peptides (68 MS/MS spectra) identified containing N-terminal acetylation or carbamylation, none were seen in the unmodified form in the selected dataset (SCX flow through sample, fractions 31 through 39 of the entire human lipid rafts dataset), resulting in 3 new pro-

---

7Due to low mass accuracy of the ion trap data, in most cases it was not possible to distinguish between N-terminal acetylation and carbamylation based on the MS/MS spectrum alone.
tein identifications and 3 identifications with increased confidence. However, additional analysis indicated that 21 of those peptides were also present in the unmodified form in other SCX fractions of the original dataset (flow through sample, SCX fractions 43 and higher). All of the peptides that were identified in both modified and unmodified form (considering all SCX fractions) were tryptic, located in the middle of the protein sequence, and were arising mostly from high abundance proteins. This strongly suggests that in those peptides the observed modification was a carbamylation, an artifact that occurred after protein digestion. N-terminal carbamylation is a common chemical modification known to occur when protein samples are digested in a solution containing urea, which was the case with the human lipid rafts dataset used in this work [Von Haller et al., 2003].

In contrast, of the 12 peptides that were identified in the modified form only (considering all SCX fractions), 10 were located at the N-terminus of their corresponding proteins starting with the residue immediately following the initiating methionine. In those 10 peptides the modification was likely an acetylation of the protein N-terminus that occurred prior to digestion. The sequences of those peptides started with either Ala (4 peptides), Ser (4), or Thr (2). As reviewed in Ref. 41, these residues are among the most frequently acetylated residues, with the frequency of acetylation of Ala and Ser close to 100%, and 80% in the case of Thr. For comparison, only 3 out of the 21 peptides identified in both modified and unmodified form (considering all SCX fractions) started with Ala, Ser, or Thr. The identification of N-acetylated peptides is important because they confirm the translation initiation start sites and assist in the determination of the mature form of the protein present in the sample [Nesvizhskii and Aebersold, 2005, Gevaert et al., 2003, Mann and Pandey, 2001]. It is also worth noting that 3 of the N-acetylated peptides identified here correspond to 3 low molecular weight proteins that could not be identified even when the initial SEQUEST search (i.e., search without looking for N-acetylated peptides) was extended to the entire human lipid rafts dataset, including two less characterized proteins containing predicted transmembrane domains: RER1 protein (IPI00005728, Swiss-Prot:O15258), and HSPC009 protein (IPI0002277, Swiss-Prot/TrEMBL: Q9Y2R0), see Tables S4 and S5.

All database searches described above were performed against the human IPI database. This database is often used in the analysis of shotgun proteomic data because it represents a good tradeoff between the completeness of protein sequences and the degree of sequence redundancy [Nesvizhskii and Aebersold, 2005]. In particular, the sequence- and identifier-based construction of the IPI database significantly reduces the need for manual filtering of redundant sequences while maintaining cross-references.
3.4 Results

to all its source data. However, minor sequence variants, e.g., polymorphisms, are not represented in the IPI database. Furthermore, this database is likely to be not complete with respect to alternative splicing. Thus, it was interesting to investigate if any of the high quality spectra unassigned in the initial search against the human IPI database could be assigned by searching more complete sequence databases.

To identify novel peptides or peptides containing sequence polymorphisms, the unassigned spectra were searched against a compressed human EST sequence database created and maintained by N. Edward. This database represents the minimum size sequence database containing length 30 amino-acid sequences from the human EST database six-frame translation [Edward and Lippert, 2004, Edward, 2005]. In order to reduce the number of entries with sequencing errors, or simply redundant entries, the database was constrained to contain only those sequences that appear in an open reading frame of at least 50 unambiguous amino-acids, and which appear at least twice in the database. Importantly, the compact nature of the database allowed an order of magnitude reduction in the database search time compared to the search time against the original EST database. More detailed information regarding the construction of this database can be found on the corresponding website.

The search against the EST database resulted in the identification of 7 peptides from 12 MS/MS spectra whose sequences were not present in the then current version of the IPI database used in the initial search (version 2.35). The sequences of 2 of those peptides are now present in the latest version of that database (version 3.01), a positive development indicating continuing improvement of the protein sequence databases by their developers and annotators. In addition, the EST database search resulted in the identification of several sequence variants of known proteins. For example, a tryptic peptide LSGLVFFPHLDK from Endonuclease G like 1 protein (Swiss-Prot Q9Y2C4) was identified from 3 MS/MS spectra, revealing a single nucleotide polymorphism (SNP), annotated in the dbSNP database [Kitts and Sherry, 2003] maintained by the National Center for Biotechnology Information, NCBI (V/G at the third position in the peptide sequence, refSNP ID: rs1141223). In another example, a tryptic peptide LQTASDESYKDPTNIQLSK was identified from 2 MS/MS spectra corresponding to an alternative transcript of the protein Spectrin alpha chain, brain (Swiss-Prot Q13813, isoform 2), annotated in Swiss-Prot but not present as a separate sequence entry in the human IPI database.

Importantly, the EST database search also resulted in the identification of two novel peptides (see Tables S4 and S5). One of the peptides, tryp-

http://www.umiacs.umd.edu/~nedwards/research/pepseqdb.html
tic peptide LQGSATAEAQVGHQTAR, is particularly interesting and de-
serves special attention. This peptide is present in 4 EST sequences and was
identified by SEQUEST (XCorr score of 2.8), and additionally confirmed
by Mascot (expectation value $5 \cdot 10^{-5}$), from one high quality MS/MS
spectrum (see Figure 3.9B). This intron-exon spanning peptide identifies
a novel splice variant of the Lck-interacting transmembrane adaptor 1 pro-
tein (LIME1). At the moment, the NCBI Entrez Protein sequence database
contains only the “standard” form of this protein, accession number NP_060276,
which was shown to be a raft-associated protein in several recent studies
[Brdickova et al., 2003; Hur et al., 2003]. However, the identified peptide is
present in several predicted alternative splice forms of LIME1 according to
the AceView gene models (November 2004 version). All these predicted
forms also contain a transmembrane domain and a putative raft-targeting
motif, suggesting that all of them can be raft-associated. To gain further in-
sight, the entire dataset (all SCX fractions from both the flow through and
ICAT experiments and without quality filtering) was researched against a
database of all predicted LIME1 splice forms created based on the AceView
gene models. This resulted in the identification of another MS/MS spec-
trum produced by the same peptide, but in a different charge state, thus
further increasing the confidence in this identification. However, no pep-
tides from the “standard” form (or any other predicted form) were iden-
tified, suggesting that different forms of this protein can be expressed in
different cell types. Although comprehensive follow-up analysis of these
and other interesting findings was beyond the scope of this work, these
results clearly illustrate that new biological insight can be gained by re-
analysis of existing datasets. They also demonstrate the potential of mass
spectrometry-derived data as a new source of information useful for vali-
dation and annotation of the genome [Desiere et al., 2005; Nesvizhskii and
Aebersold, 2005; Mann and Pandey, 2001; Kuster et al., 2001; Choudhary
et al., 2001; McGowan et al., 2004].

3.5 Discussion

The quality of MS/MS spectra in shotgun proteomic experiments depends
on multiple factors, including the complexity of the sample, the type of
mass spectrometer and the fragmentation parameters used, see e.g. [Wen-
ner and Lynn, 2004]. Moreover, even run to run variations in the signal to
noise ratio in MS/MS spectra acquired on the same instrument can be sig-
nificant. The dynamic quality scoring method developed in this work can

http://www.ncbi.nih.gov/IEB/Research/Aembly
automatically adapt, to a significant degree, to variations caused by acquisition methods or instrument to instrument differences. It is well suited to the problem of finding high quality spectra left unassigned after the initial search of all spectra (with no loss of potential identifications due to filtering prior to database searching).

The basic idea behind the dynamic quality assessment method is that given the information on what spectra got assigned a peptide with high or low confidence during the first pass through the data, the notion of what constitutes a bad or good quality spectrum can be learned from the analyzed data itself, i.e., without relying on a training dataset of spectra taken from a different experiment. The statistical classifier is thus automatically developed for each dataset anew, making the method more robust toward variations in the MS/MS data than the static classifiers. Importantly, it does not require creating training datasets using manual expert ranking [Nesvizhskii, 2001; Xu et al., 2005], which is time consuming and complicates adoption of the method to different instrument types, or generation of MS/MS data using samples of purified proteins [Bern et al., 2004; Purvine et al., 2004], in which case the training dataset may not accurately represent the complexity of MS/MS spectra observed in a shotgun proteomic experiment using complex protein samples. It also incorporates a larger set of spectral features than any other recently described spectrum quality scoring methods [Moore et al., 2000; Nesvizhskii, 2001; Bern et al., 2004; Purvine et al., 2004; Savitski et al., 2005].

In addition to the development and testing of the MS/MS quality assessment method, a significant effort in this work was devoted to the reanalysis of high quality spectra left unassigned in a typical shotgun proteomics experiment. The results of those analyses should be useful for designing more efficient peptide identification strategies, especially those concerned with the identification of modified peptides. Allowing for multiple variable modifications of amino acids when searching spectra against a protein sequence database significantly increases the number of candidate peptides in the database that must be evaluated. Depending on the frequency of occurrence of the amino acid, the increase in computation time often becomes prohibitive. In addition, the potential for false positives also increases. The analysis performed here generally confirms the validity of the subset database search approach implemented in several existing database searching tools, including X! Tandem, Mascot, and SpectrumMill. Many of the peptides containing modifications identified in the human lipid rafts dataset corresponded to a relatively small number of high abundance proteins identified with sufficiently high confidence by at least one unmodified peptide in the initial search. Similarly, the dominant
majority of the identified semi-tryptic peptides were fragments of tryptic peptides also identified in the dataset, indicating that they are the product of in-source fragmentation. This suggests that in some cases it should be sufficient to search for peptides containing certain types of modifications (e.g., oxidation or sodiation), and for semi-tryptic peptides using a subset database comprised of proteins already identified in the initial, fast (tryptic peptides only, no modifications) scan through the data, thereby reducing the computational time. The same subset database search approach should be sufficient for re-analysis of MS/MS spectra in an attempt to interpret the spectra arising from peptide ions with 4+ or higher charge states, or spectra with inaccurately determined precursor ion mass.

At the same time, the subset database search approach would likely be less successful in identifying peptides containing certain types of modifications, e.g., peptides with N-terminal acetylation. The strategy described in this work enables efficient analysis without limiting the protein search space, since more comprehensive re-analysis is carried out only on those spectra that are of high quality and remain unassigned after the initial search. Both strategies, spectrum quality filtering and subset database searching, can be combined for even better computational efficiency. Quality scoring can also be combined with spectrum clustering [Beer et al., 2004; Tabb et al., 2005] to achieve even higher reduction in the number of spectra that need to be analyzed in each experiment. However, the development of efficient iterative database search strategies also requires new statistical models for combining peptide identifications obtained on different iterations and for computing accurate probabilities of their corresponding proteins.

It was also observed, in the human lipid rafts dataset used here and in other similar datasets, that in most cases where a peptide containing a polymorphism was identified with high confidence, one or more related proteins (often the main variant of that protein) were already identified in the initial search against the IPI database. Similar to the subset database approach for the identification of modified peptides, discussed above, one can largely limit the search for peptides containing polymorphisms to searching only those translated EST sequences that are homologues to proteins already identified by one or more peptides in the initial search against more compact databases such as IPI. Since novel or less characterized gene products are naturally of lower abundance relative to other proteins in the sample, or have low molecular weight, many such proteins are likely to be identified by a single peptide.

The computational strategy that implements quality assessment and filtering, as described in this work, is particularly useful for the identifi-
cation of sequence polymorphisms and novel proteins. Searching large numbers of MS/MS spectra against genomic databases requires significant computational resources. Even more importantly, searching unfiltered (i.e., including low quality spectra) datasets against such databases invariably leads to high false identification rates simply due to increased database search size [Resing et al. 2004] [Colinge et al. 2005]. At the same time, even the basic follow-up analysis, not to mention the biological validation, of each identification of a novel peptide to understand its nature and significance requires querying multiple databases, performing protein and DNA sequence alignment etc., and is very time consuming. Thus, large genomic databases, including EST databases, should be searched using MS/MS spectra of sufficiently high quality to minimize the number of incorrect identifications and to reduce the efforts spent on the interpretation of those erroneous findings.

In addition, in the case of low mass accuracy mass spectrometry data, such as ion trap data used in this work, unambiguous peptide identification is often not possible due to inability to resolve among different sequence alternatives (e.g., Ile/Leu, Asp/Asn, Glu/Gln/Lys etc.). For example, it was observed that many MS/MS spectra with inaccurately determined precursor ion mass (mass error of more than 3 Da, which prevented their identification in the initial search) matched with high scores to incorrect (but highly homologous) peptides in the EST search. In addition, the EST search resulted in multiple peptide assignments to spectra that appeared to be identification of sequence polymorphisms, when in fact the true peptides that gave rise to those spectra were high abundance peptides containing a chemical modification. For example, when the unassigned high quality spectra in the human raft dataset were searched against the EST database, peptide TGTIVMDSGHVTHTVPiyEGYALPHAILR was assigned to multiple MS/MS spectra with high scores, suggesting a polymorphism at position 10 in the peptide sequence (Asp/His substitution). However, the normal variant of this peptide (with Asp instead of His) was identified from more 40 MS/MS spectra in the initial search, and thus by far the most plausible explanation for the observed 22 Da mass difference is sodiation of the aspartate residue (addition of 22 Da to Asp, which makes it indistinguishable from His) rather than a point mutation. The abundance of such examples observed in this work warrants a general warning that, before reporting an “identification” of a polymorphism or a novel peptide based on low mass accuracy data, it is necessary to eliminate the spectra for which a more likely explanation can be found (e.g., chemical modification of a high abundance peptide).

Application of multiple database search engines to the same dataset is
not a new concept and is known to result in improved performance, see e.g. [Croft, 2000]. The concept of meta-analysis has been discussed in the context of MS/MS database searching as well [Resing et al., 2004] [Kapp et al., 2005]. In this work, the re-analysis of the unassigned high quality spectra using multiple database search tools allowed identification of additional peptides and protein, although the overall increase in the number of proteins was not significant. A larger benefit of combining the results of multiple search engines could be in the improved discrimination between correct and incorrect identification, leading to an increase in the number of protein identifications at certain fixed error rate. Still, since the application of multiple database search tools increases the overall database search time, in many applications it should be sufficient to apply additional database search tools only to the remaining unassigned high quality spectra after performing the initial search using a single database search tool.

More comprehensive interrogation of high quality unassigned spectra performed in this work was largely limited to database searching using tools that are currently available in most proteomics laboratories. However, a number of other approaches have been recently described that could be better suited for the identification of modified peptides or sequence polymorphisms. These methods include de novo sequencing [Dancik et al., 1999] [Taylor and Johnson, 2001] [Chen et al., 2001] [Ma et al., 2003] [Fischer et al., 2004] [2005], reviewed in [Standing, 2003], sequence-tag based error-tolerant database searching [Russell et al., 2004] [Tabb et al., 2003a] [Frank et al., 2005], and other approaches [Hernandez et al., 2003] [Liska and Shevchenko, 2003] [Searle et al., 2004] [Hansen et al., 2005] [Salek et al., 2005]. These computational algorithms can be used in conjunction with the quality scoring method described here as well. The human lipid rafts dataset and the results of its comprehensive analysis such as the list of peptide and protein identifications, reported in this work, can serve as a useful test dataset for the development and benchmarking of new and improved peptide identification strategies. For example, it will be interesting to apply those tools to the high quality spectra from the human lipid rafts dataset that still remain unassigned after all the additional efforts. Moreover, the analysis performed here with regard to the extraction of sequence tags from MS/MS spectra should be useful for optimizing peptide identifications methods involving extraction of sequence tags and their use as low-pass database filters [Tabb et al., 2003a] [Frank et al., 2005] for spectrum library searching [Stein et al., 2005], or for any other applications involving the separation of true fragments from noise peaks in the spectra.

The main aim of this work was to develop a robust approach for finding high quality spectra left unassigned in the initial database search, and to in-
investigate what prevented their identification. A different application of the quality scoring method is simply to filter out low quality MS/MS spectra prior to database searching to reduce the computational time spent on the initial search [Moore et al. 2000]. This would be a particularly attractive option in the case of MS/MS datasets generated using mass spectrometers such as LTQ, which provide much higher data acquisition rate at the expense of producing high numbers of low quality spectra. To use it for that purpose, the spectrum quality classifier can first be trained using a dataset of already searched MS/MS spectra acquired under similar experimental and acquisition conditions, and then applied to all newly acquired spectra prior to database searching (without retraining the classifier). In a large-scale study, a simple approach would be to search all spectra acquired in a few LC-MS/MS runs, train the classifier, and then apply it to all remaining spectra prior to database searching.

The choice of the filtering threshold is important for all practical applications, and requires further discussion. The quality score threshold selected in this work (SQS > 1) represented a reasonable trade-off between the sensitivity and specificity of the quality score-based filtering, i.e., elimination of more than 95% of all spectra at the expense of losing less than 25% of additional peptide identifications that potentially can be extracted from the remaining unassigned spectra. The distribution of quality scores, however, is dataset-dependent, reflecting, among other factors, the overall proportion of high quality spectra in the dataset. Furthermore, depending on the goals of the analysis, it might be desirable to filter the data with different stringency. The selection of the filtering threshold can be facilitated by the conversion of the spectrum quality score SQS into a probability that the spectrum belongs to the class of good quality spectra, $p_{\text{good}}$. This conversion can be performed by modeling the observed distributions of spectrum quality scores as a mixture of the two underlying distributions constructed separately for each class (good and bad quality spectra), and then using those distributions to convert the spectrum quality scores into class probabilities using Bayes’ formula. The probabilities computed this way are more transparent than the somewhat arbitrary spectrum quality scores SQS, and allow easy estimation of the sensitivity and specificity rates resulting from filtering the data using any minimum $p_{\text{good}}$ value as a filtering threshold.

Some of the spectrum features, introduced in this work, can also be used as additional parameters to validate assignments of peptides to MS/MS spectra. Computational tools such as PeptideProphet and ProteinProphet, used in this work, already employ an extended set of parameters in addition to the database search scores. As a result, the probabilities computed
by those tools allow good discrimination between correct and incorrect
identifications. Nevertheless, those tools should benefit from the inclusion
of certain spectrum features in their statistical model.

The calculation of spectrum quality scores in this work was based on
40 spectrum features, many of which are redundant or only marginally
discriminative. It is possible that the number of spectrum features can
be reduced without a significant decrease in the accuracy of the compos-
ite spectrum quality classifier. This question should be investigated more
thoroughly in future work using a more diverse set of experimental data
(different instrument types and experimental conditions). If some of
the spectrum features are indeed unnecessary under all conditions, they can
be easily removed from the analysis if desired. However, the presence of
redundant features does not create a problem for the method, and the in-
crease in the computational time is insignificant. Furthermore, new fea-
tures, not considered in this work but possibly developed by others, could
be easily added.

3.6 Conclusion

We developed a computational method for finding high quality spectra
left unidentified in large shotgun proteomic datasets under most common
conditions. The method does not require training datasets created using
time consuming manual ranking of MS/MS spectra, or datasets generated
using mixtures of purified proteins that might not adequately reflect the
complexity of real data. Instead, the quality scoring classifier is developed
dynamically using the results of the initial database search pass through
the data. This ensures the robustness of the method toward variations in
the properties of MS/MS spectra caused by the differences in acquisition
methods or instrument to instrument variability.

The practical utility of the method was investigated using a dataset of
spectra from a human lipid rafts experiment. Comprehensive computa-
tional re-analysis of the high quality spectra left unassigned in the initial
search resulted in a 20% increase in the number of identified high quality
spectra, including identifications of numerous modified peptides. Further-
more, searching unassigned high quality spectra against the EST database
led to the identification of peptides containing sequence polymorphisms
and novel peptides, including a peptide confirming a computationally pre-
dicted splice form. It is thus expected that the strategy explored in this
work will play an increasingly important role in the analysis of shotgun
proteomics data, and will help to realize the long touted potential of shot-
3.6 Conclusion

<table>
<thead>
<tr>
<th>Category</th>
<th>Spectra</th>
<th>Peptides</th>
<th>New peptides</th>
<th>New proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tryptic</td>
<td>3552 (1591)</td>
<td>627(476)</td>
<td>627(476)</td>
<td>311</td>
</tr>
<tr>
<td>Semi-tryptic</td>
<td>81(48)</td>
<td>21(16)</td>
<td>21(16)</td>
<td>4</td>
</tr>
<tr>
<td>Met+16</td>
<td>92(56)</td>
<td>34(25)</td>
<td>4(3)</td>
<td>0</td>
</tr>
<tr>
<td>Mass deviation &gt; 3 Da</td>
<td>64</td>
<td>46</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Charge states +4/+5</td>
<td>20</td>
<td>13</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Alternative search tools</td>
<td>64</td>
<td>42</td>
<td>6</td>
<td>2(1)</td>
</tr>
<tr>
<td>N-term. Gln to pyro-Glu</td>
<td>116</td>
<td>48</td>
<td>34(33)</td>
<td>14(8)</td>
</tr>
<tr>
<td>N-term. acetyl/carbamyl</td>
<td>68</td>
<td>33</td>
<td>33(12)</td>
<td>6(3)</td>
</tr>
<tr>
<td>Other modifications</td>
<td>15</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EST search</td>
<td>12</td>
<td>7</td>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3.1: The results of the initial SEQUEST search are shown separately for all spectra and for the subset of spectra with spectrum quality score (SQS) above 1 (in brackets). 4034 spectra were assigned in the initial search, of which 1695 had a SQS above 1. Also listed are the number of previously unseed peptides (starting with fully tryptic peptides identified in the initial search) and the number of resulting protein identifications based on peptides from each category. The number of new proteins resulting from additional searches includes proteins identified in the initial search but with ProteinProphet probability below 0.9.

gun proteomics data as a rich source of information useful for the validation and annotation of genomes.
Bibliography


Riko Jacob. Personal communication about lower bound justifying sortedness assumption, 2006.


Curriculum Vitae

Franz Felix Roos

Date of birth: February 8, 1977
Place of birth: Uznach SG, Switzerland
Citizen of: Zürich and Romoos LU, Switzerland

Education:

2003 – 2006: PhD student at Institute of Theoretical Computer Science ETH Zurich, Switzerland
Advisor: Prof. Dr. Peter Widmayer
Academic title: Dr. sc. ETH

1998 – 2002: University studies in biotechnology at ETH Zurich
Academic title: Dipl. Natw. ETH

1997 – 1998: Civilian Service in Geneva (13 months)
Cambridge Proficiency in Cheltenham, UK
Diplôme Supérieur de l’Alliance Française

1992 – 1997: High school in Zurich, Switzerland
Degree: Matura with focus on economics

1983 – 1992: Primary and secondary school in Volketswil ZH, Switzerland