The Importance of Being Proportional: A Paradigm Shift for Intensity-based Label Free Relative Quantification in Mass Spectrometry Proteomics

A DISSERTATION

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Susan Kaye Van Riper

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Doctor of Philosophy

Dr. John V. Carlis & Dr. Timothy J. Griffin

May, 2013

Acknowledgements

I would like to thank all people who have helped and inspired me during my doctoral studies culminating in this disseration. Without the continued support of family, friends, colleagues, and advisors, the completion of this dissertation would not be possible.

First and foremost, I offer my deep gratitude to John Carlis, my primary Ph.D. advisor and mentor for life. Eight years ago, he inspired me to alter my life's plans, to leave industry and pursue a career in academia. His leadership in interdisciplinary work fostered an environment to embrace the "joyful struggle" of interdisciplinary research and writing. Without his unwaivering support, I may not have perservered this long and arduous journey. I also offer my sincere gratititude to Tim Griffin, my Ph.D. co-advisor. Tim introduced me to proteomics and allowed me to embed in his biochemistry laboratory. Without Tim's and John's encouragement, patience, and support, I would not have been able to transform from a computer scientist to an interediscplinary proteomics researcher on the prescipice of a career in academia.

In addition to my advisors, I am indebted to other members of my dissertation committee, Chad Myers and Nelson Rhodus. Their encouragement, support, and feedback have been invaluable.

I would like to thank my lab-mates, both past and present, for their friendship and support. Thank you to Sri Bandhakavi, Matt Stone, and Ebbing de Jong for their patience answering all my questions and for teaching me how a biochemistry lab works. Thanks also to Getiria Onsongo and Daniel Feldman for helping me navigate the first few years; to Joel Kooren and Rich Beck for using RIPPER/PIN in their own work and

providing valuable feedback; and to Maggie Mahan for supplying friendship, humor, and encouragement.

I especially want to thank Ebbing de Jong and LeeAnn Higgins, for sharing their knowledge and insights, without which my venture into quantitative proteomics would not have been possible. Thank you also to Katie Doraschak and John Chilton for contributing modules to the software.

I would also like to thank those people who inspired me and encouraged me to pursue and complete my Ph.D. Kelly Albrant and Lyndon Stinson for being my lifelong friends, Josh Richard, my inspiration for not settling for the status quo, and Susan Jack, for keeping me sane.

Finally, I would like to recognize and thank my family. My parents, Lee and Marlene Ploetz, who instilled in me the work ethic, the will, the perseverence, and the courage to take on such a pursuit. I owe everything that I am to you. My sister, Sandy Diercks, who is always there for me. And above all, I give my deepest gratitude to Tracy Peterson, my partner in life, my most fervent supporter, and my rock.

Dedication

For Bob, you are forever with me.

Abstract

Background: Researchers conduct discovery-based studies to find biological variation that not only provide insight into the molecular machinery of disease progression, but accurately inform clinicians about a patient's health status, both current and future. Researchers discover biological variation by conducting large scale comparative studies and detecting differences in the molecular makeup (biomarkers) of healthy and diseased cells. Ideally suited for biomarkers are proteins because their cellular composition (proteome) and their degraded parts, endogenous peptides (peptidome) change in response to disease by creating new proteins, modifying existing proteins, and degrading proteins into endogenous peptides.

Increasingly, researchers employ high performance liquid chromatography, coupled with electrospray ionization tandem mass spectrometry (HPLC-ESI-MS/MS) for proteomics and peptidomics. Unfortunately, while HPLC-ESI-MS/MS biological discovery has vast potential, the potential remains unmet. For discovering biological variation in complex biological samples, for example, biomarker discovery, intensity-based label free relative quantification (iLFRQ) is desirable. When compared to labeled relative quantification, iLFRQ is more cost effective and does not limit the number of samples analyzed in a single experiment. Unfortunately, iLFRQ for proteins, and especially peptides, is difficult. Here, I list three challenges facing researchers employing iLRFQ HPLC-ESI-MS/MS for comparative proteomic and peptidomic studies. 1) The current relative abundance paradigm for iLFRQ is ill-suited to detect biological variation. This paradigm asks the question, "Are the constituent peptides differentially abundant?" To answer this question, researchers select peptides with intensity fold-changes greater than some threshold between two HPLC-ESI-MS/MS runs. Unfortunately, systematic differences, for example, sample amount, distort MS^1 peptide signal measurements and

is, therefore, especially problematic for iLFRQ. 2) HPLC-ESI-MS/ MS analyses produce poorly repeatable and reproducible results, primarily due to extraneous variability. While current normalization methods work well to mitigate global extraneous variability (systematic bias) in HPLC-ESI-MS/MS measurements, they fail to mitigate localized extraneous variability (complex variability in measurements) from transient stochastic events occurring during an HPLC-ESI-MS/MS run. 3) Finally, current software frameworks report protein level quantification rather than peptide level quantification. This limits a researcher's ability to conduct quantitative peptidomics studies. With these three iLFRQ challenges in place, researchers must conduct extensive hypothesis-driven experiments to weed out excessive false positive results. Worse, researchers miss real biological variation because they do not see false negatives. As a result, researchers can draw incorrect conclusions about biological variability and thus miss key insights.

Contributions: To remove these challenges, I offer three contributions. 1) The proportionality paradigm for iLFRQ, in contrast to the relative abundance paradigm, asks the question, "Are the constituent peptides differentially proportional?" Researchers answer this question by computing a peptide's proportional abundance and then detecting their statistically significant differences across multiple samples. 2) Proximity-based Intensity Normalization (PIN) is an embodiment of the proportionality paradigm that normalizes a peptide's signal intensity measured via HPLC-ESI-MS/MS by constructing its temporal neighborhood and then computing its relative proportion within that neighborhood. 3) A new software framework named RIPPER reports normalized peptide signal intensities rather than protein intensities. Optionally, researchers can match peptide signals to third party software peptide and protein identifications. RIPPER is available at https://z.umn.edu/ripper.

Evaluation: I evaluated the Proportionality Paradigm for iLFRQ, PIN, and RIPPER using datasets from HPLC-ESI-MS/MS analyses of complex peptide mixtures. PIN dominates current normalization methods in reducing systematic bias and complex variability. Furthermore, it finds statistically significant biological variation which is otherwise falsely reported or missed when using current normalization methods.

Discussion: Given that the relative abundance paradigm is ill-suited for discovering biological variation in complex biological samples via HPLC-ESI-MS/MS workflows, I contend that researchers must shift to the proportionality paradigm. Shifting paradigms will improve repeatability and reproducibility revealing biological variation (candidate biomarkers). Furthermore, the proportionality paradigm is widely applicable to many 'omics fields, especially peptide-centric fields using HPLC-ESI-MS/MS, for example, lipidomics, glycomics, and metabolomics. Therefore, I expect the Proportionality Paradigm for iLFRQ, embodied in PIN, and implemented in RIPPER, to change the way researchers analyze HPLC-ESI-MS/MS experimental data. Furthermore, HPLC-ESI-MS/MS biological discovery will be much closer to meeting its vast unmet potential. The upshot will, I expect, be reproducibility and repeatability improved, and otherwise falsely reported or missed, statistically significant biological variation discovered.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Variations of chemical behviour are everywhere present in minor degrees and just as no two individuals of a species are absolutely identical in bodily structure, neither are their chemical process carried out at exactly the same times. – Sir Archibold Garrod, 1902

1.1 Overview

Disease is a natural process in living organisms but at the cost of pain, suffering, and death. Of the approximately 2.5 million deaths in the United States in 2011, the Center for Disease Control and Prevention (CDC) reported that the leading causes of deaths continue to be heart disease (∼600,000) and cancer (∼575,000) [1]. While the incidence rate of cancer continues to grow primarily due to the aging population, deaths from cancer are now in decline. Among adults diagnosed with cancer during the period from 1974 through 1976, the 5-year relative survival rate for all cancers combined was 50%. According to the latest data available, for adults diagnosed with cancer in 2007, the 5-year relative survival rate for all cancers combined is now nearly 68% [2]. This dramatic increase in survival rates of the last 30 years is largely due to improvements in treatment and earlier diagnosis [2].

Unfortunately, for those patients suffering from disease, inadequate, or worse, no clinical assays for detection and diagnosis, the development of diagnostic tests and their translation to a clinical setting is a long and arduous task which takes years, even two

or more decades to complete [3]. For example, in order to predict cancer progression, researchers must discover biomarkers, that is, biomolecules that can serve as markers of biological state [4]. The presence, absence, or abundance of these markers not only provides insight into the molecular machinery of disease progression, but accurately informs clinicians about a patient's health status, both current and future. As shown in Figure 1.1, the result of this discovery process is a list of candidate biomarkers, which must be validated prior to the development of a biomarker assay. Finally, extensive clinical trials must be conducted to determine their efficacy in disease diagnosis and prognosis prior to clinical translation. Reducing the amount of time it takes to get a candidate biomarker from bench to bedside is critical for saving lives and reducing pain and suffering.

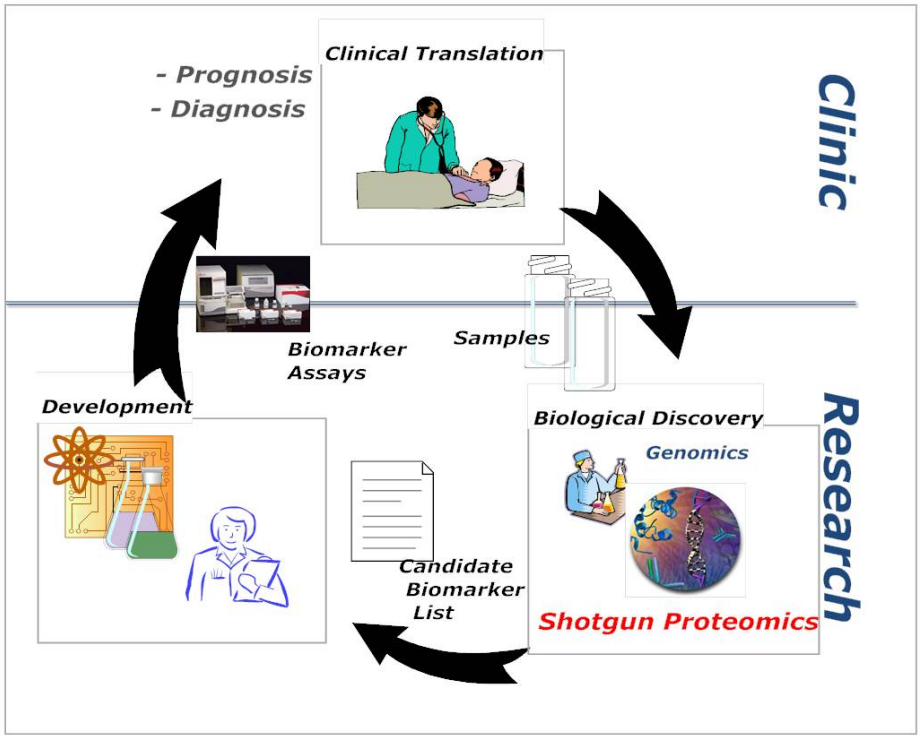


Figure 1.1: Biomarker discovery to clinical translation is a cycle that can take more then two decades to complete.

1.2 From Biological Variation to Biomarkers

Researchers discover biomarkers by studying biological variation. Biological variation offers researchers insights into the molecular machinery of biological processes. The 'central dogma' of molecular biology states that, for biological processes, biological information flows from double-stranded DNA (the 'recipe of life'), to single-stranded RNA, to protein. In this dogma, DNA serves as the blueprint for building the cellular machinery, and its study can reveal potential biomarkers.

In 1990, the National Institutes of Health (NIH) and the Department of Energy (DOE) joined with international partners in a quest to sequence the human genome (the complete set of DNA in the human body). They named this project the Human Genome Project (HGP) [5]. The first working draft of the human genome was released in 2000, just 10 years after the project's initiation. In 2003, the HGP was declared complete.

By all reasonable measures, the HGP was a phenomenal success, ushering in the post-genome era, the age of 'omics and informatics. In addition to sequencing the human genome, the HGP revolutionized molecular biology research. The complexity of the HGP project necessitated the development of multifaceted, interdisciplinary approaches where researchers employ high throughput instruments to collect data at the molecular level. Analyzing this data with advanced computational tools became collectively known as informatics.

So, if the genome and their constituent genes are the 'recipe for life', why not just study genes? After all, many researchers describe cancer as a series of genetic mutations that lead to uncontrolled cell growth. Furthermore, a single, inherited, gene mutation can cause premature death [6]. The epitome of this phenomenon is Huntington's disease [7]. It lies dormant through childhood and adolescence but becomes symptomatic during early to middle adulthood. Huntington's is an insidious disease. The afflicted person's brain degenerates and wastes away over months and years before they die. A mutation of a gene on chromosome 4 causes Huntington's; a single codon CAG repeats excessively

(> 36 repeats vs. < 28 repeats). Notwithstanding considerable effort, researchers have yet to produce treatment options.

Despite the success of the HGP, the research community has realized that genomic data alone does not provide sufficient insight into the mechanisms of diseases [8]. Fortunately, in the post-genome era, researchers have leveraged the HGP's core principles and approaches to develop numerous 'omics based disciplines studying molecular mechanisms beyond the central dogma. Researchers can now study what happens after protein transcription using glycomics, proteomics, and peptidomics, to name a few.

Of the aforementioned 'omics disciplines, proteomics is arguably the most well researched. Researchers are interested in protemics (and peptidomics for that matter) because the cellular protein composition (proteome) and their degraded parts, endogenous peptides (peptidome) changes in response to disease. Cellular environments create new proteins, modify existing proteins, and degrade proteins into endogenous peptides [9].

1.3 Challenges in Proteomics

Unfortunately, complete proteomic profiling, that is, identification and quantification of all proteins in a biological system, remains elusive. For example, for unicellular organisms, the proteomic identification coverage of the genome has been occasionally achieved beyond 50%, but coverage for higher organisms rarely exceeds 10%. Furthermore, only a fraction of all identified proteins can also be reliably quantified (see Figure 1.2).

1.3.1 Biological Challenges in Proteomics

Why is complete proteomic profiling so challenging? The challenge starts in the nature of protein biology.

1. A biological system's proteome is more diverse than its corresponding genome.

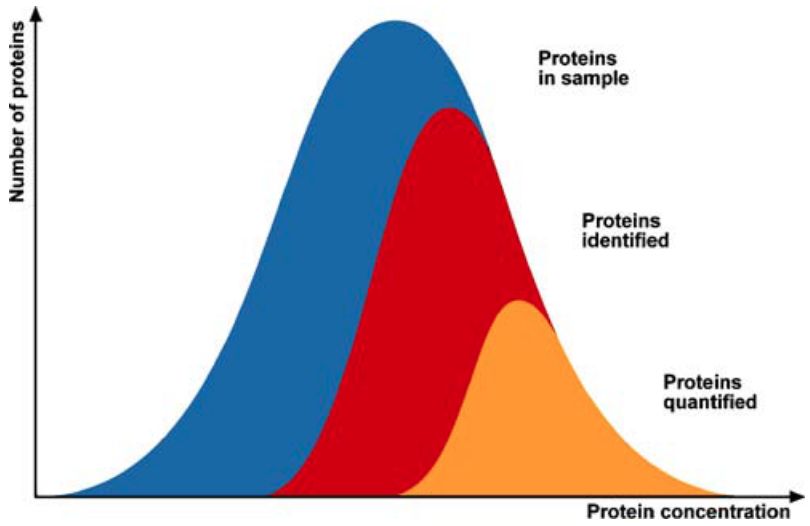


Figure 1.2: "Schematic representation of the fraction of a proteome that can by identified or quantified by mass-spectrometry-based proteomics." (Figure taken from Bantscheff et al. - 2008. [10])

The diversity arises because a gene can generate multiple proteins, either by polymorphisms in DNA sequences or by alternate splicing of genes [11].

2. A protein, once released by the ribosome into its cellular environment, is dynamic, responding to changes in its cellular environment. A protein can be become post translationally modified, interact with other proteins, form protein complexes, and degrade. Therefore, any proteomic analysis is simply a snapshot of the proteome at a point in time.

3. A proteome's dynamic range, its range of concentrations, exceeds that of any single analytical method or instrument used today. In fact, researchers have estimated that a proteome's dynamic range reaches ten orders of magnitude [12].

1.3.2 Technical and Analytical Challenges in Proteomics

Mass spectrometry-based profiling of complex biological samples has the potential to fundamentally change the way life science researchers interrogate biological systems [13]. For decades, hypothesis-driven studies were the cornerstone of biochemistry research. Classical wet bench methods and mass spectrometry (MS) studied single proteins in a top down approach [14]. However, in the past decade, there has been a clear trend in life science research towards a systemic study of biological entities [13]. Consequently, discovery-based studies have gained popularity. In discovery-based studies, researchers employ multi-faceted workflows combining high performance liquid chromatography coupled, electrospray ionization, and tandem mass spectrometry (HPLC-ESI-MS/MS) [15] to study a biological system in a bottom up (shotgun) approach. Here, biological systems can be sub-cellular, cellular, or extracellular. While progress has rapidly evolved over the past decade [14], analysis of resulting chromatographic data remains much more challenging than other high-throughput technologies, for example, microarray analysis in genomics [16, 17, 18]. Here, I list three analytical challenges that limit the potential of HPLC-ESI-MS/MS for revealing biological variation.

1. The current relative abundance paradigm is ill-suited to detect biological variation. This paradigm asks the question, 'Are the constituent peptides differentially abundant?" To answer this question, researchers select peptides with intensity fold changes greater than some threshold between two HPLC-ESI-MS/MS runs. Unfortunately, systematic differences, for example, sample amount, distort recorded peptide signals and is, therefore, especially problematic for detecting biological variation in complex biological samples.

2. HPLC-ESI-MS/MS analyses produce poorly repeatable and reproducible results, primarily due to extraneous variability. While current normalization methods work well to mitigate global extraneous variability (systematic bias) in HPLC-ESI-MS/MS measurements, they fail to mitigate localized extraneous variability

(complex variability in measurements) from transient stochastic events occurring during an HPLC-ESI-MS/MS run. Complex variability is particularly problematic for intensity-based label free relative quantification (iLFRQ) strategies aimed at revealing biological variation.

3. Finally, current software frameworks report protein level quantification rather than peptide level quantification. Furthermore, although many frameworks purport to be easily extensible, I found that extending them would have taken more time than creating a new framework to fit our team's needs.

While these challenges remain, researchers must conduct extensive hypothesis-driven experiments to weed out excessive false positives. Worse, real biological variation is missed because researchers do not see false negatives. As a result, researchers can draw incorrect conclusions about biological variation and thus miss key insights.

1.4 Contributions

In this dissertation, I offer three contributions addressing the three identified analytical challenges.

1. **The Proportionality Paradigm for iLFRQ:** In contrast to the relative abundance paradigm, asks the question,"Are the constituent peptides differentially proportional?" Researchers answer this question by computing a peptide's proportional abundance and then detecting their statistically significant differences across multiple samples.

2. **Proximity-based Intensity Normalization (PIN):** An embodiment of the proportionality paradigm that normalizes a peptide's signal intensity measured via HPLC-ESI-MS/MS by constructing its temporal neighborhood and then computing its relative proportion within that neighborhood.

3. **RIPPER: A new iLFRQ Software Framework:** Reports normalized peptide signal intensities rather than protein intensities. Optionally, researchers can match peptide signals to third party software peptide and protein identifications.

Although I continue to make enhancements to RIPPER, the Proportionality Paradigm for iLFRQ, PIN, and RIPPER are now under the auspices of the Office Technology Commercialization (OTC) at the University of Minnesota. RIPPER is available from the web site (https://z.umn.edu/ripper). The OTC has also applied for a patent: U.S. Patent Application Serial No.: 61/731,302, entitled "NON-PARAMETRIC METHODS FOR MASS SPECTROMIC QUANTIFICATION AND ANALYTE DIFFERENTIAL ABUNDANCE DETECTION"

Finally, I contend that **the proteomics community must shift to the proportionality paradigm for detecting biological variation** via HPLC-ESI-MS/MS workflows. What constitutes sufficient evidence for warranting a paradigm shift? Thomas Kuhn, one of the most influential philosophers of science in the twentieth century, outlined six criteria for warranting a paradigm shift in his 1962 book, *The Structure of Scientific Revolutions*, [19].

1. "Accurate - empirically adequate with experimentation and observation."
2. "Consistent - internally consistent, but also externally consistent with other theories."
3. "Simple - the simplest explanation, principally similar to Occam's Razor."
4. "Broad Scope - a theory's consequences should extend beyond that which it was initially designed to explain."
5. "Fruitful - a theory should disclose new phenomena or new relationships among phenomena."

In the following chapters, I provide arguments and evidence to fulfill each of these criteria. In Chapter 8, I provide justification that the criteria are met, warranting a shift in paradigms.

In summary, I expect my contributions described herein, the Proportionality Paradigm for iLFRQ, embodied in PIN, and implemented in RIPPER, to change the way researchers analyze HPLC-ESI-MS/MS experimental data. I expect with my contributions' widespread adoption, mass spectrometry-based proteomics will be much closer to meeting its potential as a biomarker discovery vehicle. Armed with more accurate results, researchers will expend fewer recourses exhaustively validating results through hypothesis-driven experiments. This will shorten the biomarker discovery cycle, thus reducing the bench to bedside cycle time (see Figure 1.1) Ultimately, patients will suffer less, and lives will be saved.

1.5 Dissertation Organization

The remainder of this dissertation is organized as follows.

- Chapter 2: I provide background information on comparative proteomics workflows. This chapter is intended for audiences not familiar with comparative proteomics.
- Chapter 3: I summarize work related to this dissertation. I focus on four topics: repeatability/reproducibility, the relative abundance paradigm, normalization, and iLFRQ software frameworks.
- Chapter 4: I present my 1^{st} contribution - The Proportionality Paradigm for iLFRQ.
- Chapter 5: I present my 2^{nd} contribution - Proximity-based Intensity Normalization (PIN).
- Chapter 6: I present my 3^{rd} contribution - RIPPER - An iLFRQ Framework.
- Chapter 7: I evaluate my contributions and present results demonstrating PIN's dominance over common normalization methods in reducing extraneous variability.
- Chapter 8: I draw conclusions from the research presented in this dissertation and discuss possible future work directions. Finally, I discuss the impact of my contributions and defend my position that a shift from the relative abundance paradigm to the proportionality paradigm for discovering biological variation via HPLC-ESI-MS/MS is warranted.

Chapter 2

Background

The more I know, the more I know I don't know - Socrates

2.1 Introduction

The research presented in this dissertation is interdisciplinary. As such, I expect that readers of this dissertation will originate from vastly different fields. Furthermore, the hierarchy and terminology in discovery-based proteomics can be confusing. To make matters worse, many terms in discovery-based proteomics are not succinctly defined, used loosely, or are overloaded. To help alleviate confusion, I present a mindmap for discovery-based proteomics (see Figure 2.1).

Prior to the last decade, with few exceptions, protein science researchers were limited to using enzymatic and/or chemical methods, in conjunction with ultraviolet (UV) absorbance or fluorescent spectroscopy to interrogate the structure of single, highly purified proteins [11]. For example, using enzymatic methods, researchers sequenced peptides by stepwise chemical degradation using Edman degradation. Sequencing proceeds from the n-terminus to the c-terminus, and UV absorbance spectroscopy subsequently identified the amino acid derivatives of the released amino acids [11]. Sequencing in this manner is labor intensive and does not scale to the level needed to conduct large scale comparative studies for the discovery of candidate protein biomarkers.

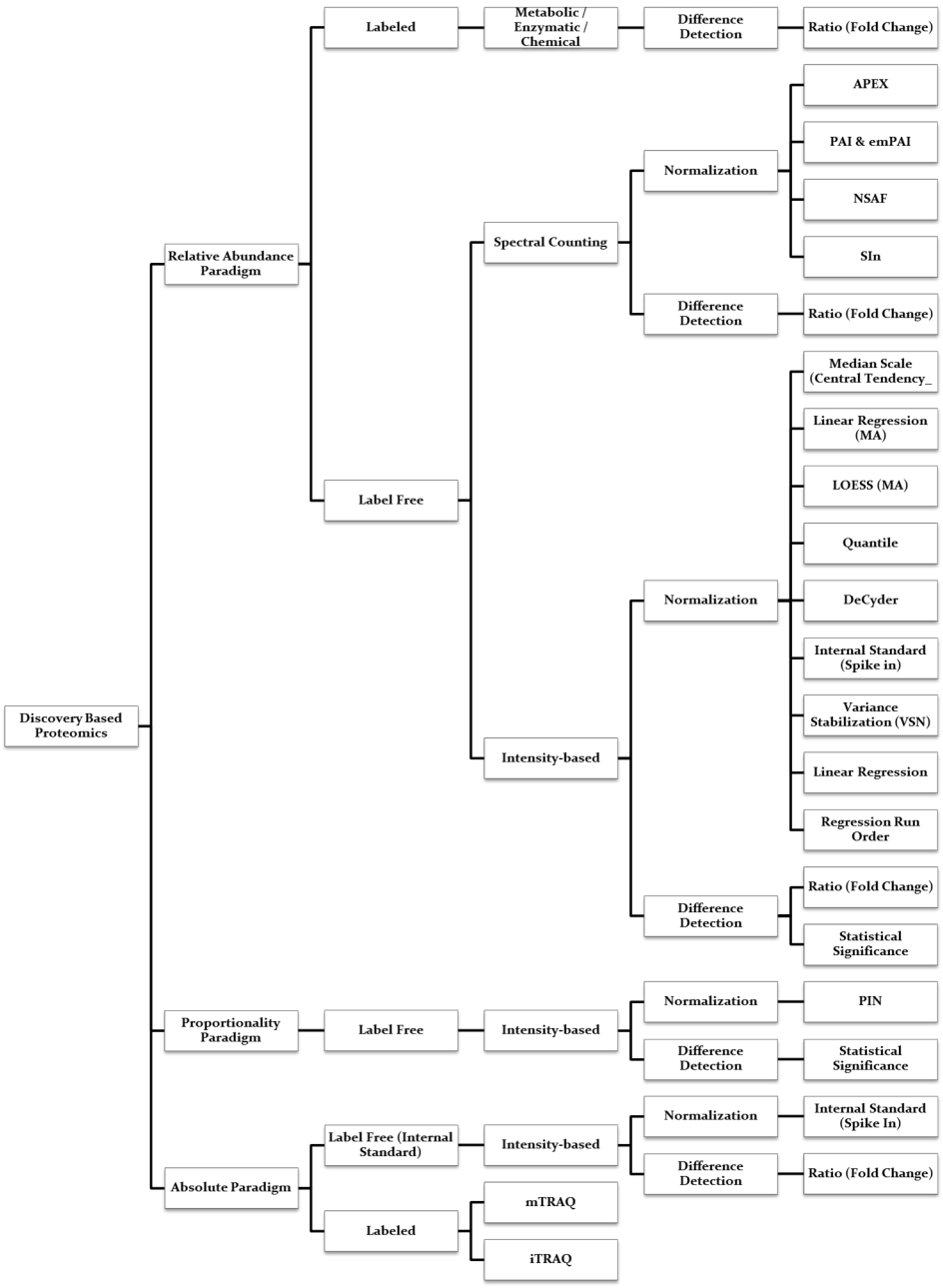


Figure 2.1: Discovery-based Proteomics - Mind Map. Starting from the left (Discovery-based Proteomics), the mindmap first depicts three paradigms for quantification, then progresses to the right until it lists the specific methods within each paradigm.

Fortunately, the genomics revolution ushered in a new age of high throughput biological systems analysis. Motivated by the success of genome sequencing, protein science researchers sought to develop new high throughput instruments and techniques targeting the study of proteins. As these new instruments and techniques developed, so did vocabulary in protein science. To make an analogy to the genome, Mark Wilkins and colleagues first coined the term *proteome* in the mid 1990's [20]. A proteome has since been defined by IUPAC as the "complete set of proteins encoded by the genome" [21]. Soon after, and the term proteomics was defined as " the study of all the protein forms expressed within an organism, as a function of time, age, state, external factors, etc." [22].

Today, MS, rather than chemical methods, is the workhorse of proteomics [15, 23]. MS possesses scalable high throughput capabilities, thus enabling large scale comparative proteomic studies. (Comparative proteomics, a sub-discipline of proteomics, is a specialized field that seeks to discover biological variation between two or more populations at the protein and/or peptide level.) [24]

The remainder of this chapter describes comparative proteomics workflows. It is intended to provide those unfamiliar with comparative proteomics a context for the remaining chapters. Section 2.2 describes comparative proteomics workflows in the context of a candidate biomarker discovery experiment using HPLC-ESI-MS/MS analyses of complex biological samples. The section contains several subsections, including sample collection; sample preparation; mass spectrometry and tandem mass spectrometry; and data analysis.

2.2 Comparative Proteomics Workflows

Despite biological challenges, researchers have developed complex, multi-faceted workflows that make great strides towards complete proteome profiling. While they differ in their implementations, that is, which protocols, technologies, and analytical tools they

employ, typical workflows have the same fundamental organization. As depicted in Figure 2.2, a typical comparative proteomic workflow starts with sample collection from two or more populations, followed by sample preparation, tandem mass spectrometry, and data analysis. The result is a list of candidate proteomic biomarkers that can be then used for further targeted (hypothesis-driven) validation experiments.

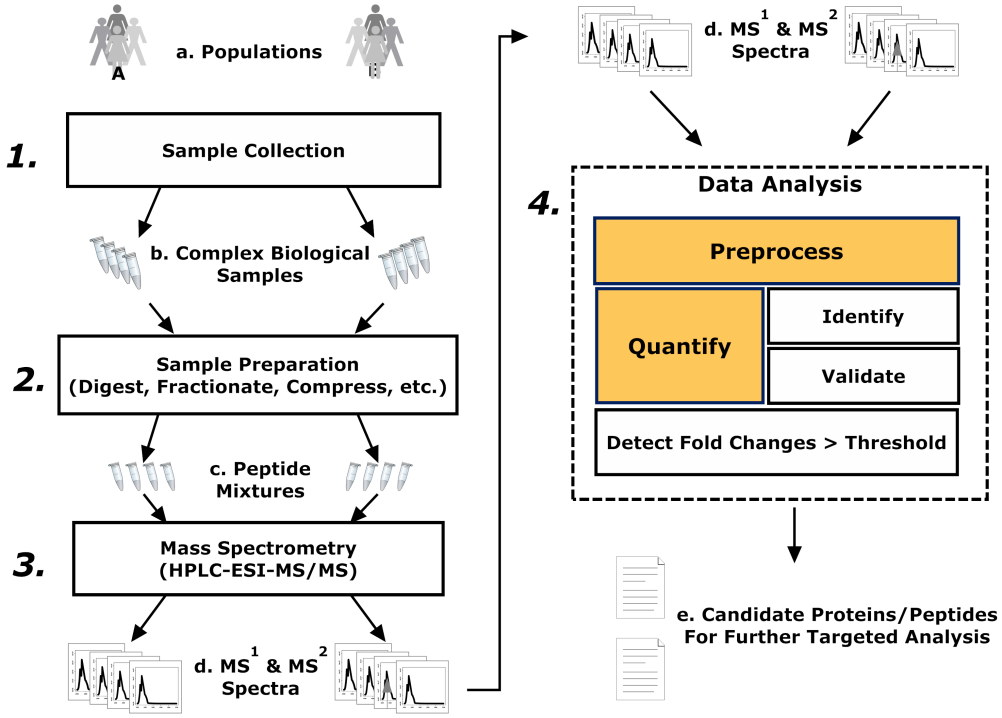


Figure 2.2: Typical Comparative Proteomics Workflow - The four square boxes correspond to the four distinct processing stages with inputs and outputs sandwiched between them. The square box with the dashed outline and titled *Data Analysis* contains a collection of computational steps, ranging from preprocessing to statistical analysis. My work falls within the two shaded boxes, preprocessing and quantification, within data analysis.

Figure 2.2 also serves as the organizing principle for the remainder of this section. The following subsections review each element in a typical workflow, with an in depth treatment of preprocessing and quantification within data analysis (shaded boxes in Figure 2.2).

2.2.1 Sample Collection

Sample collection starts by selecting appropriate populations meeting the experimental design requirements [25]. Here, an experimental design is a "the protocol that defines the populations of interest, selects the individuals for the study from the populations and/or allocates them to treatment groups, and arranges the experimental material in space and time." [26] Furthermore, a population comprises biological and/or technical replicates having a biological state in common, for example, healthy or diseased. As shown in Figure 2.3, a population and its biological and technical replicates form a nested hierarchical structure.

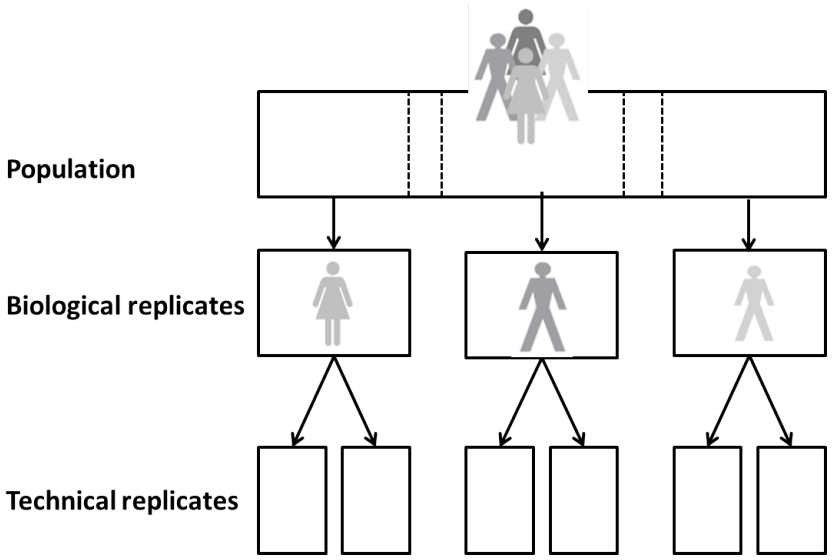


Figure 2.3: A simple schematic portraying the hierarchical nature of a population, its biological and technical replicates. (Adapted from Cairns - 2011 [25])

Researchers collect biological samples according to prescribed protocols. Unfortunately, sample collection protocols are diverse; each sample type and experiment type combination has its standard protocol for collection. Furthermore, research labs often tailor the standard protocols to fit their own needs. Thus, it is difficult to collectively, but succinctly, describe sample collection here. However, the specific protocols followed

for sample collection used in this dissertation are documented in detail, for example, de Jong et al. (salivary endogenous peptides) and Paulovich et al. (standard protein mixture spiked into yeast background).

2.2.2 Sample Preparation

Once researchers collect the biological samples, the samples must be prepared for MS/MS analysis. Just as sample collection protocols are diverse, so too are sample preparation protocols. Therefore, instead of describing individual protocols, here, I describe two sample preparation strategies pertinent to the work described in this dissertation, digestion and separation.

2.2.2.1 Digestion

Unfortunately, intact proteins are too large to be analyzed via MS/MS. To overcome this size limitation, researchers commonly employ a shotgun approach, similar to that used in genomic sequencing. In a shotgun approach, researchers enzymatically digest a protein sample, converting proteins into peptides using proteases, a process termed proteolysis.

While the choice of enzymes for protein digestion is vast [29], today almost all large scale MS-based proteomic studies use the protease trypsin for protein sample proteolysis [30]. Here, I highlight three reasons why trypsin is so popular. First, trypsin is stable under a wide variety of conditions. This allows trypsin to be used in a broad array of experiments. Second, trypsin is aggressive, meaning that it digests more sample compared to less aggressive proteases. Third, and most importantly, trypsin has high cleavage specificity. It cleaves at the c-terminal side at arginine or lysine residues. This leads to peptides in the preferred mass range for effective fragmentation and analysis by MS/MS [30], approximately 8-20 amino acids in length.

2.2.2.2 Separation

Digested protein mixtures from biological samples are naturally complex, which interferes with MS/MS accurately and completely detecting these peptides [31]. Each MS/MS run has a maximum computable peak capacity, that is, a maximum number of resolvable components [32]. Furthermore, within local regions, chromatographic peaks of complex mixtures show little hint of regular spacing [33]. Thus without sufficient resolving power, chromatographic peaks become convoluted, that is, a single peak may result from the detection of two different amino acid sequences simultaneously.

Two Dimensional polyacrylamide gel electrophoresis (2D-PAGE)

To increase resolving power, researchers separate mixtures prior to analysis via MS/MS. In the classical approach, researchers employ two dimensional polyacrylamide gel electrophoresis (2D-PAGE), which combines two orthogonal separation dimensions to improve resolution of complex protein mixtures. In the first dimension, proteins are separated by their isoelectric point, using a technique known as isoelectric focusing. In the second dimension, proteins are separated by the electrophoretic mobility using SDS-PAGE. After staining reveals protein clusters as dark spots in the underlying gel, researchers excise clusters, enzymatically digest the cluster, and analyze the resulting peptide mixture via MS/MS. Unfortunately, 2D-PAGE can only differentiate about 1500 proteins, far less than typically present in a protein mixture [34]. Furthermore, protein mixture dynamic range and solubility issues limit the detection of low-abundance and hydrophobic proteins via 2D-PAGE [35].

High Performance Liquid Chromatography (HPLC)

Today, digested protein mixtures from biological samples are commonly separated by high performance liquid chromatography (HPLC) [36, 11, 37]. Hunt and colleagues first pioneered HPLC coupled to MS/MS for separation of peptide mixtures [38]. HPLC comprises three components.

- Adsorbent: often a mixture of solvents, for example, water and acetonitrile
- Sorbent: typically a granular material, for example, silica or polymers, packed into a capillary column amenable for peptide mixture interaction (ion exchange (IEX) [39], reverse phase (RP) [40], hyrdophillic interaction [41], and affinity [42]) [36].
- Programmable pump: varies adsorbent component concentration while forcing the adsorbent and peptide mixture through the sorbent packed column

For example, in RP-HPLC, a typical gradient profile might start at 5% acetonitrile (in water or aqueous buffer) and progress linearly to 95% acetonitrile over 60 minutes [43, 44, 45].

Preliminary Separation

Unfortunately, HPLC remains a single dimensional separation, which provides enough peak capacity to resolve simple peptide mixtures, but not complex peptide mixtures from biological samples [46]. Researchers address the limited peak capacity problem by combining complementary separation techniques to create two-dimensional separation strategies [31]. While a plethora of separation combinations are in use today, two are notable: 1) multi-dimensional protein identification technology (MudPIT) which uses strong cation exchange (SCX), coupled to Reverse Phase HPLC (RP-HPLC) [29] [47] and 2) offline immobilized pH gradient gels (IPG) using isoelectric focusing (IEF) [48, 49, 50] or free flow electrophoresis (FFE), coupled with RP [51, 52]. Of the two, it has been shown that IEF outperforms SCX in cataloguing proteins in complex biological samples [53].

Three Dimensional Separation and Dynamic Range Compression

More recently, researchers have added a third dimension to peptide separation. For example, three-dimensional separation combines 1) preparative IEF, 2) SCX HPLC and 3) RP HPLC [54, 55]. As described by Bandhakavi et al., the new strategy adds dyanamic

range compression (DRC) to the three dimensional strategy. Here, they employ combinatorial chemistry derived hexapeptide bead treatments, thus enabling increased mass spectrometric detection of lower abundance proteins. In this DRC scheme, millions of hexapeptide sequences act as affinity "baits", with each hexapeptide putatively having high binding affinity for one/few related proteins within the complex protein mixture. Most proteins present at lower-abundance levels are effectively fully bound by their hexapeptide baits. Simultaneously, very high abundance proteins quickly saturate their hexapeptide baits, such that a significant proportion of these proteins do not bind and are carried away in flow-through. Upon elution of bound proteins from the hexapeptide bead library, the resultant complex mixture is compressed for dynamic range owing to the "partial depletion" of higher abundance proteins and "enrichment" of lower-abundance proteins".

2.2.3 Mass Spectrometry and Tandem Mass Spectrometry

2.2.3.1 Mass Spectrometry (MS)

Over the past decade, MS instrumentation has been the core technology driving proteomic advances. As shown in Figure 2.4, mass spectrometry takes as input a prepared sample, typically a peptide mixture, analyzes the peptides using a mass spectrometer, and outputs spectral data. Mass spectrometers analyze peptides by measuring the mass to charge ratio (m/z) of ions and counts the number of ions present.

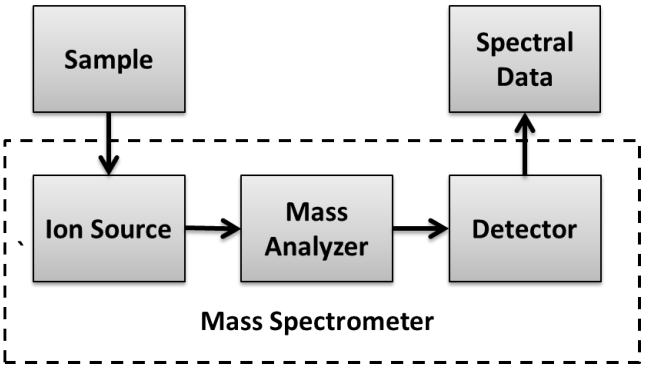


Figure 2.4: A generalized schematic of mass spectrometry. The dashed box depicts a mass spectrometer. Each mass spectometer has an ion source, a mass analyzer, and a detector. A mass spectrometer takes in a prepared sample and outputs spectral data.

While mass spectrometry instrumentation has advanced in the last decade, a figure from Aebersold et al. in 2003 provides an introductory overview of mass spectrometry instrumentation (see Figure 2.5) [46]. This figure includes ionization sources, instruments with a single mass analyzer for mass spectrometry, and instruments with two or more mass analyzers for tandem mass spectrometry. The following three sections describe a mass spectrometer's components, an ionization source, a mass analyzer, and a detector.

Ionization Source

An ionization source causes a peptide to lose one or more electrons, making it a positively charged ion. The two most important ionization techniques for proteomics are Matrix Assisted Laser Desorption/Ionization (MALDI) shown in the upper right panel of Figure 2.5 [56, 57, 58, 59, 60] and ESI shown in the upper left panel of Figure 2.5 [61]. The advent of MALDI and ESI revolutionized analysis of peptides via MS because they are considered soft ionization techniques, that is, they ionize peptides without smashing them to bits. In fact, they each won their respective inventors the Nobel Prize in Chemistry in 2002.

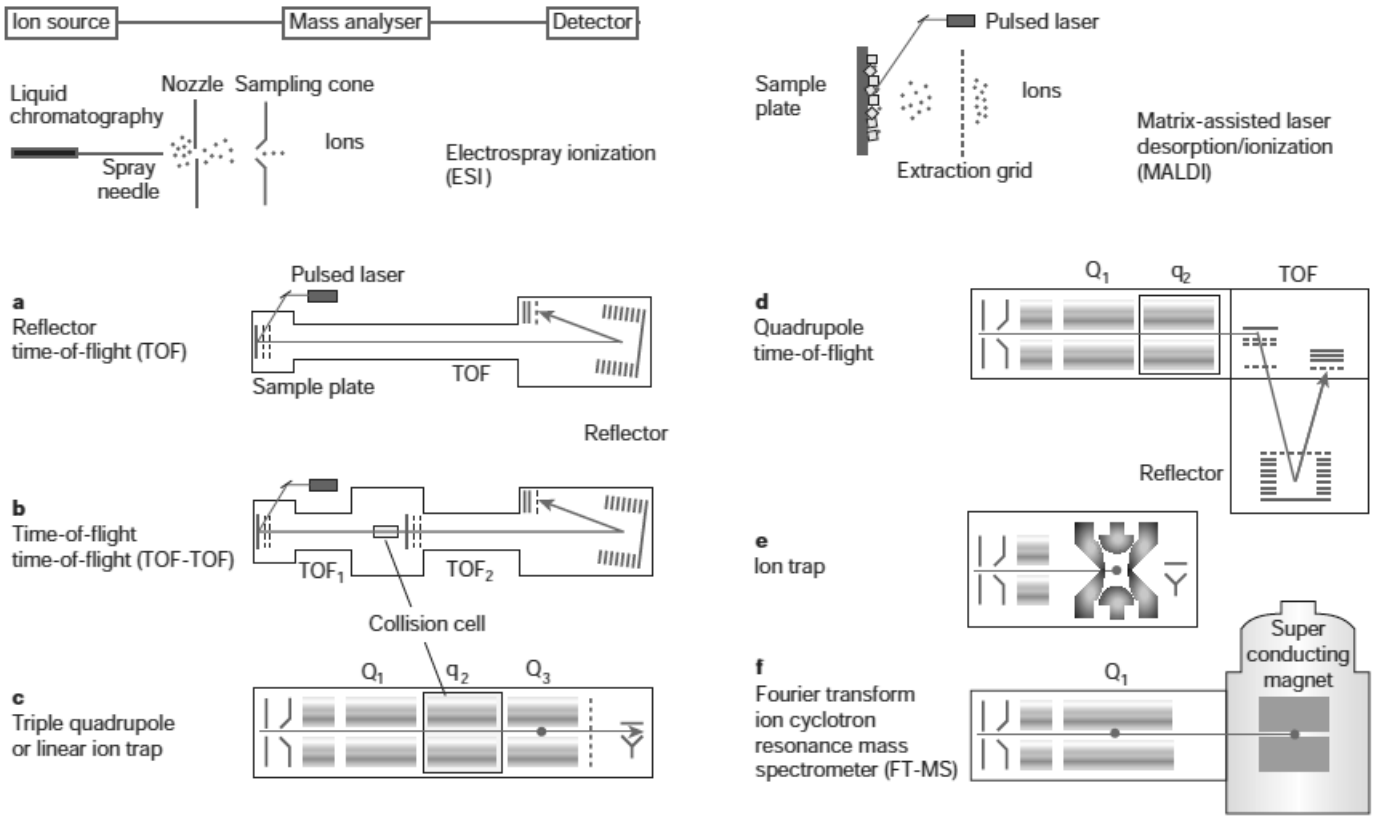


Figure 2.5: Mass spectrometers used in proteomics research. The upper left and right panels depict two common ionization sources, electrospray ionization (ESI) and matrix assisted laser desorption/ionization (MALDI). (a-f) Different instrumental configurations shown with their typical ionization sources. a) time of flight (TOF) b) TOF-TOF c) quadrupole d) quadrupole-TOF e) ion trap f) FT-MS. (Adapted from Aebersold et al. - 2003 [46])

Mass spectrometers equipped with MALDI type ionization sources, play an important role in proteomics, particularly for whole protein analysis [62]. On the other hand, mass spectrometers equipped with ESI ionization sources are particularly well suited for aqueous mixtures [63], for example, complex biological mixtures such as serum, blood, urine, or saliva. ESI requires two steps: dispersal of highly charged droplets and their subsequent evaporation [64]. When using HPLC, the capillary carries a small flow of liquid peptide mixture. As the peptides elute, a high electric field is applied at the tip of the capillary. As a result, the peptides become ionize and then vaporize. Unfortunately, HPLC-ESI flow rates at atmospheric pressure are not sufficient for complex biological samples. Furthermore, when researchers increase flow rates through higher pressure or larger internal diameters, larger droplets form, which leads to electrical breakdown. Fortunately, adding a nebulizing gas such as N^2 stabilizes the mixture [64]. Together, the electrical field and nebulizing gas produce gaseous phase ions which then escape through an orifice to the mass analyzer. The ions exiting the ion source tend to have 2, 3, or 4 electrons lost, that is, ions generally have a 2^+, 3^+, or 4^+ charge state. The ions m/z values can then be expressed as follows:

$$m/z = (MW + nH+)/n \tag{2.1}$$

where m/z = the mass to charge ratio - a unit-less measure, MW = the molecular weight of the peptide, n = the integer number of charges on the ions, H = the mass of a proton = 1.008 Da.

Mass Analyzer

The mass analyzer is the component in a spectrometer that separates ionized peptides based on the m/z value and outputs them to the detector. While there are numerous types of mass analyzers, five of which are commonly used in proteomics research [46]. These are the ion trap, time-of-flight (TOF), quadrupole, Fourier transform ion cyclotron (FT-MS) analyzers, and Orbitraps.

- Time of Flight (TOF): In TOF mass analyzers, ions are accelerated by an electrical field through a tube towards a detector, and the time that it takes to travel to the detector is used to estimate the ion's m/z value [65].

- Quadrupole/Octopole: In quadrupoles and octopoles, ions travel through 4 or 8 parallel rods where direct current (DC) voltage and radio frequency (RF) are applied between pairs of rods. Varying the ratio of voltage causes similar m/z ion trajectories to stabilize and eject from the quadrupole (see Figure 2.6a 2.6aa).

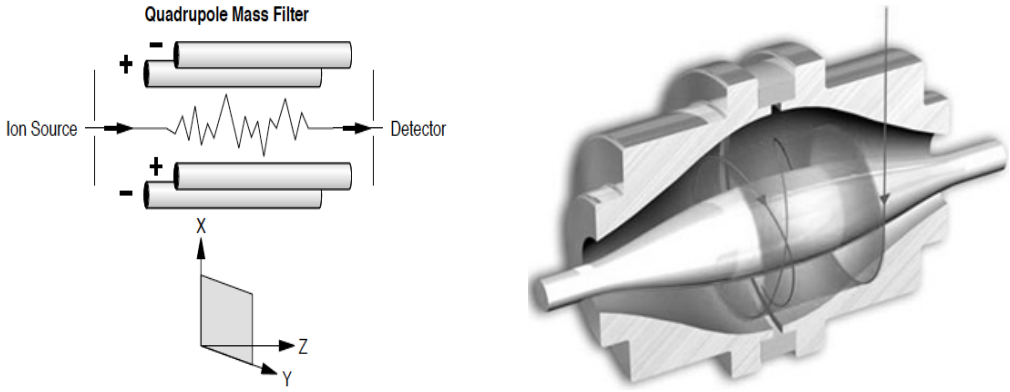
- Ion trap: An ion trap operates with similar principles as quadrupole. However, after entry, the electric field in the cavity due to the electrodes causes the ions of certain m/z values to orbit in the space. As the radio frequency voltage increases, heavier-mass ion orbits become more stabilized, and light-mass ions become less stabilized. This causes them to collide with the wall, eliminating the possibility of traveling to and being detected by the detector (see Figure 2.5e). Varying the electric field along over time allows the peptides to break centrifugal force, which propels them towards the detector.

- Fourier Transform Ion Cyclotron Resonance (FT-ICR): The FT-ICR also traps ions, but does so with extremely strong magnetic fields. Figure 2.5f shows the FT-ICR in combination with a linear ion trap. (MS instruments employing FT-ICR are also referred to as FT-MS).

- Orbitrap: In an Orbitrap mass analyzer, the ions combine rotation around an electrode system with harmonic oscillations along the axis of rotation at a frequency characteristic of their m/z [66, 67] (see Figure 2.6b).

Detector

After the mass analyzer separates the ions, they reach the ion detector. The number of ions ejected from the mass analyzer at a particular moment in time is typically very small,



(a) Quadrupole (b) Orbitrap

Figure 2.6: LTQ-Orbitrap Schematic. a) Depiction of a quadrupole mass analyzer (figure from www.currentseparations.com) b) Depiction of an Orbitrap mass analyzer (figure from Thermo Scientific) [68]

therefore, amplification is required to detect a signal. Modern commercial instruments commonly use microchannel plate detectors [69]. For example, in FT-MS and Orbitrap instruments, the detector consists of a pair of metal surfaces within the mass analyzer or ion trap area which the ions only pass near as they oscillate. A weak alternating current image is produced in a circuit between the electrodes and then recorded. As shown in Figure 2.7, the signal recording is subsequently converted to a mass spectrum by sophisticated software using a fast Fourier transform (FFT).

2.2.3.2 MS^1 Spectra

Researchers can visualize spectral data many ways. First, the total intensity from all m/z values detected and recorded in a single run can be viewed as a chromatogram (sometimes referred to as a total ion chromatogram). Figure 2.8 shows a chromatogram derived from a single HPLC-ESI-MS/MS run analyzing complex biological sample. Here, the y-axis is ion abundance (intensity) and the x-axis is time. Second, as shown in Figure 2.9, individual MS^1 scans can be visualized in two dimensions, with the m/z values of ionized peptides along the x-axis ion abundance (intensity) along the y-axis. Within each

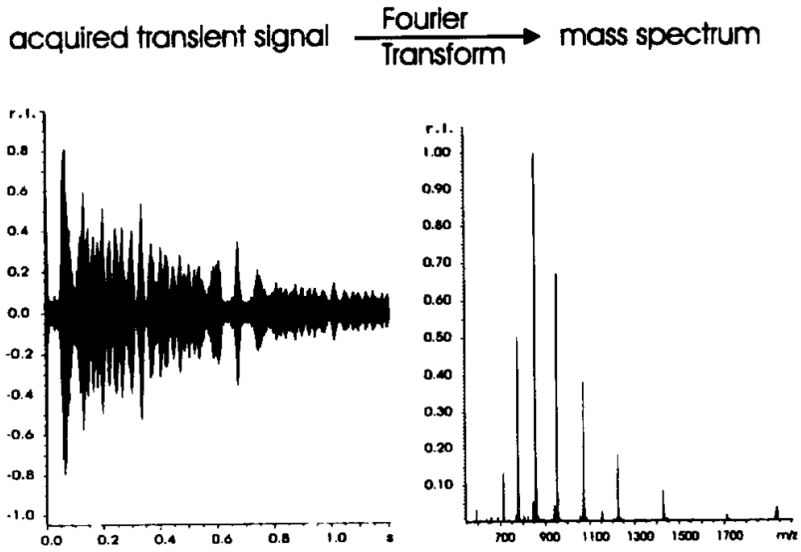


Figure 2.7: Fast Fourier Transform (FFT) converts raw spectral data into spectra (figure from Amster - 1996) [70].

scan, peptides' isotopes form isotopic distributions (see Appendix B.1 for a description of isotopes). The spacing between individual peaks in the distribution is dependent on the charge acquired by the detected peptide during ionization. Figure 2.10 is a portion of the spectrum in Figure 2.9 showing a typical isotopic distribution of an ionized peptide with a 2^+ charge (as evidenced by .5 m/z units between each detected isotope). Third, as shown in Figure 2.11, a small m/z range window can be viewed along the retention time axis in an extracted ion chromatogram (XIC). While an XIC is often depicted as a two-dimensional, with retention time along the x-axis and intensity along the y-axis, here I show an XIC in three dimensions, adding m/z along the z-axis.

2.2.3.3 Tandem Mass Spectrometry (MS/MS)

Tandem mass spectrometry-based proteomics experiments rely on the same principle as Edman degradation, a long standing chemical technique for peptide sequencing [71]. In Edman degradation stepwise degradation from the peptide's n-terminus followed by

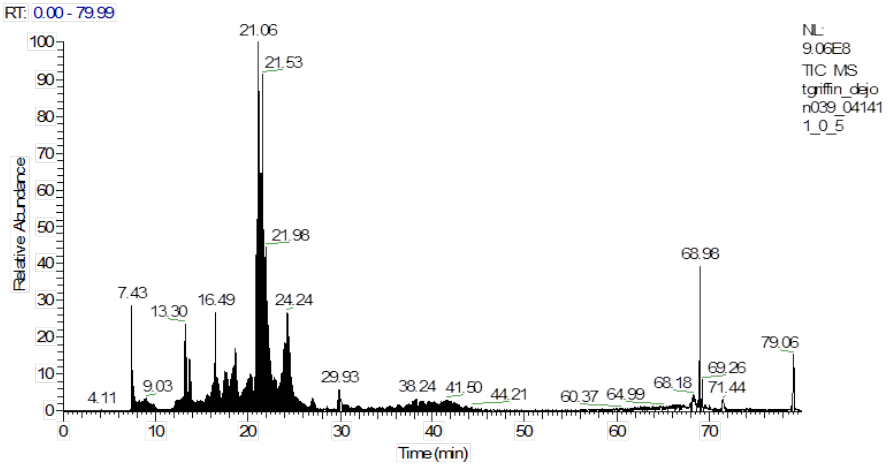


Figure 2.8: Example of a chromatogram for an HPLC-ESI-MS/MS run using an LTQ-XL Orbitrap mass spectrometer. The chromatogram was generated using the software Xcalibur from ThermoScientific.

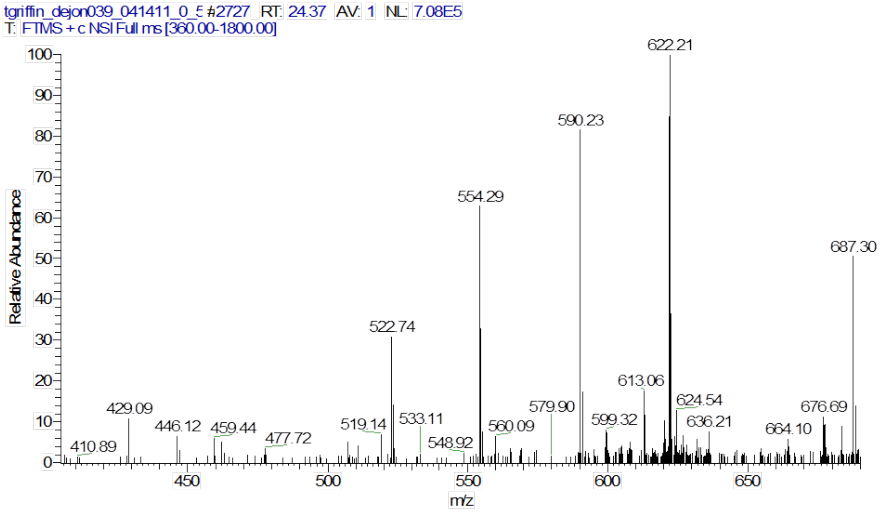


Figure 2.9: A MS^1 scan can be viewed in two dimensions, with m/z values along the x-axis and ion abundance (intensity) along the y-axis. The chromatogram was generated using the software Xcalibur from ThermoScientific.

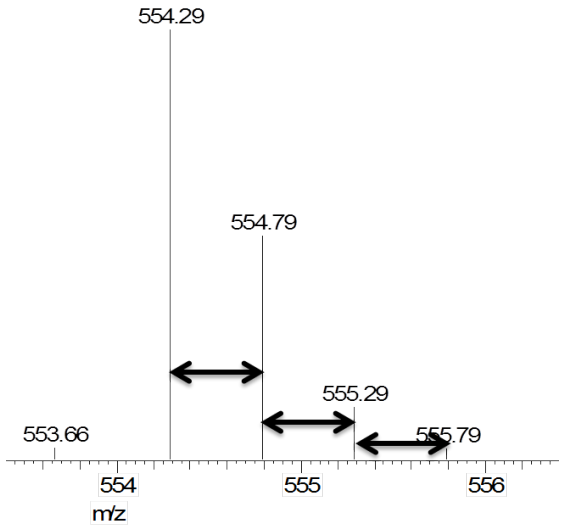


Figure 2.10: A typical isotopic distribution of an ionized peptide. Peptide with monoisotopic peak at m/z value 554.29 has a 2^+ charge (as evidenced by .5 m/z units between each detected isotope).

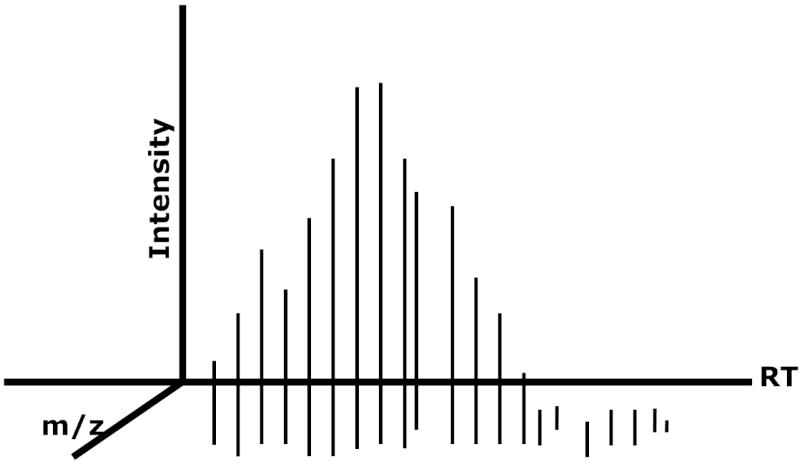


Figure 2.11: Cartoon of an extracted ion chromatogram (XIC) in three dimensions, with retention along the the x-axis, intensity along the y-axis, and m/z along the z-axis.

chromatographic analysis of the released derivatives determines the amino acid sequence.

MS/MS was introduced in 1980 by McLafferty et al. [72] and is still essentially the same today. The process of MS/MS is perhaps best described by in Hunt et al. in 1986 using a triple quadrupole mass spectrometer [73].

> *"...the first quadrupole of the instrument is used to select a single (M+H)+ ion from the mixture and to transmit it to quadrupole 2, a collision chamber, where the peptide undergoes collisions with argon atoms and suffers fragmentation primarily at the various amide linkages in the molecule. The resulting fragment ions are then transferred to the third quadrupole, which separates them according to mass. The end result is a mass spectrum containing ions characteristic of the sequence of amino acids in the selected peptide."*

Today, the scanning and ion selection that occurs in the first quadrupole is referred to as MS^1 and the fragmentation in the second quadrupole combined with the scanning in the third quadrupole is referred to as MS^2. The fragmentation that occurs during MS^2 mimics Edman degradation because MS^2 fragmentation (dissociation) randomly breaks along the backbones between amino acid residues. This results in two, rarely more, fragment ions, one each containing the n-terminus and the c-terminus. The m/z values of fragment ions are recorded in the MS^2 spectra for every selected precursor peptide ion.

To accomplish MS/MS researchers combine two or more mass analyzers in sequence within a single mass spectrometer. Tandem mass spectrometers are produced with varying combinations of mass analyzers. Classical combinations include TOF-TOF (Figure 2.5b), triple quadrupole/linear ion trap (Figure 2.5c), and quadrupole TOF (Figure 2.5d). Newer instruments, such as the LTQ-Orbitrap Velos shown in Figure 2.12, combine quadrupoles, octopoles, and orbitraps with multiple fragmentation cells such as those described by Hunt, collision induced dissociation (CID), and recently introduced higher-energy collisional dissociation (HCD).

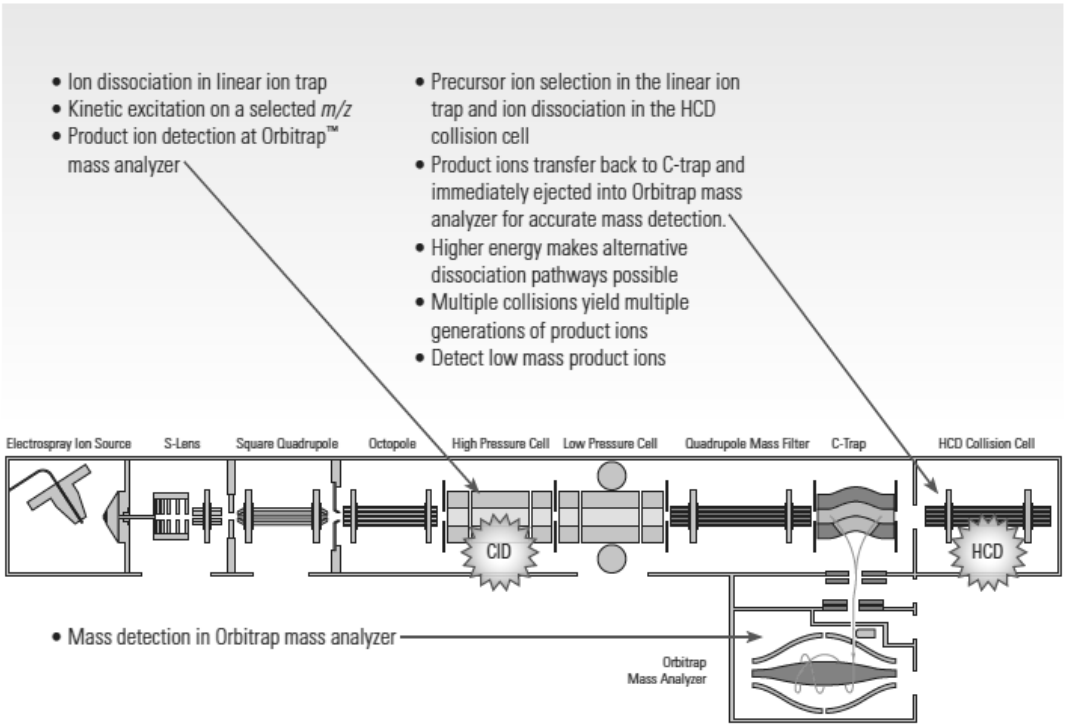


Figure 2.12: LTQ-Orbitrap Velos Schematic (figure from Thermo Scientific)

2.2.3.4 MS^2 Spectra

HPLC-ESI-MS/MS MS^2 stage fragmentation breaks a peptide's backbone. Figure 2.13 depicts the three possible breaking points along the backbone. The standard nomenclature for these fragment ions identifies both the point of fractionation as well as which terminus retains the charge. Ions a, b, and c are n-terminus fragments and x, y, and z are c-terminus ions. Although the exact point of fragmentation depends on many factors, the primary factor is the type of dissociation applied. CID and HCD produce primarily b and y ions, with a few a ions sprinkled in while elctron transfer dissociation (ETD) produces primarily c and z ions. Their resulting fragmentation patterns differ enough to impact the programs interpreting mass spectra.

The m/z values of fragment ions are recorded in the MS^2 spectra for every selected precursor peptide ion. However, individual fragmentation peaks are not valuable; as in

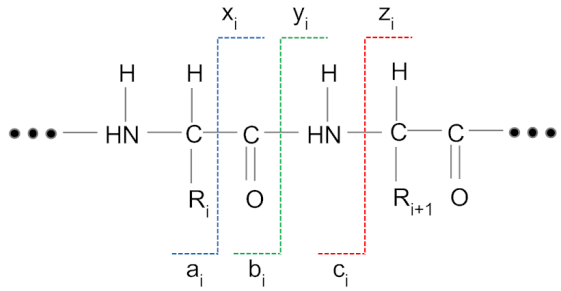


Figure 2.13: Cartoon of a generic amino acid sequence within a peptide. The backbone is the horizontal line in the center of the figure. The amino acid residues are located below the backbone and are denoted as R_i and R_{i+1}. The dashed lines represent possible fragmentation points. The standard nomenclature for these fragmentation types is x_i, y_i, z_i, a_i, b_i, and c_i.

Edman degradation, it is their m/z differences that are informative. As shown in Figure 2.14, the m/z differences between these peaks determine both the amino acid residue identities and their positions, thus identifying a peptide.

2.2.4 Data Analysis

Analyzing spectral data from HPLC-ESI-MS/MS runs is a multistep process. As shown in Figure 2.2 data analysis comprises five steps. The first two sub-steps, preprocessing and quantification, will be only briefly described here.

2.2.4.1 Preprocessing

Preprocessing of data comprises many sub-steps. While workflows may differ in grouping, order, and implementation of preprocessing sub-steps.

Baseline Correction

Baseline correction, smoothing, and calibration of data is needed to clean noisy spectra. Baseline correction subtracts background and chemical noise from spectra. Commonly, researchers compute an intensity threshold, such as median intensity or signal to noise ratio and subtract the threshold for each recorded peak [74].

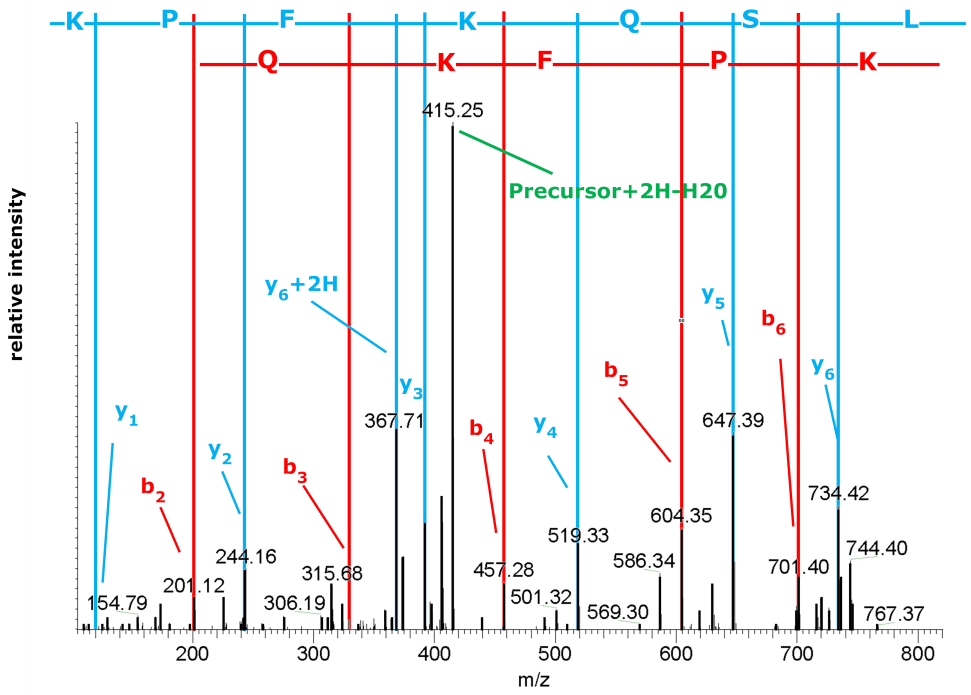


Figure 2.14: A representative HPLC-ESI-MS/MS spectrum with its b and y ions identified. Along the top, the horizontal line contains the amino acids identified for y ion spacing starting from the n-terminus. Just below this line is a second horizontal line which contains the amino acids identified for b ion spacing starting from the n-terminus.

Peptide Signal Extraction

Peptide signal extraction is a multi-step process. First, isotopic peak envelopes (clusters) must be extracted from MS^1 scans (see Figure 2.10). Commonly, algorithms first detect isotopic peak clusters using employ by wavelet techniques [75] [76] [77] Then they perform isotope pattern matching in the m/z dimension to find an isotopic envelope that differentiates a possible peptide signal from noise [74]. If the identity of the peptide is unknown, the averagine [78] model or a pseudo-Poisson distribution can be used to model the isotopic distribution. Different software solutions employ different scoring models [74] to evaluate the goodness-of-fit of the model. For example, the software framework msInspect [75] employs Kullback-Leibler divergence score [79].

After detecting and extracting isotopic peak clusters, peptide signal extraction reconstructs XICs by combining isotopic peak clusters in the retention time dimension. The choice of models, from a simple m/z tolerance to complex elution time models will affect the output. As a rule-of-thumb, more modeling results in fewer, but higher quality, peptide signals [74].

Retention Time Alignment

Retention time alignment corrects for non-linear drifts in elution time profiles between HPLC-ESI-MS/MS runs. Researchers commonly employ warping algorithms to align retention times. Time warping for alignment is the process of optimally matching two temporally different series. Dynamic time warping (DTW) was originally developed for speech recognition through synchronization [80], but later was also applied to chromatographic data [81]. It uses dynamic programming with very local stretching and compression (every individual time point may individually be stretched or compressed) to estimate the optimal warp. Parametric dynamic time warping (PDTW), on the other hand, was developed for chromatographic application [82, 83] and has been shown to outperform DTW on MS^1 chromatograms. PDTW, as the name implies, use a nonlinear

parametric function to align the spectra, operating globally on the entire set of spectra. Function parameters are typically estimated through a partial least squares (PLS) machine learning algorithm [84].

2.2.4.2 Quantification

Unfortunately, mass spectrometry is not inherently quantitative [46]. This is because proteolytic peptides show great variability in physiochemical properties that in turn result in variability in mass spectrometric response between runs. Additionally, mass spectrometers only sample a small percentage of the total peptides in a sample [85]. Nonetheless, researchers have devised various schemes to estimate protein and peptide quantities from spectral data. Protein and peptide quantification via HPLC-ESI-MS/MS can be categorized as absolute or relative [86]. Absolute quantification measures the number of individual proteins or peptides in a sample or mixture, commonly using either mole/volume or mass/volume units [13]. Relative quantification, on the other hand, compares the abundance of proteins or peptides in each of two or more samples [86]. In theory, relative quantification is a subset of absolute quantification. If I know the absolute amounts of proteins or peptides in two or more samples, I can compute the relative abundance [15].

Absolute Quantification

Internal standards are primarily used to determine absolute abundance, but have also been used to determine relative abundances between two or more samples in separate HPLC-ESI-MS/MS runs. The internal standard approach attempts to mitigate systematic extraneous variability. Researchers can employ one of two options as internal standards [87]. First, they can spike in a protein prior to enzymatic digestion of a sample, or alternatively, spike in a synthetic peptide into a peptide mixture. While the internal standard can be either a protein or peptide, for simplicity, hereafter, I use the peptide level description to include both. In this option, the spiked in peptide must not naturally

occur in the sample or mixture being analyzed. Second, researchers can select a housekeeping peptide as the internal standard. The housekeeping peptide should be present in constant relative concentrations across samples or mixture. After initial quantification, each protein intensity is normalized by the intensity of the spiked in peptide.

Relative Quantification

Relative quantification compares the intensity of corresponding proteins (or peptides) in two or more samples. While I describe relative quantification in detail in Chapter 3, I will also briefly describe it here. In general, researchers conduct relative quantification using relative abundance paradigm. The relative abundance paradigm first determines protein or peptide's abundance in one sample and then computes it abundance ratio (fold change) between samples [88]. Two approaches exist under the relative abundance paradigm, labeled and label free. Label free approaches separately analyze samples via MS or MS/MS and compare the spectra (either MS^1 intensity or MS^2 spectral count). Labeled strategies, on the other hand, metabolically, enzymatically, or chemically tag sample peptides with label variants, one variant for each sample [86]. Researchers then combine the samples into a single mixture and analyze it once via MS or MS/MS runs.

2.2.4.3 Identification

In discovery-based proteomics the proteins or peptides in analyzed samples are unknown. Protein identification via HPLC-ESI-MS/MS is peptide-centric, that is, researchers first identify a peptide's amino acid sequences and then infer proteins from those peptide identifications. The following sections describe the three approaches researchers take to identify peptides (database search, spectral library search, and de novo sequencing) and then describes protein inference.

Database Search

Database search compares the observed MS^2 spectrum to theoretical spectra constructed

from a silico digested FASTA database and then scores resulting peptide spectrum matches. Many scoring functions exist, including the number and intensities of matching peaks and intervals, spectral contrast angle, cross-correlation, rank-based scoring, and SEQUEST scoring [89]. Since SEQUEST is a popular database search engine, I briefly describe it here. SEQUEST has two phases of scoring: preliminary and final. The preliminary scoring phase first restricts the search space to the top 500 scoring theoretical spectra using 1) b and y ion continuity scores, that is, do b ion and y ion sequences mirror each other, and 2) presence of immonium ions, that is, diagnostic ions observed at the low end of an MS^2 spectrum indicating the presence of specific amino acid residues in a peptide sequence [90]. The final phase computes a number of scores for those 500 top scoring spectra [89], most importantly the cross correlation score, a common function for computing the correlation between two signal series [91]. These scores serve as input to validation algorithms, which I describe in Section 2.2.4.4.

Spectral Library Search

Spectral library search strategies are similar to database search strategies, except the observed MS^2 spectra are compared to collections of experimentally generated spectra rather than hypothetical spectra [92]. These strategies outperform database search strategies in terms of error rates, speed and sensitivity. Using spectral libraries reduces the time spent repeatedly identify the same identifiable peptides by database searching [93], but can only identify a peptide if it has been previously analyzed by tandem mass spectrometry and its sequence positively identified.

De Novo Sequencing

De novo sequencing for proteomics has a long and rich history. Researchers originally sequenced spectra manually, a process which does not scale well. In 1990, Bartels introduced the idea of representing a spectrum as a graph [94]. Although the proteomics community has termed them spectrum graphs, these graphs should not be confused with

spectral graphs formalized in graph theory. In spectral graph theory, the set of eigenvalues of a graph's adjacency matrix defines the graph's spectrum. It should also not be confused with a general graph of nodes and edges, where a node does not have a position and an edge can connect any two nodes. In this novel spectrum graph representation, the vertices represent the spectrum m/z values and edges line two vertices if their mass difference is equivalent to the mass of an amino acid. Figure 2.15 shows a theoretical spectrum graph of the spectrum in Figure 2.14. Formally, Bartels defined the problem as:

> *Given amino acid masses* $M = \{m_1, ..., m_{20}\}$, *spectrum* $S = \{s_1, ..., m_c\}$, *transform spectrum graph* $G(V, E)$, *such that* $V = \{v_1, ...m_c\}$ *and* $G = \{g_1, ..., g_c\}$, *such that* v *represents a singular integer* m/z *and two vertices* $v_n, v_q, q \neq n$ *are connected by a directed edge* e if $|v_q - v_n|$ m_i.

Researchers then score each edge according to one of many scoring schemes.

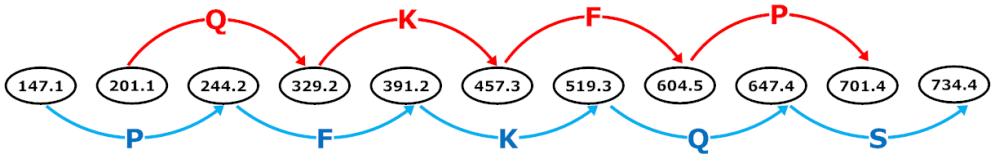


Figure 2.15: Spectrum graph representation - the vertices represent the spectrum m/z values and two vertices are linked by edges if their mass difference is equivalent to the mass of an amino acid.

The straightforward approach for determining the amino acid sequence from the spectrum graph is to traverse the graph from the start node, the n-terminus to the c-terminus [95]. The graph traversal can be done in a depth-first or breadth-first manner [89]. Of course, multiple paths inevitably exist in these spectrum graphs. Therefore, heuristic approaches select the highest scoring path as the amino acid sequence. Although Bartel's approach is now the de facto basis for most de novo peptide sequencing programs, several unresolved issues limited spectrum graph's, and, therefore, de novo

sequencing's, adoption. First, spectrum graph models were instrument specific which required training a new model for each new mass spectrometer. Second, the predominantly used CID has a propensity for incomplete fragmentation and results in multiple disconnected graphs, graph gaps, limiting the effectiveness of spectrum graph algorithms. Finally, lack of standardized scoring models for spectrum graphs hindered researchers' ability to compare experimental results. SHERENGA, introduced by Dancik et al. in a landmark publication 1999, addressed each of these limitations using a spectral graph-based algorithm [96]. Several research groups have since made additional enhancements to these algorithms, most notably dynamic programming [97] and probabilistic models using networks learned over annotated spectra [95, 98].

Protein Inference

Researchers use the peptide's amino acid sequence to search a FASTA database to infer the protein identity. Unfortunately, going from peptide to protein is not straight forward, a problem known as the protein inference problem. The protein inference problem refers to the difficulty in matching MS^2 spectra to peptide sequences, and then peptide sequences to proteins. The difficulty of inferring proteins from peptide identifications stems from the loss of the connectivity between peptides and proteins at the digestion stage. The loss of connectivity complicates computational analysis because same peptide sequence can be present in multiple different proteins [99]. When shared peptides occur, they can be assigned to a protein randomly, by some scoring scheme, or by Occam's razor loosely translated as, "when you have two competing theories that make exactly the same predictions, the simpler one is the better" [100].

2.2.4.4 Validation

Since MS-based proteomics results are inherently prone to inaccuracies, without careful filtering, its results are riddled with false positive identifications at both the peptide

identification and protein inference levels. To reduce the number of false positives, several scoring models have been proposed and developed to impart a confidence level on identified peptides and inferred proteins. Each of thousands of single-peptide identifications or protein inferences can be assigned an individual score. However, single case scores do not take into consideration the fact that multiple hypotheses are being tested. The p-value's close relative for multiple testing, E-value, or Bonferroni correction [101] is often used. However, the Bonferroni correction has been shown to be too conservative given the thousands of hypothesis tests in a single experiment [102].

Less conservative than the Bonferroni correction is the False Discovery Rate (FDR) controlling procedure, introduced by Benjamini and Hochberg [103]. They define FDR as the expected fraction of mistakes among the rejected hypothesis and suggested to control FDR in multiple testing. A well-established mechanism to implement FDR for database search results is to search against a decoy FASTA database of invalid peptide sequences, most often concatenated to the end of the target FASTA database with valid peptide sequences [104]. The premise for this approach is that a spectrum will match valid and random (invalid) sequences with equal probability, and target and decoy sequences do no overlap. All peptide spectrum matches above the threshold determined by the FDR controlling procedure are accepted as valid matches.

Because threshold statistical models tend to be instrument specific, researchers turned to machine learning, notably mixture modeling, to build a generic model that could process results from multiple instrument types. Mixture modeling uses models of two normal distributions, one for correct identifications and one for incorrect distributions, to determine a score threshold. Perhaps the most widely used example of mixture modeling for peptide identification validation is Keller et al.'s PeptideProphet [104]. It uses a discriminant score which is derived by converting several scores from the database search programs into a single score. To apply a two-component mixture model, PeptideProphet creates a histogram of discriminant scores and uses curve fitting to draw the correct and incorrect distributions. Using Bayesian statistics, it computes the probability of an

identification being correct given its discriminant score. Similar to PeptideProphet is ProteinProphet, which is used to validate protein inferences. It uses results from PeptideProphet as input to accurately compute the probability that an inferred protein is present in the sample [105] and derives a mixture model of correct and incorrect protein inferences, using an expectation-maximization routine (EM). As evidenced by their use in a number of prominent laboratories, PeptideProphet and ProteinProphet remain attractive options for validating peptide identifications and protein inferences. They are attractive because they are open source, freely available and integrated into freely available software, for example, the Trans-Proteomic Pipeline (TPP) [106], and Scaffold [107, 108].

2.2.4.5 Detect Statistically Significant Differences

Testing for statistically significant differences between two samples provides researchers an alternative to ratio (fold change) thresholds. While conducting statistical tests are generally more computationally expensive computing a simple fold change, statistical testing allows researchers to detect biological variation that might be otherwise missed. As stated by Bantscheff et al., when using the fold change approach "small but potentially significant changes go unnoticed (false negatives) and, in the absence of repeating the experiment, there is no way of assessing if the observed large protein changes that are backed by few spectral observations can be reproduced (false positive). Even small numbers of repetitions can increase confidence in the results considerably. In addition, the use of statistical testing models adds options to determine the probability of false decisions." Unfortunately, resarchers have been slow to adopt statistical analysis for detecing significant differences [13]. Nonetheless, I provide Tables 2.1 and 2.2 which list some common statistical tests that can be performed with and without replicates.

Test	Requirements	Stat. power	Application
t-test	Replications,n>=3 Data normally distributed	+++	All quantitation strategies
LPE-test	Replications,n>1	++	All quantification strategies 2-3 replicates Strong changes

Table 2.1: Tests for finding statistically significant differences between two or more replicated samples. (Adapted from Bantsheff et al. - 2007 [85])

Test	Requirements	Stat. power	Application
G-test	(Very) large # of spectraa	+	Spectrum counting
Fischer's exact test	(Very) large # of spectra	+	Spectrum counting
AC-test	(Very) large # of spectra	+	Spectrum counting

Table 2.2: Tests for finding statistically significant differences between two or more un-replicated samples. (Adapted from Bantsheff et al. - 2007 [85])

Chapter 3

Related Work

If I have seen further it is by standing on the shoulders of giants. - Isaac Newton

3.1 Introduction

This chapter describes work related to this disseration. Section 3.2 describes repeatability and reproducibility in the context of HPLC-ESI-MS/MS analyses of complex biological samples. Section 3.3.2 describes the relative abundance paradigm, the current standard for detecting biological variation via comparative protomics workflows. Section 3.5 describes common normalization methods and evaluates their ability to reduce variance. Section 3.6 describes common intensity-based label free relative quantifcaiton (iLFRQ) software frameworks.

3.2 Repeatability and Reproducibility

High repeatability and reproducibility are two tenants underpinning credible scientific experiments [109]. Repeatability represents variation observed for an analytical technique when a researcher (or operator) takes measurements multiple times using the same sample, system, and location [110, 111]. Reproducibility, on the other hand, represents variation from researcher (or operator), instrumentation, time, or location changes between multiple measurements [112, 113, 111].

3.2.1 Evaluating Repeatability and Reproducibility

Researchers evaluate repeatability and reproducibility by computing precision. According to the International Standards Organization (ISO) standard for reference laboratories: Error requirements, "precision is the closeness of agreement between repeated measurements." Furthermore, ISO 5725 states that "precision is usually expressed in terms of imprecision, and computed as either, standard deviation, variance, or coefficient of variation (CV) of the test results, with less precision reflecting in large standard deviations."

According to the IUPAC Gold Book, "A problem often arises when the combination of several series of measurements performed under similar conditions is desired to achieve an improved estimate of the imprecision of the process. If it can be assumed that all the series are of the same precision although their means may differ, the pooled standard deviations" can be calculated. Furthermore, "results from various series of measurements can be combined in the following way to give a pooled relative standard deviation" [112, 113], which is alternatively referred to as pooled estimate of variance (PEV). Appendix C.1.2 defines formulas for computing CVs and PEVs in HPLC-ESI-MS/MS workflows.

3.2.2 Repeatability/Reproducibility in HPLC-ESI-MS/MS Workflows

Unfortunately, the complex nature of HPLC-ESI-MS/MS analysis of biological samples gives rise to variation in the peptides and proteins identified [111, 114, 87] and quantified [115, 116, 15, 16]. For example, slight changes in sample preparation [27], environment, and liquid chromatography [111] can have dramatic effects on the recorded peptide signals' intensities. Workflows employing HPLC-ESI-MS/MS produce results with poor repeatability and reproducibility [111].

3.2.3 Variation/Variability

Some variance in HPLC-ESI-MS/MS chromatographic data comes from true biological variation. It also comes from extraneous variability comprised of systematic bias (sample variability and instrument variability), and complex variability. Sample variability stems from inconsistent sample preparation, including incomplete enzymatic digestion, pipetting errors, and so on. It tends to be global. By global, I mean errors similarly affect each peptide in a sample, or in the cases of pipetting errors, each peptide in an aliquot. Global variability results in systematic bias [117, 118, 27]. Instrument variability stems from physical changes in the mass spectrometry hardware or environment. These changes include HPLC column degradation, calibration drift, and volatiles in the lab air that affect ionization [119, 120, 121, 122]. Instrument variability can be global in nature, resulting in systematic bias. However, instrument variability can also be complex. Complex variability, unlike systematic bias, is local in nature. It stems from signal distortion due to transient stochastic events that occur during an HPLC-ESI-MS/MS run. For example, variability in ESI performance due to the mobile phase composition or flow rate fluctuations [123, 124] can cause complex instrument variability. Researchers deem the variability complex since each event affects only a narrow temporal window of an HPLC-ESI-MS/MS run, window duration varies, and windows can overlap.

3.3 Measurement Paradigms

3.3.1 Absolute Abundance Paradigm

In protoemics and peptidomics, absolute quantification is the measurement of the protein concentration per cell, ng ml^1 or the copy number of a protein or peptide per cell [15]. To determine the absolute abundance of a protein or peptide via HPLC-ESI-MS/MS workflows, researchers spike an isotopically-labeled peptide or other non-endogenous standard (or reference) into the prepared mixture. Researchers then quantify peptides

by comparing the MS^1 signal intensity of the reference to MS^1 signal intensity that is generated upon proteolytic cleavage of the target protein.

3.3.2 Relative Abundance Paradigm

Today, comparative proteomics researchers use relative protein and peptide quantification to reveal biological variation. To reveal biological variation, researchers ask the question, "For each protein (or peptide), is it differentially abundant between populations?" They answer the question using the relative abundance paradigm. The relative abundance paradigm transcends proteomics quantification technology. Researchers have used it in vastly different techniques, from quantitative biochemical assays to mass spectrometry.

Under the relative abundance paradigm, researchers compare two samples by measuring the relative abundance of corresponding proteins or peptides in two or more samples. Here, relative abundance is the abundance of a protein or peptide in one sample divided by the corresponding abundance in another sample [86]. In this definition, suppose we have two samples (A and B). Then relative abundance, as defined by Griffin et al. and Eidhammer et al. [88, 86] can be computed as

$$i_a/i_b, \tag{3.1}$$

where i_a is the intensity of peptide i in sample A and i_b is the intensity of the same peptide i in sample B. (Researchers commonly refer to the resulting value as either the peptide's ratio or fold change.). Researchers then identify differentially abundant peptides as those whose fold change between samples meets some criterion. Commonly, researchers use the de facto criterion, a fold change of two or greater (ratio > 2.0 or < 0.5) signifies differentially abundant peptides.

While the relative abundance computation has the virtue of beng simple, it is valid only if the following assumptions hold: the large majority of proteins or peptides to have

a ratio of 1/1; a small subset of proteins or peptides to change significantly; the linear range of the detector is broad enough to encompass the possible fold changes.

3.3.3 Proportionality Paradigm

Currently, comparative proteomics researchers do not employ compositional measurements (which I call the proportionality paradigm) for relative protein and peptide quantification to reveal biological variation. In fact, according to Lovell et al. in 2011, "we know of no one who has actual experimental data on both raw and compositional measurements in a molecular biology setting." [125] However, researchers in other disciplines, notably geology, commonly use the proportionality paradigm for measuring rock compositions. The compositional proportions are calcluated using

$$\left(i_{ja} / \sum_{j=1}^{n_a} i_{ja} \right) \tag{3.2}$$

where i_{ja} = abundance of peptide j in sample a and n is the number of peptides in the respective samples. Then, a peptide's relative proportions using fold changes is a straightforward formula. Here, a peptide's relative proportion is

$$\left(i_{ja} / \sum_{j=1}^{n_a} i_{ja} \right) \ / \left(i_{jb} / \sum_{j=1}^{n_b} i_{jb} \right) \tag{3.3}$$

where i_{ja} = abundance of peptide j in sample a, i_{jb}= abundance of peptide j in sample b and n is the number of peptides in the respective samples.

3.4 MS-Based Relative Quantification

Although mass spectrometry has become the workhorse of proteomics identification and quantification, mass spectrometers by themselves are inherently not quantitative instruments [46]. Nonetheless, researchers have devised various approaches to estimate

relative protein and peptide quantities from mass spectrometry data [86]. In general, relative abundance can be computed using labeled or label free schemes using MS^1 or MS^2 spectra. Figure 3.1 depicts two strategies and four representative schemes to compute relative abundances [126]. The following sub-sections describe the two schemes in detail.

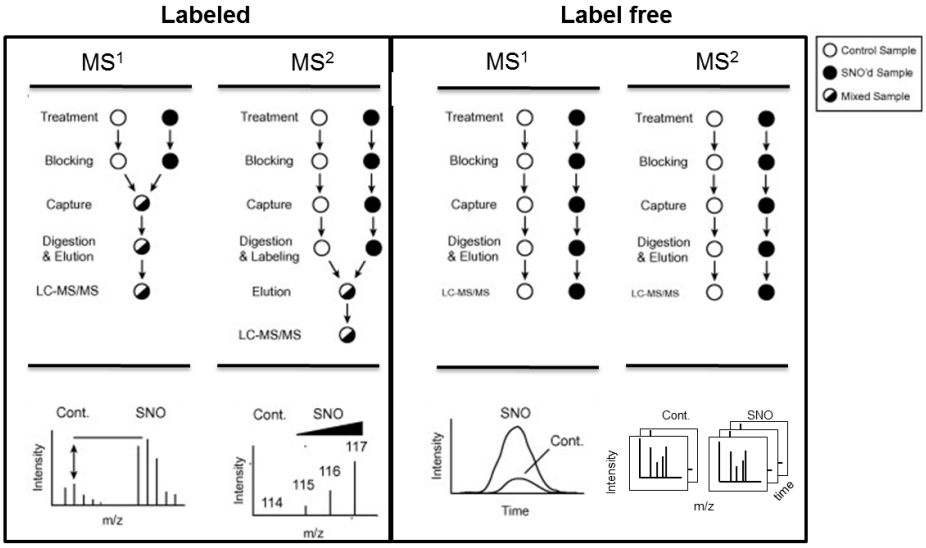


Figure 3.1: Relative abundance strategies - In this experiment, relative abundance is computed for S-nitrosylated proteins (SNO) in treatment vs. control samples. (Adapted from Thompson et al. - 2012) [126]

3.4.1 Labeled Relative Quantification

In labeled relative quantification, labeling techniques offer researchers a way to mitigate instrument variability. When researchers analyze two or more samples together, instrument variability disappears. However, the samples must be distinguishable. One way to make them distinguishable after analysis via HPLC-ESI-MS/MS is to label each sample's proteins or peptides with a substance that alters their mass, but does not change their physiochemical properties.

Different labeling techniques have different characteristics (see Table 3.1). First, there are three ways of incorporating labels into samples, metabolically, chemically, or enzymatically [86].

- Metabolic: In metabolic labeling, the labels are incorporated directly into living cells by growing them in a medium containing stable isotopes (SILAC) [127]. As the cells grow, proteins turn over; old proteins degrade; ribosomes manufacture new proteins. For comparing two cell cultures, researchers grow one culture with no stable isotope added (light) and the other with a stable isotope (heavy). They then combine, prepare, and analyze them simultaneously via HPLC-ESI-MS/MS
- Chemical: A chemical label covalently binds to the proteins or peptides.
- Enzymatic: A protease incorporates a label during digestion.

Second, some labeled techniques allow simultaneous analysis of just two samples while others allow up to eight. Third, labels can be either incorporated at the protein or peptide level. At the peptide level, the peptides can be ether endogenous or the result of enzymatic digestion. Fourth, and finally, quantification can occur at either the MS^1 or MS^2 level.

	Incorporation Method	No. of Samples	Incorporation Level	Quantification Level	Ref.
SILAC	Metabolically	2-3	Protein	MS	[127]
ICAT	Chemically	2	Protein	MS	[128]
^{18}O	Enzymatically	2	Peptide	MS	[129]
iTRAQ	Chemically	4 or 8	Peptide	MS/MS	[130]
TMT	Chemically	2 or 6	Peptide	MS/MS	[131]

Table 3.1: Commonly Used Labels for Relative Quantification - Adapated from Eidhammer et al. [86]. Note that the first column contains the name of the labeling technique, which is often an acronym. I define these acronyms in Appendix A.

While widely adopted by the proteomics community, labeling techniques have their limitations. The first limitation researchers face is cost. Amino acids labeled with stable

isotopes or chemical tags prove costly to synthesize. Thus, purchase of these can run from hundreds to thousands of dollars [132, 13]. Another limitation is that only certain biological sample types are suitable for labeling. For example, SILAC, arguably the most accurate stable isotope labeling method [15], is only applicable to experiments using cell culture models. However, despite being extremely expensive, researchers have described studies labeling a whole organism with stable isotopes (for example, mouse and worm) [133]. Unfortunately, those studies are only feasible in large research laboratories with extensive resources. Therefore, research on higher organisms, such as human and other animal studies, chemical tagging schemes, such as iTRAQ or TMT, must be used for stable isotope labeling. Unfortunately, the accuracy of iTRAQ and TMT for measuring relative abundances is not as high as SILAC [134]. Finally, labeling limits the number of samples compared in a single experiment. As shown in Table 3.1, on the low end, SILAC limits experiments to two samples and at the high end, iTRAQ limits experiments to eight samples.

3.4.2 Label Free Relative Quantification

In response to labeled relative quantification limitations, researchers are increasingly turning to label free schemes [13, 135, 132]. Label free schemes require no label, which make them less expensive, and more scalable. However, they require high repeatability and reproducibility, which is lacking in HPLC-ESI-MS/MS workflows. While, mass spectrometry's exponential advances in speed and resolution has helped repeatability and reproducibility, extraneous variability in HPLC-ESI-MS/MS remains a challenge.

Despite label free relative quantifications challenges in repeatability and reproducibility, researchers have devised two schemes that are commonly used today. As shown in Figure 3.1, two schemes historically underpin the label free relative quantification, spectral counting and intensity-based [119].

3.4.2.1 Intensity-based (MS^1)

Intensity-based label free relative quantification (iLFRQ) measures a peptide signal's intensity in MS^1 spectra. Studies show that the three dimensional peak area of a peptide signal's XIC should correlate to their original peptide concentration [136, 137, 138]. Figure 3.2 shows a representative XIC for a single peptide [139]. The shaded area of this peak represents peptide signal's AUC. Studies show that a peptide signal's AUC is linearly proportional to the concentration of the measured peptide ($R^2 = 0.991$) for peptides in the range of 10 fmol to 100 pmol. Furthermore, studies show that intensity-based schemes can detect proteins in complex mixtures with concentration ranges of 10 fmol to 1000 pmol ($R^2 = 0.9978$) [87, 136]. The principle is to first re-constructing a peptide signal's XIC, fit a polynomial curve to the XIC, derive the polynomial curve function, and then compute the area under the curve (AUC) using numerical integration [138]. However, computationally, numerical integration is expensive. Therefore, in practice, researchers use the trapezoidal approximation instead of numerical integration [140]. An alternative method is to estimate the XIC's AUC by summing its peak intensities [136]. Compared to integration and trapezoidal approximation algorithmss, XIC summing is computational inexpensive.

3.4.2.2 Spectral Counting (MS^2)

Spectral counting is based on the observation that the number of peptides identified from MS^2 spectra is proportional to the abundance of its corresponding protein in the original sample [29, 141, 142]. Empirically, more abundant proteins result in more identified peptides while less abundant proteins result in fewer identified peptides. Protein quantification simply counts the number of MS^2 spectra assigned to peptides within a given protein, without taking into consideration the peptide MS^1 signal intensity.

Early on, researchers computed spectral counting in a rather simple manner, simply summing the number of peptide identifications corresponding to each inferred protein

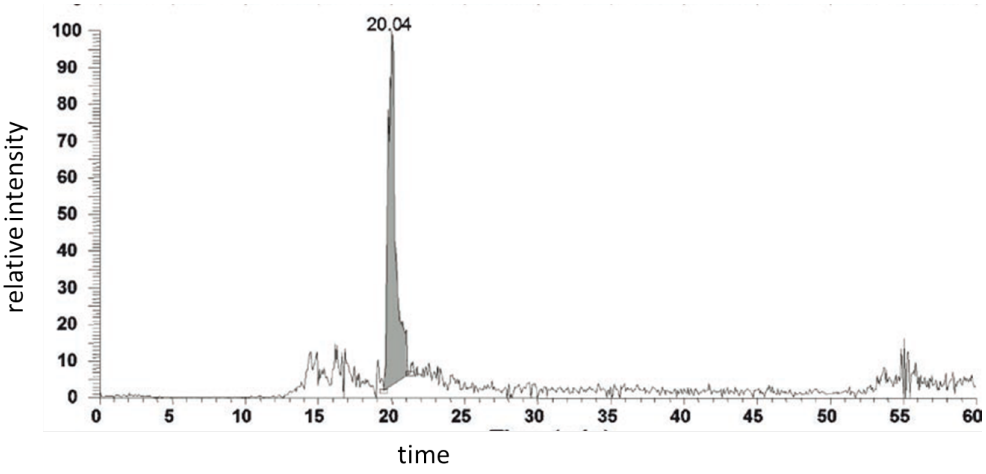


Figure 3.2: Peptide Signal XIC - Area / Intensity - XIC for a particular peptide showing a significant peak. The area of this peak represents the total ion intensity of the peptide. (Adapted from Wong et al. - 2010 [139]).

[132]. However, this assumes that the detection response via HPLC-ESI-MS/MS for each peptide in each protein is linear. Unfortunately, every peptide possesses different physiochemical properties that effect the intensity measurement via HPLC-ESI-MS/MS [13]. Therefore, even accurate quantification requires many spectra for a given protein [85], which precludes quantification for low abundance proteins. Furthermore, without normalization, spectral counting is biased towards longer proteins. This is because larger proteins, when enzymatically digested, will produce more peptides than shorter proteins. Consequently, researchers have developed numerous methods to normalize spectral accounts to estimate protein abundance, including, Normalized Spectral Abundance Factor (NSAF) [143], Protein Abundance Index (PAI) [144], Exponentially Modified Protein Abundance Index (emPAI) [145], and Normalized Spectral Index (SIn) [135]

3.4.3 Evaluation of Quantification Strategies

Each quantifcation strategy has it advantages and disadvantages. Here, I first present an overview of the labeled and label free relative quantification straegies' characteristics.

Second, for label free relative quantification, I present a detailed comparison of spectral counting and intensity-based strategies.

3.4.3.1 Overview of Quantification Characteristics

Here, I provide an overview of quantifcation strategies in Table 3.2. The first two columns in the table list the different quantification schemes and indicates their use in quantification. The remaining columns list characteristics for proteome coverage, sample preparation complexity, repeatability (Quant. precision - fourth column), dynamic range, number of samples possible, and cost per sample. Here, RSD in the fourth column stands for relative standard deviation, which is simply CV multiplied by 100 and displayed in percent units.

3.4.3.2 Comparing Label Free Relative Quantification Shemes

Researchers have conducted several studies specifically comparing spectral counting and intensity-based relative quantification. However, as Milac et al. point out, researchers employ different schemes to quantify and statistically summarize MS^1 and MS^2 data, thus complicating the comparison spectral counting and intensity-based schemes [147]. The terms spectral counting and intensity-based quantification are general schemes rather than defined methods.

Milac, et al. summarized their findings as follows. First, in 2005, Old et al. concluded that spectral counting is more sensitive than peptide signal intensities in finding differentially abundant proteins. However, they also conclude that peptide signal intensity is a more accurate estimate of protein ratios [122]. Thus, their conclusions were not definitive [147]. Second, in 2006, Zyabailov et al. concluded that spectral counting was more reliable than peptide signal intensity [143]. Finally, as described in the following section, in 2011, Milac et al. conducted a study using the CPTAC data set to detect biological variation. They concluded that intensity-based schemes were better than spectral counting schemes. Based on these three studies, overtime, iFLRQ has overtaken spectral

Type	Scheme	Proteome coverage	Sample preparation workflow complexity	Quant. precision	Quant. dynamic range (log10)	Samples	Cost per sample
Relative Label free	Spectral counting	High	Low	Low (>30% RSD)	2-3	Unlimited	Low
	Intensity-based	High	Low	Medium (10-30% RSD)	2-3	Unlimited	Low
Relative Labeled	SILAC	High	High	High (<10% RSD)	1-2	2-3	High
	^{18}O	Medium	Medium	Medium (10-20% RSD)	1-2	2	Low
	ICAT	Low	Medium	High (<10% RSD)	1-2	2	Medium
	iTRAQ/TMT	Medium to high	Medium	High (<10% RSD)	1-2	2-8	High
Absolute	Internal standard	Medium	Medium	Medium (10-20% RSD)	1-2	Unlimited	Low

Table 3.2: Overview of the characteristics of different HPLC-ESI-MS/MS quantification strategies (Adapted from Xie et al. - 2011 [146])

count in reliably detecting biological variation.

CPTAC Data Set Relative Quantification

This section describes the only related study found to date for CPTAC Data Set quantification. In 2011, Milac et al. studied subset of the data collected for the CPTAC study, comparing the samples designated QC2, with the UPS1 spike-in samples, designated A, B, C, D and E [147]. The four laboratories participating in the study each collected three technical replicate observations of the QC2, A, B, C, D and E samples for a total of twelve (12) observations of each. They searched the $MS2^2$ spectra using X!Tandem against the same protein database used by the CPTAC authors, which includes both target and decoy (reversed) protein sequences. They quantified the CPTAC data by spectral count and peptide signal intensity using their software framework Sahale. Finally, they normalized the spectral counts in each sample by the total spectral count, and the peptide signal intensities by the median scale.

Figure 3.3 provides Milac et al.'s results. First, they define three levels of rollup for HPLC-ESI-MS/MS data.

- Species: The base rollup level.
- Peptide: The collection of all species with the same primary amino acid sequence.
- Protein: The collection of all species that by virtue of their primary amino acid sequence possibly originated from a particular protein.

Then, for each level, they provide the spectral count and intensity-based (which they call ion abundance) data for numbers of UPS1 and yeast proteins found to be present in greater and lesser abundance in samples A-E relative to QC2 at FDR level 0.05 (see Figure 3.3) [147].

Interestingly, Milac et al. found that the intensity of yeast proteins decreases as a function of increasing UPS1 spike-in (see the grey shaded column in Figure 3.3).

		sample (vs QC2)	UPS1 ⇑	UPS1 ⇓	yeast ⇑	yeast ⇓	total
species	ion abundance	A	0	0	0	0	0
		B	2	0	0	0	2
		C	17	0	0	5	22
		D	35	0	0	12	47
		E	41	0	2	347	390
	spectral count	A	0	0	0	1	1
		B	3	0	0	1	4
		C	21	0	2	7	30
		D	35	0	2	4	41
		E	42	0	27	69	138
peptides	ion abundance	A	0	0	0	0	0
		B	3	0	0	0	3
		C	16	0	0	5	21
		D	35	0	0	18	53
		E	41	0	2	403	446
	spectral count	A	0	0	0	1	1
		B	8	0	1	0	9
		C	23	0	4	9	36
		D	35	0	5	6	46
		E	42	0	20	71	133
proteins	ion abundance	A	0	0	0	0	1
		B	3	0	0	0	3
		C	15	0	0	5	20
		D	35	0	1	31	67
		E	40	0	5	522	567
	spectral count	A	0	0	0	0	0
		B	5	0	0	1	6
		C	24	0	1	8	33
		D	34	0	1	6	41
		E	42	0	11	83	136

Figure 3.3: CPTAC Study 6 Quantifcation (Figure from Milac, et al. - 2012)[147].

After a thorough investigation, they concluded that this was the result of increasing ion competition as the amount of UPS1 spike-in protein increases [147].

Finally, Milac et al. summarized their findings, "There are a variety of reasons to favor ion abundance for quantification. The statistical models for protein rollup by Clough et al. [148], for example, are implicitly based on ion abundances. Also, as noted by Podwojski et al. [149] and Lundgren et al. [150], spectral counts may be dominated by a few proteins having a large number of counts, and the spectral count breaks down as a statistical quantity when very few counts are observed. Although the estimated ion abundance of an identified species is subject to low signal and the stochastic nature of the CID sampling, it has the potential to more robustly quantify seldom-seen species."

3.5 Normalization

Normalization has several meainings. For example, in statistics, normalization can mean adjusting the values measured on different scales to a nominal value. In measurements, normalization attempts to mitigate extraneous variability by making two or more distributions similar. In the context of HPLC-ESI-MS/MS workflows, this means similar chromatographic intensity distributions with the goal of biological variation being the only observable difference between two runs.

Normalization is a concept well known in the area of genomics studies using gene expression microarrays [118, 151, 152, 153]. As a consequence, many methods developed for microarray data have also been adapted for normalizing peptide data produced with HPLC-ESI-MS techniques [118, 151, 152, 153, 117, 154, 155]. The underlying assumption for applying these techniques is that the total or mean/median peak abundances should be equal across different experiments, in this case between HPLC-ESI-MS/MS analyses. Thus, in MS-based studies, the general form for computing fold changes incorporating normalization is:

$$(i_{jx}/S_{jx}) \,/\, (i_{jy}/S_{jy}) \tag{3.4}$$

where i_{jx} = intensity of ion j in run x, i_{jy}= intensity of ion j in run y, and S_{jx} and S_{jy} are scaling factors computed by a global function for runs x and $y$respectively.

3.5.1 Normalization Method Descriptions

The following subsections describe ten normalization methods. The first four were described for the proteomics community by Callister et al. in 2006 [117]. Because the authors described the methods in detail, I provide the algorithm description for each. The next six normalization methods were described for the peptidomics community by Kultima et al. in 2009 [118]. The authors describe the methods in less detail, thus I do not provide the algorithm descriptions here, but give a general description.

3.5.1.1 Median Scale (Central Tendency)

Median scale (MedScale) normalization centers peptide signal intensities around the median to adjust for the effects of independent systematic bias. It assumes that systematic bias is constant and has often been defined as global or total intensity normalization by the genomics microarray community [153, 156, 157]. Callister et al. originally defined this as central tendency, which is a broader definition than median scale. The broader definition allows for centering a mean or other fixed constant [117]. However, in practice, the centering tends to be around the median. The central tendency normalization algorithm's steps follow.

- Log transform the peptide signal intensities.
- Plot transformed peptide signal intensities in an MA plot (see Appendex C.2).
- Subtract the center (mean, median, or fixed constant) from each peptide signal (See equations in C.1.2).
- De-convolute peptide signals into their original scale (See equations in C.1.2).

3.5.1.2 Linear Regression (MA)

Linear regression using a MA (RegrMA) plot applies least squares regression to a scatterplot. It assumes that systematic bias is linearly dependent on the intensity of peptide signals [158, 159]. For example, systematic bias resulting from HPLC column carry over can inflate measured peptide signal intensities [151, 160]. The resulting first-order regression is used to compute each normalized peptide signal intensity [117]. The algorithm for linear regression (MA) normalization follows.

- Log transform the peptide signal intensities.
- Plot transformed peptide signal intensities in an MA plot (see Appendix C.2).
- Fit a regression line to the MA plot and predict peptide signal intensities
- Subtract the predicted peptide signal intensities from each peptide signal (See Appendix C.1.2 equations).
- De-convolute peptide signals into their original scale (See Appendix C.1.2 equations).

3.5.1.3 LOESS (MA)

LOESS normalization using a MA plot (LOESSMA) applies a least squares local regression to a scatterplot. LOESS is is a commonly used generalization of LOWESS [159], which stands for locally weighted regression. Researchers use the terms interchangeably. I choose LOESS because it is the more general form.

LOESSMA assumes that systematic bias is not linearly dependent on the intensity of peptide signals. This nonlinearity can result from the effects of ion suppression on measured peptide abundances, or on measured peptide abundances approaching detector saturation or background [117]. Normalized peptide ratios were calculated in the same manner as the linear regression technique. The fraction of peptides for inclusion around

the peptide ratio to be normalized was set at 0.4. The value of this fraction increases in magnitude up to 1.0 with an increasingly greater number of peptides included in the subdivision surrounding the peptide ratio to be normalized. Their selection of 0.4 was based on the observation that values <0.4 resulted in plotted regression lines not being smooth, while values >0.4 resulted in plotted regression lines being approximately linear. The algorithm for LOESS (MA) normalization follows.

- Log transform the peptide signal intensities.
- Plot transformed peptide signal intensities in an MA plot (see equations C.2).
- Fit a LOESS regression line to the MA plot and predict peptide signal intensities
- Subtract the predicted peptide signal intensities from each peptide signal (see Appendix C.1.2 equations).
- De-convolute peptide signals into their original scale (see Appendix C.1.2 equations).

3.5.1.4 Quantile

Quantile normalization was originally designed for multiple high-density arrays, such as those created by Affymetrix Gene-Chip (Santa Clara, CA) [151]. Callister et al. adapted quantile normalization for use by the proteomics community in 2006 [117]. Researchers based quantile normalization on the premise that they expect the distribution of peptide abundances in different samples to be similar. Furthermore, differences can be accounted for by adjusting these distributions [117]. The quantile normalization algorithm, as defined by Callister et al. follows.

- Assign each sample replicate to a column and place the abundance values for peptides common to all replicates in the same row.
- Assign an index to each peptide abundance value in the column.

- Sort each column.
- After sorting all replicates by abundance value, substitute the mean arbitrary abundance for the abundance value in each row.
- Restore the original order of the assigned indeces for each replicate, thus normalizing relative abundance for a given peptide.

3.5.1.5 DeCyder MS Normalization (DeCyder and Spike)

DeCyder provides two types of normalization methods, global normalization and spike-in. As described by Kultima et al. [118], "for global normalization, the intensity distributions of the peaks detected in each sample were used for normalization [161]. For spike-in normalization, the shift of data for each analysis was calculated by fitting a linear regression to the spiked-in peptides in each analysis and then subtracting the average regression coefficient across all analyses from each run. The normalized values produced with the DeCyder MS2.0 software were exported for both normalization procedures."

3.5.1.6 Reference Run

Reference run (RefrRun)is a type of global intensity normalization. Here, researchers choose one analysis as a reference and then normalize all other runs to the chosen one [118]. Then, they select a normalization constant from the median intensity ratios for peaks matched between runs [120, 162, 163]. The choice of the reference data set can be arbitrary or based on some criterion. Kultima et al. selected the run with the most matched peptides as the reference set.

3.5.1.7 Variance Stabilization Normalization (VSN)

Variance stabilization normalization (VSN) was originally normalizing single and two-channel gene expression microarray data by Huber, et al. [152]. Kultima et al. adapted VSN for peptidomics [118]. They described the the algorithm as "The function calibrates

sample-to-sample variations through shifting and scaling of intensities and transforms the intensities to a scale where the variance is approximately independent of the mean intensity."

3.5.1.8 Linear Regression

Kultima et al. adapted RegrMA to create linear regression (Regr) [118] normalizaiton. As with LOESSMA, Regr assumes systematic bias to be linearly dependent on the magnitude of peak intensities. However, instead of constructing MA plots, Kultima et al. constructed a single reference analysis by taking the median peak intensity for all matched peaks. They describe Regr as "normalization was then performed on this constructed reference by applying least squares regression to each individual analysis. Based on the linear regression equation new values were predicted for each analysis, taking both intercept and slope of the regression line into account." [118].

3.5.1.9 Regression Run Order

Kultima et al. observed that the analysis order of the HPLC-ESI-MS/MS experiments contributes to bias in the data [118]. Thus, they posited that normalization based only on analysis order alone would not correct for global differences, such as differences in amount/concentration of one sample compared with the rest of the samples. Therefore, they combined linear regression with analysis order (RegrRun) normalization to address different types of biases. Kultima et al. define their RegrRun algorithm as follows. "Loess regression was fitted using the function loessFit() in the R package Limma [164] (Version 2.12.0) for each matched peak versus the analysis order, and the mean value across all analyses for each peak was then added to retain the native intensity dimension. For each matched peak a certain proportion of neighbors (analyses), weighted by their distance to the measurement, was used for controlling the smoothness of the fit, the span. A high span value gives more smoothness, and a value of 1 returns values similar to those of linear regression. Equal weight was given for all types of samples, and default

settings were used for the loessFit() function (span of 0.3)." [118]

3.5.2 Normalization Method Evaluation

Kultima, et al. evaluated the performance of ten normalization methods [118]. They used three data sets of endogenous peptides analyzed via nano-LC mass spectrometry. Briefly, the data sets were derived from brain tissue (Mouse, Rat, and Quail), and each data set consisted of 35 to 45 runs. "For the variation between technical replicates (Pool) and the variation including both technical and biological variation, biological samples belonging to the treatment group (BioRep) were ranked based on their average percent reduction in median log2 standard deviation (SD) and reduction of log2 pooled estimate of variance (PEV) compared with raw data. In the Rat and Quail data sets, the BioRep only consisted of different biological samples, whereas the Mouse data set included one to three technical replicates of biological samples. For the Quail data set, which lacked spiked-in peptides, only nine methods were evaluated, resulting in a slight bias in the average rank across the different data sets, but this had only a minor effect on the final result. To get a more general overview of how much the different normalization methods decreased median SD and PEV compared with the raw data, the average percent reduction for the three data sets was calculated." Their results are listed in Tables 3.3 and 3.4.

To compare the performance of normalization methods of data in terms of detecting differential abundance, Kultima et al. fitted a mixed linear model to each data set. They included or omitted the block term in the model. They made the following observations.

- The largest fraction of differentially expressed peaks for all three data sets was found using the method RegrRun.
- Spike in normalization yielded the least favorable results in both data sets.

- RegrRun decreased median standard deviation by 42-43%. The second top performing normalizaiton method was Regr, which decreased median standard deviation by 38%. Other global normalization methods reduce the median SD by 15-28%.

Method	Mouse Data Set				Rat Data Set				Quail Data Set			
	PEV		Median SD		PEV		Median SD		PEV		Median SD	
	Pool	Bio. Rep.	Pool	Bio. Rep.	Pool	Bio. Rep.	Pool	Bio. Rep.	Pool	Bio. Rep.	Pool	Bio. Rep.
Raw	0.32	0.51	0.42	0.54	0.10	0.7	0.25	0.67	1.44	1.50	0.65	0.75
DeCyder	0.22	0.35	0.29	0.40	0.08	0.43	0.19	0.48	1.25	1.35	0.45	0.68
Spike	0.24	0.39	0.31	0.45	0.08	0.63	0.21	0.63	n.a.	n.a.	n.a.	n.a.
LOESSMA	0.22	0.34	0.28	0.39	0.08	0.42	0.20	0.46	1.02	1.18	0.41	0.66
RegrMA	0.23	0.35	0.29	0.39	0.08	0.42	0.20	0.48	1.15	1.24	0.49	0.70
MedScale	0.23	0.36	0.31	0.41	0.08	0.45	0.21	0.51	1.28	1.38	0.47	0.69
Quantile	0.24	0.35	0.31	0.40	0.08	0.43	0.21	0.51	1.08	1.22	0.53	0.69
RefRun	0.22	0.35	0.28	0.40	0.08	0.43	0.19	0.47	1.25	1.36	0.45	0.69
Vsn	0.23	0.36	0.30	0.40	0.08	0.41	0.20	0.46	1.25	1.33	0.44	0.68
Regr	0.20	0.30	0.28	0.37	0.05	0.31	0.16	0.43	0.58	0.67	0.36	0.53
RegrRun	0.11	0.21	0.17	0.28	0.06	0.24	0.17	0.36	0.55	0.54	0.42	0.45

Table 3.3: The median SD and PEV for the three data sets when omitting (Pool) and including biological variability (BioRep) are shown. (Adapted from Kultima et al. - 2009 [118])

Method	Mouse Data Set				Rat Data Set				Quail Data Set			
	PEV		Median SD		PEV		Median SD		PEV		Median SD	
	Pool	Bio. Rep.	Pool	Bio. Rep.	Pool	Bio. Rep.	Pool	Bio. Rep.	Pool	Bio. Rep.	Pool	Bio. Rep.
DeCyder	0.31	0.29	0.3	0.27	0.22	0.37	0.25	0.28	0.13	0.08	0.31	0.08
Spike	0.26	0.21	0.25	0.16	0.19	0.06	0.15	0.05	n.a.	n.a.	n.a.	n.a.
LOESSMA	0.32	0.31	0.32	0.28	0.22	0.39	0.22	0.31	0.29	0.16	0.36	0.10
LinRegMA	0.31	0.31	0.30	0.27	0.21	0.39	0.20	0.27	0.20	0.13	0.24	0.05
MedScale	0.28	0.28	0.26	0.25	0.22	0.35	0.19	0.24	0.11	0.06	0.28	0.07
Quantile	0.25	0.30	0.26	0.26	0.20	0.37	0.18	0.23	0.25	0.13	0.18	0.06
RefRun	0.32	0.29	0.33	0.26	0.23	0.37	0.24	0.29	0.13	0.07	0.32	0.07
Vsn	0.28	0.28	0.29	0.26	0.19	0.4	0.21	0.3	0.13	0.09	0.33	0.08
Regr	0.37	0.41	0.34	0.32	0.49	0.54	0.37	0.35	0.59	0.51	0.45	0.28
RegrRun	0.66	0.58	0.59	0.48	0.39	0.66	0.33	0.46	0.62	0.61	0.36	0.39

Table 3.4: The the reduction in median SD and PEV for the three data sets when omitting (Pool) and including biological variability (BioRep) are shown. (Adapted from Kultimat et al. - 2009 [118])

3.5.3 Overfitting

When normalizing, researchers should consider overfitting [89]. For example, in the relative abundance paradigm, most of the ratios (fold changes) should be one, or nearly one because the abundances for the vast majority of proteins (or peptides) should not change between samples. The risk is moving too much of the data towards one. Doing so obscures real (biological) variation. For example, too many iterations of the LOESS normalization method can overfit data [86]. Furthermore, overfitting can have the opposite effect; researchers can detect biological variation when there is none [86].

3.6 iLFRQ Software Frameworks

This section describes commonly used iLFRQ software frameworks. Here, I describe non-commercial software frameworks. I chose not to describe commercial iLFRQ software frameworks because, in general, commerce generally protects the details of their methods as intellectual property.

Table 3.4 lists each software framework described. The first column in the table lists the software framework's name. The second and third columns list the type of normalization employed by the software framework. A question mark next to two normalization methods indicates that the method was not described in the original publication, but described by researchers outside the software framework research team. The fourth and fifth columns list which instruments and file formats are accepted. The sixth column contains yes or no, indicating whether or not the software framework offers a graphical user interface (GUI). The seventh column also contains yes or no, but indicates whether the software framework is published as open source. The eighth columns lists the programming languages used for creating each software framework. The ninth column lists the operating systems on which each software frameworks run. Finally, the last column contains the reference for the publication describing the software framework.

Software	*Normalization*		*Instruments*	*Input files*	*GUI*	*Open Src?*	*Lang.*	*OS*	*Ref.*
MaxQuant	Global	Median Central Tendency?	LTQ, Orbitrap, (Thermo)	.raw, mzXML	Yes	No	C#	Windows	[16]
msInspect	Global	Simple linear regression	Any via mzXML, mzML	mzXML, mzML	Yes	Yes	Java	Windows, OSX, Linux	[75]
MZmine2	Global	Internal standard	Any via mzXML or mzML	.raw, mzData, mzXML,mzML	Yes	Yes	Java	Windows, Linux, OSX	[76]
OpenMS	Global	Median?	Any via mzXML or mzML	.dta, mzData, mzXML, mzML	Yes	Yes	C++	Windows, Linux, OSX	[165]
PEPPeR	Global	Scale by total extracted intensity	Any via mzXML	mzXML	No	Yes	Perl	Windows	[166]
SuperHirn	Regional	Modified Central Tendency	Any via mzXML or mzML	Via TPP	No	Yes	C++	Linux, OSX	[121]

Table 3.5: Open source iLFRQ software frameworks and their characteristics.

Cox et al. introduced MaxQuant in 2008 as a software framework for SILAC relative quantification. Since then, follow on releases included iLFRQ [16]. MaxQuant is arguably the most popular ilFRQ software framework used by the proteomics community. Note that the second column in Table 3.5 contains a question mark indicating that perhaps MaxQuant uses central tendency median normalization. Unfortunately, Cox et al. has not provided a description of MaxQuant's normalization methods for label free quantification. However, others [76] posit that MaxQuant uses central tendency median normalization. Therefore, I include it here.

Bellew et al. introduced msInspect in 2006 as a suite of software and algorithms for viewing and processing data from HPLC-ESI-MS/MS [75]. Furthermore, they use msInspect as a platform for proteomics application development and invite other developers to integrate their own algorithms. All of their tools built on the msInspect platform are free, cross-platform, and open source [75]. This software has modular components for peptide signal extraction, retention time alignment, and normalization algorithms. The developers claim that these algorithms can be replaced without altering the framework [167]. In practice, I found that this is not the case.

Katajama et al. introduced MZmine in 2005 as open-source software toolbox for HPLC-ESI-MS/MS metabolomic data processing [168]. The first version implemented simple methods for data analysis and visualization [168, 169]. The latest release, MZmine2, was completely redesigned to support modularity and enhance performance [76]. Unfortunately, MZmine2's normalization requires an internal standard. Thus, while MZmine2 outshines other software packages in visualization, its use in comparative proteomics is limited.

Sturm et al. introduced OpenMS in 2008 as a framework designed for rapid application development in proteomics data analysis [165]. It includes modules for data analysis, providing basic data structures, visualization and sophisticated algorithms. While the authors indicate that OpenMS is cross-platform, it is written in C++ and thus must be compiled for each type of operating system. The second column in Table 3.5 contains

a question mark indicating that perhaps OpenMS uses median normalization. Unfortunately, Sturm et al. did not provide a description of OpenMS's normalization methods. However, others [76] indicate that OpenMS uses median normalization. Therefore, I include it here.

Jaffe et al. introduced PEPPeR in 2006 as a bundle of algorithms for pattern matching peptide signals across multiple HPLC-ESI-MS/MS [166]. It is bundled together with other algorithms, data acquisition strategies, and experimental designs. Of particular interest is the bundling of MapQuant. MapQuant first scales peptide signals using the summed intensity of all peptide signals identified by MapQuant for the experiment and then log2 transforms results. Unfortunately, PEPPeR only runs on Windows machines [170].

Mueller et al. introduced SuperHirn in 2007 as a tool to quantitatively analyze multi-dimensional HPLC-ESI-MS/MS data in a label free strategy [121]. The authors describe their normalization as global, but others suggest that their normalization may be regional [149]. SuperHirn is now part of the the Trans-Proteomics Pipeline (TPP). TPP is a software suite distributed by the Seattle Proteome Center [106]. Unfortunately, the TPP requires the installation of several software packages such as Apache to run in a web-based environment.

Chapter 4

The Proportionality Paradigm for iLFRQ

Simplicity is the ultimate sophistication. - Leonardo Da Vinci

4.1 Introduction

The relative abundance paradigm has been the workhorse for detecting biological variation via HPLC-ESI-MS/MS for the past decade [88]. However, when trying to use the relative abundance fold change paradigm in a serial dilution experiment, I came to the conclusion that the relative abundance paradigm is ill-suited for revealing biological variation in samples analyzed via HPLC-ESI-MS/MS and iLFRQ. In response, I decided to use the proportionality paradigm. Recall from Chapter 3, it first computes a peptide's compositional proportion within a sample. Then it detects biological variation by comparing a peptide's compositional proportionality between samples. Using the proportionality paradigm reveals biological variation where the relative abundance paradigm fails.

The remainder of this chapter is organized as follows. Section 4.2 motivates the use of proportionality paradigm in the context of a simple experiment, mimicking a biomarker study. I demonstrate how a simple pipetting error can cause the relative abundance paradigm to fail. Section 4.3 presents the proportionality paradigm. Using the same protocol, I demonstrate how the proportionality paradigm overcomes challenges where the relative abundance paradigm fails. Section 4.4 extends the proportionality

paradigm, applying it to mass spectrometry data. In section 4.5, I address commonly asked questions about the proportionality paradigm.

4.2 Motivation

The relative abundance paradigm is straight forward, but problematic for detecting biological variation in complex biological samples. Recall from Chapter 3, it is the simple ratio (fold change) of the same peptide's abundances in two samples. Differentially abundant peptides are those whose fold change between samples meets some criterion. Commonly, researchers use the de facto criterion, a fold change of two or greater (ratio > 2.0 or < 0.5) signifies differentially abundant peptides.

When using the relative abundance paradigm (see Chapter 3.3.2), iLFRQ workflows produce results with poor repeatability and reproducibility [111]. As a result, they generate excessive false positive and false negative results. The false positive results will eventually be discarded via hypothesis-driven experiments but at the cost of valuable researcher time. False negatives are worse because researchers never look at rejected peptides; thus researchers can draw incorrect conclusions and can miss possible insights.

Why do these bad results occur? I posit that they do because the relative abundance paradigm is ill-suited to discover proteomic (and peptidomic) biological variation. Here, I illustrate why by using a simple biomarker study example. In this example, we construct samples according to a prescribed protocol but make an error carrying out the protocol's instructions.

To begin, suppose a protocol instructs us to construct two samples (A and B) having the same three peptides (1, 2 and 3) with, respectively, a prescribed 900, 900 and 450 units of those peptides (see Figure 4.1). Here, A and B represent different populations with no abundance differences between them. To reveal biological variation, we then ask the question *"For each peptide, is it differentially abundant between populations?"* We answer the question using the relative abundance paradigm. Here, a peptide's relative

abundance is the ratio:

$$i_a/i_b \tag{4.1}$$

where i_a = number of units of Peptide i in Sample A and i_b = the number of units of the same Peptide i in Sample B. We conclude that peptides are differentially abundant if their fold change is greater than some threshold. Here, we use the de facto fold change of two or greater (ratio > 2.0 or < 0.5). Because the prescribed peptides amounts are the same, of course, the answer is *"no"*. Table 4.1 Column 1 contains the prescribed relative abundances between Samples A and B. In Table 4.2, Column 1 answers the question *"For each peptide, is it differentially abundant between Samples A and B?"*

Now suppose the protocol instructs us to construct a third sample (C), representing another population. It also contains 900 units of Peptides 1 and 2, but only 150 units of Peptide 3. The intention is to mimic a candidate biomarker (Peptide 3) which decreases in the third population by three-fold. We again ask, *"For each peptide, is it differentially abundant between populations?"* Of course, the answer is again *"no"* for Peptides 1 and 2, as well as, Peptide 3 between A and B; however, the answer should be *"yes"* for Peptide 3 between A and C (450/150) as well as B and C (450/150). Table 4.1 also contains the prescribed relative abundances between Samples A and C as well as Samples A and B. In Table 4.2, Column 2 answers the question *"For each peptide, is it differentially abundant between Samples A and C?"* and Column 3 answers the question *"For each peptide, is it differentially abundant between Samples B and C?"* The two prescribed *"yes"* answers are in bold typeface.

In an all too realistic scenario, next suppose pipetting errors cause the actual abundances to differ from the prescribed abundances. We observe that Samples A and B are compositionally the same and, therefore, we decide to prepare a single mixture. From the single mixture, we then divide it into two samples by pipetting into eppendorf tubes labeled A and B. Unfortunately, we pipette incorrectly, causing eppendorf tube B to

contain substantially less sample than eppendorf tube A (see Figure 4.2). In this scenario, we prepare the mixture for Sample C as prescribed and pipette the mixture into an eppendorf tube labeled C.

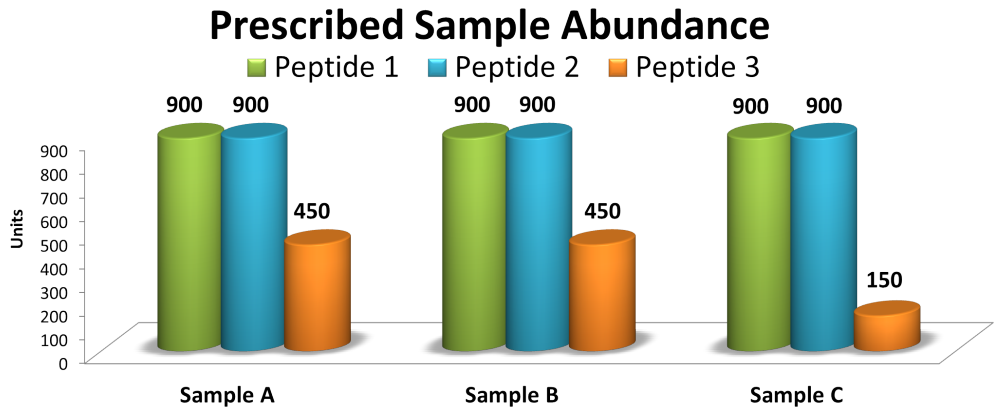


Figure 4.1: A simple biomarker study example of a protocol for constructing samples three samples, each with three peptides.

Analyte	A vs B	A vs C	B vs C
Peptide 1	1.00	1.00	1.00
Peptide 2	1.00	1.00	1.00
Peptide 3	1.00	2.00	2.00

Table 4.1: Prescribed Fold Changes

Analyte	A vs B	A vs C	B vs C
Peptide 1	no	no	no
Peptide 2	no	no	no
Peptide 3	no	**yes**	**yes**

Table 4.2: Differentially Abundant?

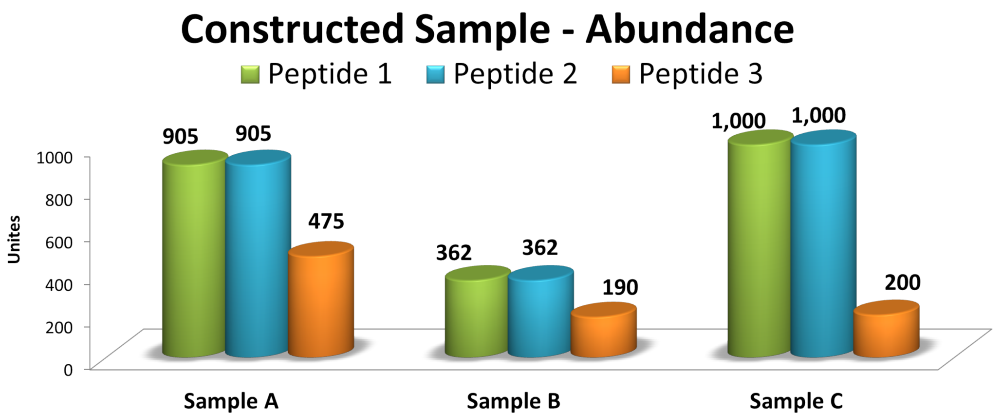


Figure 4.2: Three constructed samples based on the protocol in 4.1. Note that pipetting error caused Sample B to have incorrect overall abundances.

Analyte	A vs B	A vs C	B vs C
Peptide 1	2.50	0.91	0.36
Peptide 2	2.50	0.91	0.36
Peptide 3	2.50	2.38	0.95

Table 4.3: Constructed Fold Changes

Analyte	A vs B	A vs C	B vs C
Peptide 1	**yes**	no	**no**
Peptide 2	**yes**	no	**no**
Peptide 3	**yes**	yes	***no***

Table 4.4: Differentially Abundant?

Again, we apply the relative abundance paradigm and generate the fold change ratios listed in Table 4.3. Again, using the de facto fold change of two or more, as detailed in Table 4.4, the fold changes in column B vs C are fortunately correct (one *"yes"* which is a true positive and two *"no"* answers which are true negatives). However, fold changes in bold typeface (two *"yes"* answers, which are false positives) and one in bold and italic typeface (a *"no"* answer, which is a false negative) in column A vs C is incorrect. Fortunately, hypothesis-driven experiments should discard the two false positive answers. Unfortunately, the *"no"* answer for Peptide 3's differential abundance between B and C is worse because Peptide 3 will not be investigated further. Thus, we will miss a key

insight.

Using the relative abundance paradigm to compare A and B, by definition, the answer for each peptide in column A vs B is *"yes"* (~2.5 = 905/362). However, based on sample composition, for example, in a biomarker discovery study, I could also argue that the answer for each peptide is *"no"* because both samples originate from the same parent sample and, therefore, a *"yes"* answer is a false positive (bold and italicized in Table 4.4). Unfortunately, the relative abundance question, *"For each peptide, is it differentially abundant between A and B?"* has more than one sensible interpretation and so this question is ambiguous.

4.3 Applying The Proportionality Paradigm

Because asking whether peptides are differentially abundant using the relative abundance paradigm is ambiguous, we, the collective proteomics (and peptidomics) community, need to ask a different question: *"Are the constituent peptides compositionally different between two samples?"* [171] To answer this question, we first measure peptides' proportions within a sample and then compute their relative proportions (ratios) or statistically test for significant differences across samples. (Because the example described is simple, that is, it does not contain replicates, statistical tests are not valid. Therefore, I use relative proportions instead of statistical tests.) The same Samples A, B, and C depicted previously in Figure 4.1 are depicted again in Figure 4.3, but this time measured using proportions.

A peptide's relative proportions using fold changes is a straightforward formula. As described in Chapter 3, a peptide's relative proportion is

$$\left(i_{ja} / \sum_{j=1}^{n_a} i_{ja} \right) \; / \left(i_{jb} / \sum_{j=1}^{n_b} i_{jb} \right) \tag{4.2}$$

where i_{ja} = abundance of peptide j in sample a, i_{jb}= abundance of peptide j in sample

b and n is the number of peptides in the respective samples. That is a peptide's relative proportion is measured by first its computing the peptide's compositional proportion and then their fold changes. Table 4.5 contains prescribed fold changes using the proportionality paradigm and Table 4.6 contains prescribed answers to the question, *"Are the constituent peptides compositionally different between two samples?"*

When applying the proportionality paradigm to the constructed sample (see Figure 4.4), the fold changes in Table 4.7 yield the three correct *"no"* answers in column A vs B.

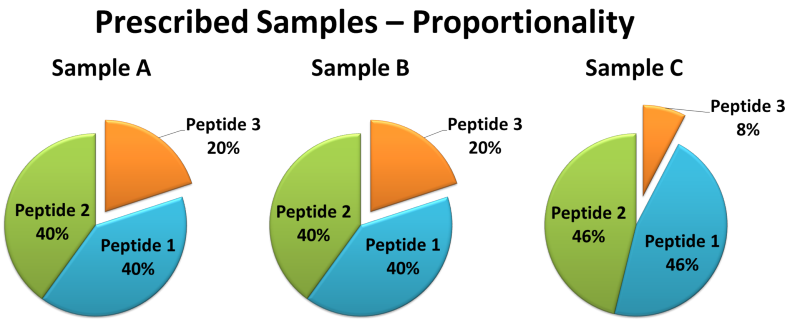


Figure 4.3: A simple biomarker study example of a protocol for constructing samples three samples, each with three peptides. In contrast to Figure 4.1, here, the peptide amounts are shown as proportions.

Analyte	A vs B	A vs C	B vs C
Peptide 1	1.00	0.83	0.83
Peptide 2	1.00	0.83	0.83
Peptide 3	1.00	2.50	2.50

Table 4.5: Prescribed Fold Changes

Analyte	A vs B	A vs C	B vs C
Peptide 1	no	no	no
Peptide 2	no	no	no
Peptide 3	no	**yes**	**yes**

Table 4.6: Differentially Proportional?

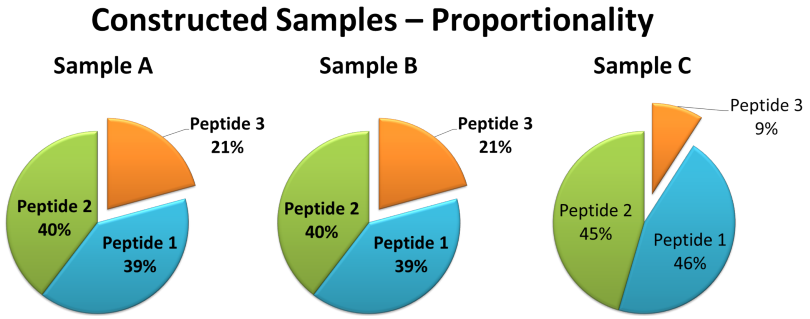


Figure 4.4: The same three constructed samples depicted previously in Figure 4.1. Note that pipetting error caused Sample B to have incorrect overall abundances, but proportionally they are the same.

Analyte	A vs B	A vs C	B vs C
Peptide 1	1.00	0.87	087
Peptide 2	1.00	0.87	0.87
Peptide 3	1.00	2.39	2.39

Table 4.7: Constructed Fold Changes

Analyte	A vs B	A vs C	B vs C
Peptide 1	**yes**	no	**no**
Peptide 2	**yes**	no	**no**
Peptide 3	**yes**	yes	***no***

Table 4.8: Differentially Proportional?

4.4 The Proportionality Paradigm and Mass Spectrometry

While the previous sections motivated and described the application of the proportionality paradigm in the context of measuring relative proportions of samples, will the proportionality paradigm work measuring relative proportions between mass spectrometry runs? The relationship between a peptide's abundance in the sample and the detector measurement are expected to be linear [168]. Unfortunately, as described in Chapter 3, mass spectrometry is inherently susceptible to extraneous variability. Thus, for iLFRQ, resulting measurements must be normalized. Therefore, with the proper normalization

methods, the answer is *"yes"*.

4.5 Discussion

Paradigm shifts are rarely fully embraced when introduced. Therefore, it is not surprising that my colleagues pose interesting and challenging questions about using the proportionality paradigm in for revealing biological variation. While my colleagues' backgrounds are diverse, commonly, they ask the same two questions.

The first question is *"Is applying the proportionality paradigm in the context of comparative proteomics simply a normalization method?"* After all, in the scenario described earlier, the proportionality paradigm, in effect, mitigates systematic bias from loading amount differences. Furthermore, as described in Chapter 3, in MS-based comparative studies, the general form for computing fold changes incorporating normalization is:

$$(i_{jx}/S_{jx}) \,/\, (i_{jy}/S_{jy}) \tag{4.3}$$

where i_{jx} = intensity of ion j in run x, i_{jy}= intensity of ion j in run y, and S_{jx} and S_{jy} are scaling factors computed by a global function for runs x and y respectively. Interestingly, if I define a new global scaling function,

$$S_{jx} = \sum_{j=1}^{n_x} i_{jx} \tag{4.4}$$

where n_x is the number of ionized peptides in run x, then the global normalization formula becomes,

$$\left(i_{jx} / \sum_{j=1}^{n_x} i_{jx} \right) / \left(i_{jy} / \sum_{j=1}^{n_y} i_{jy} \right) \tag{4.5}$$

which is similar to the proportionality paradigm's relative proportions formula,

$$\left(i_{ja} / \sum_{j=1}^{n_a} i_{ja} \right) / \left(i_{jb} / \sum_{j=1}^{n_b} i_{jb} \right) \tag{4.6}$$

The only difference is that variables x and y represent MS runs and variables a and b represent samples. Despite these similarities, however, the answer is *"no"*, it is not just another normalization method. In fact, the proportionality paradigm differs from current normalization methods in four ways.

First, the intent of the proportionality paradigm and intent of normalization differ. In statistics, normalization can mean adjusting the values measured on different scales to a nominal value (see Chapter 3). It can also mean making two distributions similar, or as in quantile normalization, making the same 1/4 of two distributions similar. In microarray studies, and subsequently adapted for their own use by the mass spectrometry community, normalization attempts to make measurement distributions comparable by removing systematic bias [117]. The intent of the proportionality paradigm as applied to comparative proteomics is not to remove systematic bias, but to measure a protein's(or peptide's) proportion within a sample, or, when measured by mass spectrometry, to measure a peptide signal's intensity proportion within a run. The fortunate side effect of measuring in this manner is that systematic bias is ignored.

Second, the assumptions for sample input are different for the proportionality paradigm and normalization. Normalization in mass spectrometry tries to remove systematic measurement bias; thus the implication is that the amount of sample analyzed in multiple mass spectrometry runs is the same [117]. This is in contrast to the proportionality paradigm where the amount of sample measured is irrelevant [125].

Third, the input requirements the proportionality paradigm and normalization are different. The proportionality paradigm reports a peptide signal's intensity in relation to other peptide signal intensities within a run. This differs from normalization where scaling factors can be independent of peptide signal intensities, such as using ion current.

Fourth, and finally, the proportionality paradigm pre-dates the advent of MS-based comparative proteomics studies. As described in Chapter 3, for decades, geologists have used compositional data analysis to analyze the mineral content in rocks [172]. Current normalization methods for proteomics, and in particular, MS-based comparative proteomics, are less than a decade old.

The second question commonly asked is, *"Is measuring relative proportions rather than relative abundances really a paradigm shift for iLFRQ via HPLC-ESI-MS/MS?"* I posit the answer is *"yes"*. First, according to Thomas Kuhn in his 1962 book, *The Structure of Scientific Revolutions*, [19], a paradigm shift is "...a change in the basic assumptions, or paradigms, within the ruling theory of science..." and "...successive transition from one paradigm to another via revolution is the usual developmental pattern of mature science." The application of the proportionality paradigm in comparative proteomics parallels Kuhn's statement because it requires a researcher to change their perspective from *"one"* to *"many"*. (The *"one"* and *"many"* is a concept that transcends scientific disciplines. For example, it is a fundamental concept in database theory and data modeling [173].) Using the relative abundance paradigm, a researcher analyzes a peptide's differential abundance across samples in isolation. However, using the proportionality paradigm, when a researcher analyzes a peptides' differential proportions across samples, they inherently analyze the sample as a whole. This is because a change in a single peptide's abundance within a sample affects other peptide's proportion within the sample. Furthermore, analyzing the sample as whole, that is, examining its composition, more closely mirrors cellular biology. In vivo, peptides and proteins do not exist in isolation; they form multi-protein complexes and interact with each other, as well as other biomolecules, for example, enzymes and metabolites, thus driving cellular processes [174]. In other words, a change in a single protein's (or peptide's) abundance affects other protein's (and peptide's) proportion's within a cell.

Third, in 1962, Kuhn also specified five criteria for warranting a paradigm shift.

- "Accurate - empirically adequate with experimentation and observation."

- "Consistent - internally consistent, but also externally consistent with other theories."
- "Simple - the simplest explanation, principally similar to Occam's Razor."
- "Broad Scope - a theory's consequences should extend beyond that which it was initially designed to explain."
- "Fruitful - a theory should disclose new phenomena or new relationships among phenomena."

In the following chapters, I provide evidence that using the proportionality paradigm instead of the relative abundance paradigm meets Kuhn's criteria for warranting a paradigm shift in iLFRQ via HPLC-ESI-MS/MS. The evidence is summarized in Chapter 8.

In sum, using the proportionality paradigm instead of the relative abundance paradigm represents a fundamental shift in how researcher's employ iLFRQ for HPLC-ESI-MS/MS. Furthermore, using the proportionality paradigm requires a change in perspective from *"one"* to *"many"*, which more closely mirrors the biological impact of peptidomic and proteomic compositional changes. In addition, a shift to a new paradigm is warranted; the proportionality paradigm meets Kuhn's six criteria for a new paradigm. Ultimately, armed with the proportionality paradigm, I expect that researchers will gain insights into the molecular machinery of biological activity and disease progression that would otherwise be missed.

Chapter 5

Proximity-based Intensity Normalization (PIN)

Computer science is no more about computers than
astronomy is about telescopes. - Attributed to Edsger Dijkstra

5.1 Introduction

Detecting biological variation via HPLC-ESI-MS/MS is problematic, primarily due to poor repeatability and reproducibility stemming from extraneous variability. Unfortunately, as described in Chapter 3, HPLC-ESI-MS/MS is inherently less amenable to standardization than other types of technology used in other 'omics fields, for example DNA microarrays [111]). While using strategies from other 'omics technologies, for example, global normaliziation, technical replication, and statistical tests help [26], localized variability from transient stochastic during HPLC-ESI-MS/MS analyses continue to interfere with accurately detecting biological variation. To mitigate this localized variability, I developed a new normalization method, Proximity-based Intensity Normalizaiton (PIN). It mitigates systematic bias and complex variability while retaining biological variation.

The remainder of this chapter is organized as follows. Section 5.2 motivates PIN using an example to illustrate the impact of variance and an example from a CPTAC study to illustrate the impact of complex variability. Section 5.3 presents PIN, using a simplified example. Section 5.4 discusses assumptions and limitations for PIN.

5.2 Motivation

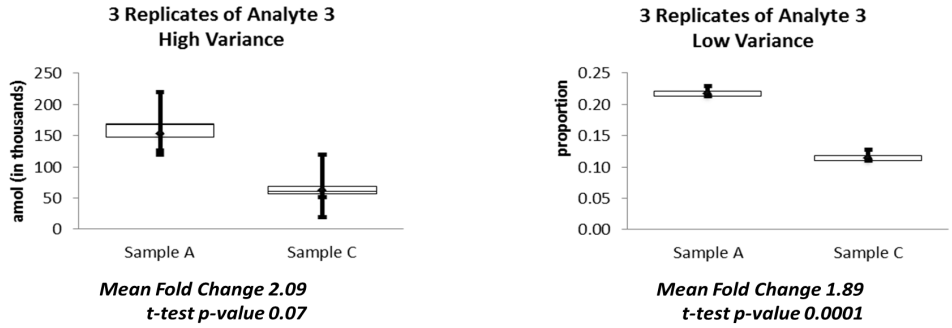
Statistical tests, such as t-test and ANOVA, are important tools for revealing biological variation in replicate analyses. Unfortunately, statistical tests are sensitive to variance, which in comparative proteomic workflows employing HPLC-ESI-MS/MS, stems from biological variation and extraneous variablity (systematic bias and complex variability). In the following subsections, I illustrate the impact of variance and complex variability on accurately revealing biological variation.

5.2.1 The Impact of Variance

Statistical tests for detecting biological variation are sensitive to variance. To illustrate the impact of variance, consider Figure 5.1 where Sample A and Sample C are each analyzed thrice via HPLC-ESI-MS/MS. The graph in Figure 5.1a shows Peptide 3's fold change between Sample A and Sample C as not statistically significant due to high variance, despite exceeding the de facto fold change threshold (2) the error bars overlap. Figure 5.1b shows Peptide 3's fold change as statistically significant due to low variance, despite not meeting the fold change threshold, the error bars do not overlap. Thus, minimizing variance, that is, mitigating extraneous variability, allows detection of biological variation not by detecting a peptides's fold change exceeding some numerical threshold, but instead by detecting statistically significant differences in measured intensities [109].

5.2.2 Complex Variability

In HPLC-ESI-MS/MS extraneous variability has two components, systematic bias and complex variability. In large scale comparative studies, researchers normalize to reduce external variability while retaining biological variation. Global normalization methods work well mitigate systematic bias. But, even when applying the proportionality



(a) High Variance

(b) Low Variance

Figure 5.1: The impact of variance

paradigm, the global functions cannot capture and mitigate temporally localized, complex variability [175]. As described in Chapter 3, complex variability during an HPLC-ESI-MS/MS runs seems inevitable, even when researchers follow strict protocols.

Using the CPTAC Study 6 dataset generated by instrument aliased LTQ-Orbitrap@65P, Rudnick, et al., found irregularities in the second (of three) replicate's analysis due to electrospray instability [176]. As shown in Figure 5.2, the chromatogram has a distinctive tooth pattern indicative of electrospray instability. However, as described in Chapter 3, chromatograms contain signal from noise as well as signal from peptides. What happens when only the peptide signals are plotted? Figure contains a plot of XC. It has a distinctive trough during the same time period as the electrospray instability. Unfortunately, applying a global normalization method such as median scale fails to mitigate the complex variability (See Figure 5.2). Furthermore, global normalization has unintended consequences and adversely affect regions where no complex variability exists: two regions of the XC now have more extraneous variability than before normalization, and can disguise true biological variation.

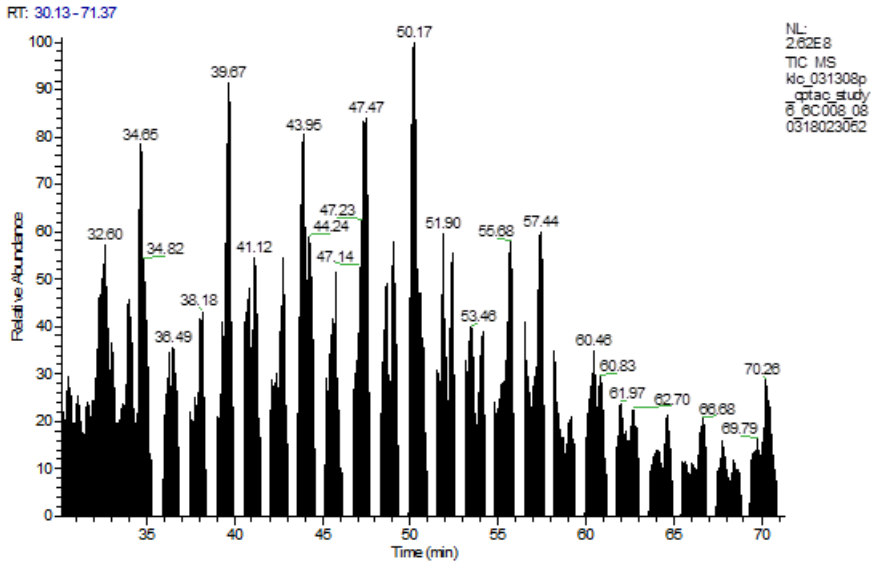
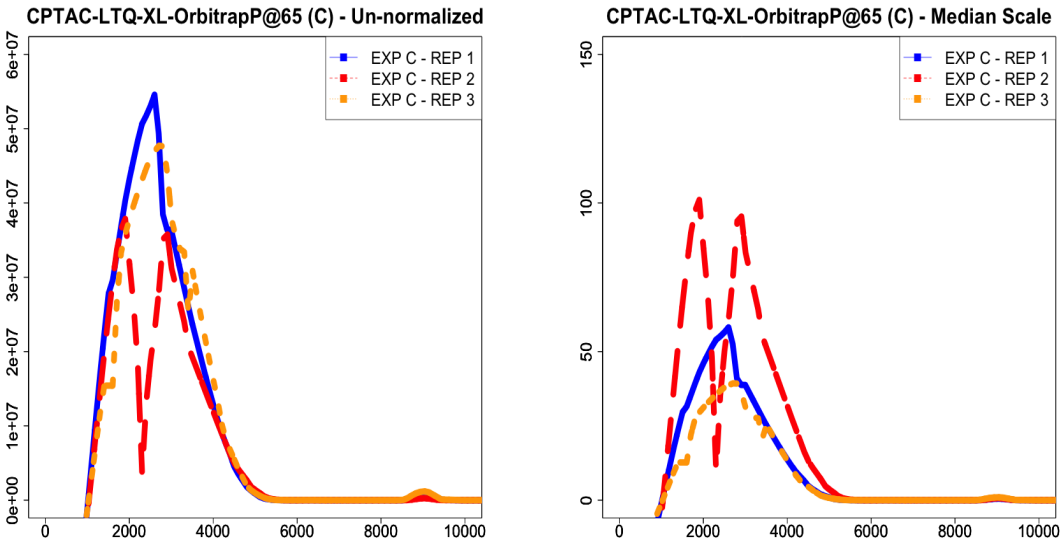


Figure 5.2: A portion of the chromatogram of the National Institutes of Health's Clinical Proteomic Tumor Analysis Consortium (CPTAC) Study 6C, Replicate 2. This is a text book example of complex variability. The distinctive saw tooth pattern is indicative of electrospray ionization inefficiency.



(a) Complex Variability

(b) Complex Variability - Median Scale

Figure 5.3: Extracted peptide signal chromatograms (XCs) from three technical HPLC-ESI-MS/MS archived replicates from the National Institutes of Health's Clinical Proteomic Tumor Analysis Consortium (CPTAC) Study 6C show that even well controlled HPLC-ESI-MS/MS experiments are vulnerable to complex variability. a) Sample 6C replicate 2 (dashed line) contains a distinctive trough indicating complex variability. As shown in the chromatogram in Figure 5.2, the trough results from intermittent electrospray ionization efficiency. b) Chromatograms normalized by median scale result in extracted chromatograms where the distinctive trough remains. XCs for replicates 1 and 3 (blue and orange lines) only slightly diverge, but replicate 2's complex variability is exaggerated after normalization by median scale.

5.3 Proximity-based Intensity Normalization (PIN)

Fortunately, while reviewing the XCs, I observed that complex variability similarly affects measured a peptide's measured intensities within close proximity (a temporal window or neighborhood). Furthermore, based on retention time alignment's premise that LC columns elute peptides based on hydrophobicity and thus when measured via MS multiple times, should at the same relative retention time, I reasoned that peptide signals should form similar neighborhoods across LC-MS runs. Finally, I reasoned that because then at the neighborhood level, complex variability becomes systematic bias. If this reasoning holds, it then follows that a proximal normalization method that will mitigate both systematic bias and complex variability.

In response, I developed a new algorithm named proximity-based intensity normalization (PIN). PIN normalizes a peptide signal's intensity using the sum of its neighboring peptide signals' intensities. It does so by first constructing its temporal neighborhood and then computing its relative proportion within that neighborhood. Here, the temporal neighborhood is bounded by a minimum and maximum retention time.

$$n_{r_{min}}, \ldots, n_{r_{max}} \in N \tag{5.1}$$

where N is the set of neighboring peptide signals, n is a peptide signal, r_{min} is the index of the peptide signal corresponding to the neighborhood's lower temporal boundary, r_{max} is the index of the peptide signal corresponding to the neighborhood's upper temporal boundary. With the neighborhood defined, PIN's normalization formula is

$$\left(n_{ja} / \sum_{i=r_{min}a}^{r_{max}a} n_{ia} \right) \tag{5.2}$$

where n_{ja} = intensity of peptide signal j in run a. Then, the relative proportion formula

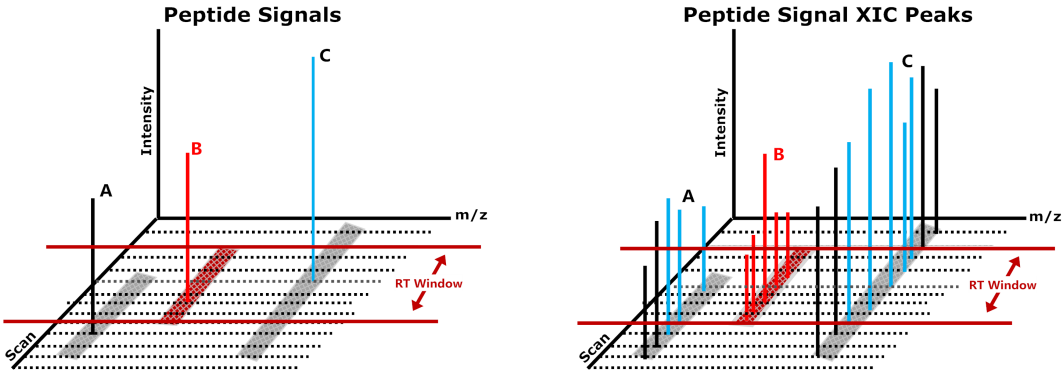
is

$$\left(n_{ja} / \sum_{i=r_{min}a}^{r_{max}b} n_{ia} \right) \Big/ \left(n_{jb} / \sum_{i=r_{min}b}^{r_{max}b} n_{ib} \right) \tag{5.3}$$

where n_{ja} = intensity of peptide signal j in run a, and n_{jb}= intensity of peptide j in run b.

Unfortunately, normalizing peptide signals by its neighboring peptide signals is not as straightforward as it seems. The complexity arises because peptide signals are constructed from XICs, and those XICs can straddle temporal window boundaries. Recall from Chapter 3, Step 2, that a peptide signal is constructed from an XIC which has three dimensions, m/z, intensity, and retention time. The peptide signal's m/z and retention time are adopted from the XIC apex, but the peptide signal's intensity is the sum of its parent XIC intensities along its retention time axis. In Figure 5.4a, the three vertical lines represent peptide signals A, B, and C. The shaded areas represent the width of each peptide signal's original XIC. The two horizontal lines represent the lower and upper boundaries for populating the B's neighborhood. In Figure 5.4b, the numerous vertical lines represent the three peptide signals' XIC peaks. As with the peptide signals, the shaded area represents the lower and upper boundaries of the XICs. Note that the two horizontal lines representing the upper and lower boundaries for B's XIC neighborhood coincide with B's shaded area's boundaries. This is because the neighborhood is defined by the width of the XIC.

Note in Figure 5.4a, that peptide signal C has a retention time near, but within, the temporal window boundary for B. Therefore, C qualifies for inclusion in B's neighborhood. However, C's XIC straddles the temporal window boundary. Thus, part of that peptide signal's intensity was measured at a time point outside the temporal window boundary. Of course, the converse is also true. For example, peptide signal C has a retention time near, but outside, the temporal window thus disqualifying it from the neighborhood, but part of that peptide signal's intensity was measured at a time point within the temporal window boundaries.



(a) Neighborhood populated with peptide signals

(b) Neighborhood populated with peptide signals XIC peaks

Figure 5.4: Peptide signal neighborhoods.

$$\left(d_{ja} / \sum_{i=r_{min}a}^{r_{max}a} d_{ia} \right) \tag{5.4}$$

where d_{ja} = intensity of deisotoped peak j in run a. Furthermore, the relative proportion formula is

$$\left(d_{ja} / \sum_{i=r_{min}b}^{r_{max}b} d_{ia} \right) \; / \left(d_{jb} / \sum_{i=r_{min}b}^{r_{max}b} d_{ib} \right) \tag{5.5}$$

where d_{ja} = intensity of deisotoped peak j in run a, and d_{ja}= intensity of deisotoped peak$j$ in run b. Using these equation for normalizing a peptide signal's intensity solves the straddling peptide signal problem.

5.4 Discussion

The question remains, *"What is the optimal temporal window size?"* I conducted initial experiments using an in-house pulled capillary tip of 100 Îijm inner diameter, packed to 13 cm with Magic C18AQ 5-μm, 200 Å pore particles (Michrom Bioresources), salivary endogenous peptides, and a sixty minute HPLC-ESI-MS/MS run a LTQ Orbitrap XL mass spectrometer (ThermoFisher Scientific). Initial results indicate that a for a

static window, a size of five to seven minutes produced good results. A window size greater than ten minutes or shorter than three minutes tended to degrade the results.

Logically, the optimal window size will vary with HPLC column type, sample complexity, and duration of the HPLC-ESI-MS/MS run. Therefore, ideally, the window size would be dynamically computed based on run characteristics. One way to compute a dynamic window with boundaries determined by the width of the peptides signal's XIC. Not surprisingly, when I tested XIC width dynamic windows, it outperformed static windows in reducing extraneous variability. Thus, I concluded that the answer to the question concerning optimal window size is, *"It depends."*

Of course, any analysis must be done with care. First, PIN relies on high resolution instrumentation with concomitant mass accuracy to extract and quantify relevant chromatographic information. Fortunately, such systems are routinely available [177]. Second, as with other methods, the peptide elution order for analyzed samples must be similar; PIN requires similar order to form similar neighborhoods. This means samples analyzed using different types of chromatographic systems may not be easily compared. Third, PIN tends to compress the dynamic range of fold changes. Thus, to determine biological variation, statistical significance, (not numerical value) of a peptide's fold change should be used.

Given that the relative abundance and global normalization analyses fail in the face of extraneous variability, I expect proportionality and proximal normalization to change the way researchers analyze HPLC-ESI-MS/MS experimental data. Furthermore, many biomolecules, when analyzed by HPLC-ESI-MS/MS, share characteristics with peptides. Thus, I expect the proportionality paradigm and PIN will be widely applicable to many 'omics fields, for example, lipidomics, glycomics, and metabolomics. The upshot will, I expect, be reproducibility and repeatability improved, and otherwise falsely reported or missed, statistically significant biological variation discovered.

Chapter 6

RIPPER: An iLFRQ Framework

An algorithm must be seen to be believed. - Donald Knuth

6.1 Introduction

This chapter describes RIPPER, a new software framework implementing Proximity-based Intensity Normalization (PIN). RIPPER reports normalized peptide signal intensities rather than protein quantities. These peptide signal intensities can serve as input to statistical tests, such as t-test, for determining statistically significant differences. Finally, these peptide signals can be matched to peptide amino acid sequences via an interface to third-party software.

The remainder of this chapter is organized as follows. Section 6.2 describes the motivation behind RIPPER. Section 6.3 describes RIPPER's software architecture. Section 6.4 describes RIPPER's graphical user interface. Section 6.5 describes RIPPER's processing steps. Section 6.6 describes RIPPER's output and optional intermediate results. Section 6.7 discusses RIPPER's advantages over existing iLFRQ frameworks, its limitations, and future work.

6.2 Motivation

RIPPER was born out of frustration with current iLFRQ software frameworks. First, while analyzing data from salivary endogenous peptide experiments, I found that current iLFRQ software frameworks report normalized intensities at the protein level. Protein level intensities are of little use to a researcher studying endogenous peptides. (Chapter 7, which follows this chapter, described these experiments in detail.) Second, using SQL, as a proof of concept, I implemented a labor intensive version of PIN (data not shown). However, to provide researcher's easy access to PIN, I wanted to automate it by integrating it into and existing framework. To do so, I experimented with converting the SQL statements to source code and retrofitting it into several open source applications. After several attempts, I found integrating PIN into an existing framework would have required an infeasible amount of modification to existing code. Frustrated, I created a new software framework for iLFRQ, which I named RIPPER.

6.3 Software Architecture

RIPPER was designed using software engineering principles [178] as a foundation for planned success.[1] While software engineering principles provide guidelines for almost every aspect of an application's development (design, documentation, coding, testing, distribution, etc.), four of them served as guiding principles for RIPPER's development.

- Cross platform: I chose to write RIPPER in Java because compiled Java byte code runs within a Java Virtual Machine (JVM), available on most operating system platforms (Windows, Linux, OSX, etc.).

- Extensible: I designed RIPPER using object oriented principles, easing algorithm modification and new algorithm integration.

[1] "Planning for Success" is a memorable *Carlis-ism.*

- Parameter driven: I designed a graphical user interface that allows user specified parameters to control RIPPER's operation.

- Freely available: Although the University of Minnesota Office for Technology Commercialization (OTC) requires licensing fees for commercial users the academic license is free. RIPPER can be downloaded from https://z.um.edu/ripper.

6.4 Graphical User Interface

While users can launch RIPPER by executing a command with a long string of parameters in a terminal, RIPPER's potential users (life science researchers) tend to be more comfortable interacting with software via a graphical user interface (GUI). Therefore, planning for success, that is, a growing user group, means that RIPPER needs a GUI, one that allows a user to select input, files output destinations, and parameter settings. Unfortunately, my experience developing GUIs is limited. Therefore, to expedite RIPPER's GUI development, I turned to John Chilton from the Minnesota Supercomputing Institute. Together, we developed a straightforward interface that allows users to select input files via the Add File(s) button, specify the output directory for results, and provide an identifier (RunGroup) for the analysis (see Figure 6.1). Optionally, users can select Scaffold extract files via the Add ID File(s) button to assign peptide sequence identifications to the generated results. Furthermore, the advanced options panel allows users to change default parameters.

RIPPER's distribution contains a ripper.pin.properties which stores default advanced parameters. However, if a user changes any of the advanced parameters, RIPPER saves them in a copy of the ripper.pin.properties file located in the user's home directory. Then, the next time a user launches RIPPER, the user's saved advance parameters will override the default parameters.

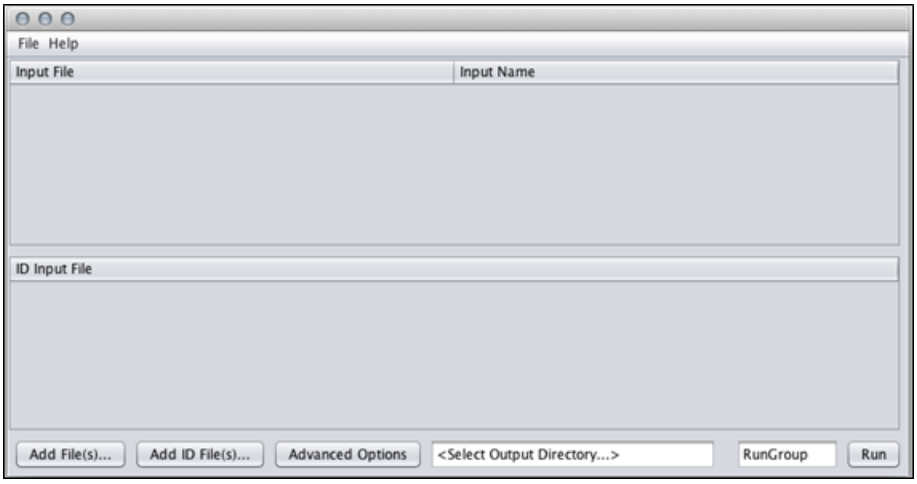


Figure 6.1: RIPPER's Graphical User Interface (GUI)

6.5 RIPPER's Processing Steps

RIPPER takes in a set of mzXML files (and optionally, third party software peptide and protein identifications), processes them, and outputs peptide signal reports with un-normalized and normalized intensities. As shown in Figure 6.2, RIPPER's processing can be logically divided into steps. The vertical boxes with dashed outlines represent mzXML files processed in a single RIPPER analysis. The horizontal boxes with solid outlines represent steps within RIPPER. Where steps cross dashed boxes, RIPPER processes combined data extracted from mzXML files. The following list provides an overview of RIPPER's 6 processing steps.

1. Preliminary Data Processing - This step is divided into two sub-steps.

   (a) Convert mzXML File - Converts a mzXML file to internal data structures using an external open source library, JRAP from the Institute From Systems Biology (ISB).

   (b) Construct Objects - Constructs RIPPER *run*, *scan*, and *peak* internal objects from JRAP's internal data structures.

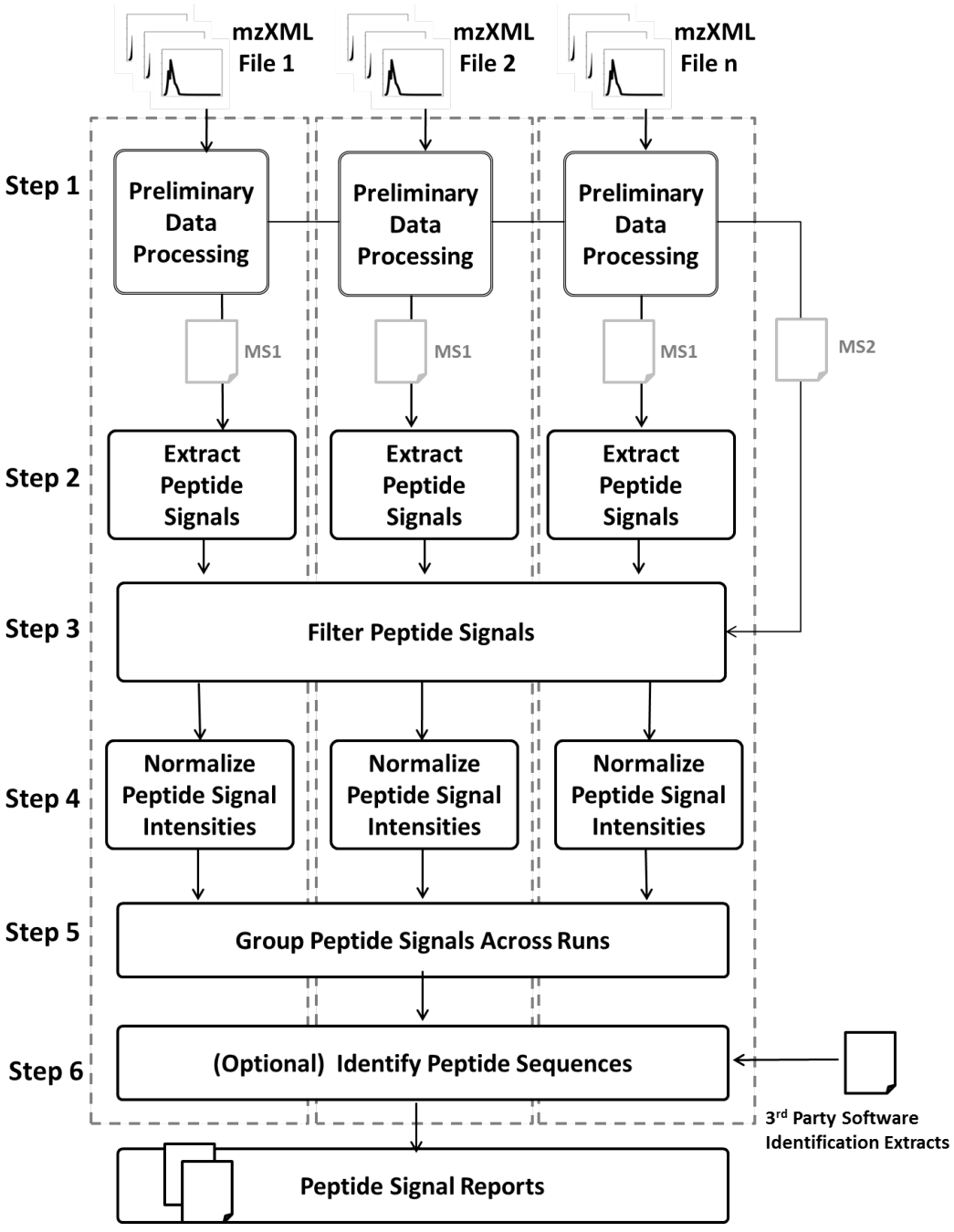


Figure 6.2: RIPPER's processing steps.

2. Extract Peptide Signals - This step is divided into five sub-steps.

   (a) Compute S/N Threshold - Computes a S/N threshold for each MS^1 *scan* using a median intensity algorithm adapted from msInspect [75].

   (b) Detect and Validate Peak Envelopes - Within each MS^1 *scan*, this step selects isotopic peak envelopes, that is, peak clusters with a characteristic inverse charge m/z spacing and a pseudo Poisson [79] intensity distribution.

   (c) Deisotope Isotopic Peak Envelopes - For each validated peak envelope, constructs a deisotoped peak object with the following attributes: charge = the envelope's inverse m/z spacing, m/z = envelope's monoisotopic peak's m/z value, and intensity = envelope's summed peak intensity.

   (d) Extract Ion Chromatograms - Constructs XICs by selecting clusters of deisotoped peaks having the same charge and similar m/z values across neighboring scan's, that is, along the retention time dimension.

   (e) Construct peptide signals - Constructs a peptide signal object from each extracted ion chromatogram with the following attributes: m/z value and charge = XIC's apex (peak with the largest intensity) m/z value and charge, intensity = sum of XIC's constituent peak's intensities.

3. Filter Peptide Signals - Removes peptide signals that do not meet minimum quality requirements, for example, a minimum number of corresponding MS^2 scans.

4. Normalize Peptide Signal Intensities - Scales each peptide signal's intensity with PIN.

5. Group Peptide Signals Across Runs - Groups peptide signals, across runs, that have the same charge and similar m/z values within a relative retention time window. This step is in lieu of retention time alignment.

6. Identify Peptide Sequences - Matches results to sequences identified by third party software. This step is optional.

The following sections describe RIPPER'S processing steps in detail. Each step's algorithms are described using pseudocode accompanied by its English translation. (Periodically, figures will augment these descriptions.) What separates pseudocode from real programs is that pseudocode employs expressive methods that specify an algorithm in a clear and concise manner [179]. Sometimes the clearest method is shorthand text or a description of an external function. However, most often the clearest method is using a defined vocabulary.

RIPPER's Vocabulary

Each HPLC-ESI-MS/MS run generates an mzXML file, which has three fundamental elements, denoted by lower case letters:

$$\begin{aligned} r &\leftarrow mzXML\ file\ (run) \\ s &\leftarrow scan \\ p &\leftarrow peak \end{aligned}$$

Within an mzXML file, these fundamental elements are nested, that is, a run has multiple scans, and scans have multiple peaks; the elements can be defined as sets, denoted by upper case letters:

$$\begin{aligned} R &= \{r_1, r_2, \ldots,\ r_m\},\ where\ m = \ number\ of\ runs \\ S &= \{s_{r1}, s_{r2}, \ldots,\ s_{rn}\},\ where\ n = number\ of\ scans\ in\ r,\ and\ r\ \in R \\ P &= \{p_{rs1}, p_{rs2}, \ldots,\ p_{rsq}\},\ where\ q = number\ of\ peaks\ in\ s,\ and\ s\ \in\ S \end{aligned}$$

When translated into pseudocode, elements become objects, denoted by lower case greek letters, with the following mapping:

$$\begin{aligned} \lambda &\leftarrow r \leftarrow mzXML\ file\ (run) \\ \omega &\leftarrow s \leftarrow scan \\ \delta &\leftarrow p \leftarrow peak \end{aligned}$$

Sets become vectors, denoted by upper case greek letters, with the following mapping:

$$\begin{aligned} \Lambda &\leftarrow (\lambda_1, \lambda_2, \ldots, \lambda_m),\ where\ m\ =\ number\ of\ r\ in\ R \\ \Omega &\leftarrow (\omega_1, \omega_2, \ldots, \omega_n),\ where\ n\ =\ number\ of\ s\ in\ S \\ \Delta &\leftarrow (\delta_1, \delta_2, \ldots, \delta_q),\ where\ q\ =\ number\ of\ p\ in\ P \end{aligned}$$

Within RIPPER's vocabulary, temporary objects are denoted by primed lower case Greek letters, and temporary vectors are denoted by primed uppercase Greek letters. Variables are denoted by lower case roman letters, and functions are denoted by the small caps font. Finally, as shown in Figure 6.3, RIPPER's vocabulary is also mapped onto to a cartoon depiction of a single mzXML file.

6.5.1 Step 1: Preliminary Data Processing

RIPPER's first processing step is preliminary data processing. This step takes in a mzXML file name and outputs run, scan, and peak objects amenable for downstream processing. RIPPER divides the preliminary data processing step into two sub-steps.

6.5.1.1 Sub-step 1.a: Convert mzXML File

The first sub-step converts data from mzXML files to internal data structures. To do so, RIPPER employs algorithms from the Java Random Access Parser (JRAP). I chose JRAP because it is object oriented and written in Java, thus easily integrated into

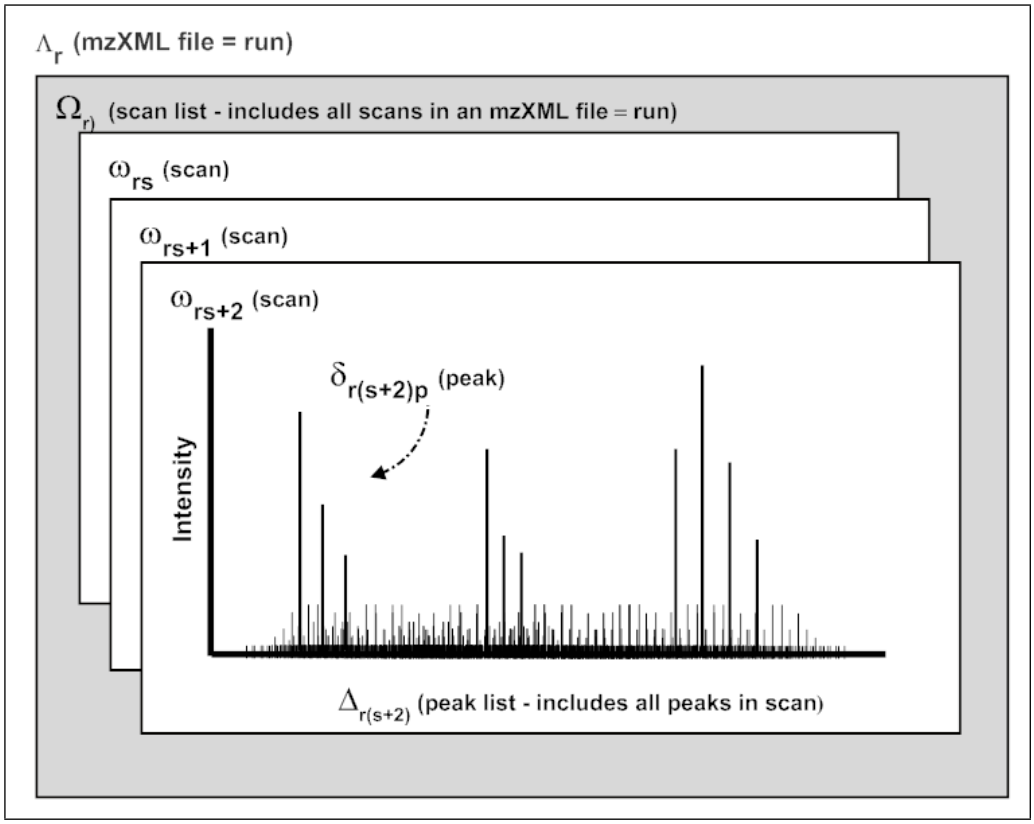


Figure 6.3: Cartoon of a single mzXML file (run) with RIPPER vocabulary mapped onto it.

RIPPER. JRAP is a Streaming API[2] for XML (StAX) parser. StAX combines the Document Object Model parser, which allows for random access, and an event based parser, which requires a smaller memory footprint and fewer CPU cycles. This allows users to use RIPPER on a relatively small computer, for example, a laptop with as little as 4GB of memory.

RIPPER, using JRAP, converts a mzXML file using the Convert mzXML File algorithm. This algorithm, described in detail in Algorithm 1:Convert mzXML File listing, takes in a mzXML file name, and after converting the data using JRAP, it outputs a

[2]API stands for Application Programming Interface

mzXMLfile object.

Algorithm 1 Convert mzXML File

Input mzXML file name
Output An mzXMLFile object

```
1: function convertMzXML(mzXMLFileName)
2:     mzXMLFile ← JRAP.getMZXMLFILE(mzXMLFileName)
3:     return mzXMLFile
4: end function
```

The Convert mzXML algorithm is straightforward. In line 2, the algorithm calls JRAP's getMZXMLFile method. Within this method, using StAX, JRAP parses the mzXML file and constructs a mzXMLfile object, which contains header information and a vector of JRAP scan objects. Within each scan object, JRAP constructs a scan header and two one-dimensional arrays each containing decompressed base64 peak data. The first array contains m/z values and the second array contains intensity values. In line 3, the algorithm returns the mzXMLfile object to RIPPER.

6.5.1.2 Sub-step 1.b: Construct Objects

The second sub-step within the preliminary data preprocessing step converts the JRAP mzXMLFile (output in Sub-step 1.a in Section 6.5.1.1)to RIPPER's version of run, scan, and peak objects. Unfortunately, RIPPER cannot use JRAP's mzXMLFile object nor the scan objects contained therein. These objects lack attributes needed by RIPPER in downstream processing. First, in a large scale comparative study, groups of runs must be uniquely identified. However, the mzXMLFile object does not provide an attribute for a run group identifier. To address this, RIPPER constructs a new run object from the mzXMLFile object's attributes, adding the user supplied run group identifier (see Figure 6.1). Second, JRAP's scan object stores peak data in two one-dimensional arrays. The first array contains m/z values and the second array contains a list of intensities. Each peak is represented as a m/z value – intensity pair identified by the arrays' indexes.

This representation does not allow additional information to be stored about each peak. To address this, RIPPER constructs a new scan object from a JRAP scan object's attributes, adding additional attributes, for example, vectors of peak objects. Within each scan, RIPPER extracts m/z value – intensity pairs from the two arrays, constructs a new peak object, and adds it to the peak list implemented as a vector. After processing all scans, RIPPER produces a run object and two vectors containing scan objects, one containing MS_1 scan objects and one containing MS_2 scan objects.

6.5.2 Step 2: Extract Peptide Signals

RIPPER's second step extracts peptide signals. This step takes in a vector of MS^1 scan objects (output from Sub-step 1.b in Section 6.5.1.2) and outputs two vectors, one containing peptide signals and the other containing valid XIC peaks. RIPPER extracts peptide signals using a series of sub-steps.

6.5.2.1 Sub-step 2.a: Compute S/N Thresholds

The first sub-step computes each MS^1 scan's S/N threshold based on the Median Intensity algorithm adapted from msInspect [75]. Briefly, the Median Intensity algorithm takes in a vector of MS^1 scans (output from Sub-step 1.b in Section 6.5.1.2) and a user specified scaling factor. After computing and updating each MS^1 scan's S/N threshold, it returns the same vector of MS^1 scans to the caller.

The Algorithm 2 : Median Intensity listing describes the Compute S/N sub-step in detail using pseudocode. Additionally, Figure 6.4 depicts a single MS^1 scan with key elements labeled to match pseudocode notation.

Algorithm 2 Median Intensity

```
   Input ⇐  Ω : A vector of MS[1] scans
            f : User supplied scaling factor
   Output ⇒ Ω : A vector of MS[1] scans with their computed S/N thresholds
 1: function MEDIANINTENSITY(Ω, f)
 2:     for i ← 0, Ω.size -1 do                          ▷ process each scan
 3:         ω ← Ω_i                                      ▷ copy scan
 4:         sort ω.Δ such that δ_x.intensity < δ_{x+1}.intensity   ▷ sort peak list
 5:         x ← ω.Δ.size                                 ▷ size of peak list
 6:         if x mod 2 ≠ 0 then                          ▷ odd?
 7:             y ← ω.Δ_{x/2}.intensity
 8:         else                                         ▷ even?
 9:             y ← ω.Δ_{x/2+1}.intensity
10:         end if
11:         ω.snThresh ← Δ_y.intensity * f               ▷ compute snThrehold
12:         Ω_i ← ω                                      ▷ copy back
13:     end for
14:     return Ω
15: end function
```

The Median Intensity algorithm processes each MS[1] scan in the scan vector using a for loop (line 2). Within the for loop, it first assigns the scan to a new variable (line 3) and then sorts the scan's peak vector by intensity in ascending order (line 4). Then the algorithm computes its median index (lines 5- 10, and using the peak at that index, it computes the scan's S/N threshold by multiplying the peak intensity by the user supplied scaling factor (line 11). Finally, it updates the new scan variable's S/N threshold attribute (line 12) and replaces the original scan vector. After the for loop finishes, the algorithm returns the updated scan vector to the caller.

6.5.2.2 Sub-step 2.b: Detect and Validate Isotopic Peak Envelopes

The second sub-step detects and validates peaks isotopic envelopes. It does so within each scan'ÂŢs peak list using the Get Peak Envelopes algorithm. Many existing frameworks detect envelopes from raw chromatographic data. However, they do not meet RIPPER's needs because RIPPER takes in as input, mzXML files, not raw files. So, rather than modify an existing implementation, I rethought the implementation and

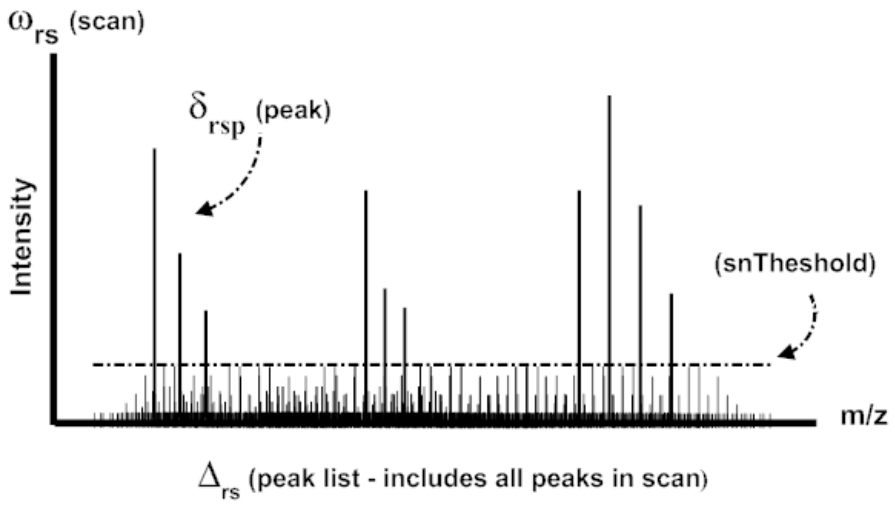


Figure 6.4: Cartoon of a single MS^1 scan with it's S/N threshold overlaid (denoted by the dashed line).

design.

In RIPPER, isotopic peak envelopes are two dimensional, that is, they have m/z and intensity dimensions. Therefore, when designing an isotopic peak detection algorithm, I had two options, 1) look for patterns in the intensity dimensions first and then the m/z dimension, or 2) look for patterns in the m/z and and then the intensity dimension. While many existing frameworks use the first option, I chose the second option for two reasons. First, looking for an intensity pattern requires a sophisticated, CPU intensive, algorithm. Second, by looking for m/z spacing first, the algorithm requires only the first peak in the envelope having its intensity above the scan'ÂŢs S/N threshold, thus allowing the detection of lower abundance peptides.

Once isotopic peak envelopes are detected, they must be validated. Similar to msInspect, RIPPER employs a Poisson pattern matching algorithm [75] and a Kullback-Leibler (KL) similarity measure [180]. As described in Chapter 3, this algorithm computes the theoretical Poisson distribution for candidate envelope's monoisotopic peak m/z and charge. Using the KL similarity measure's deviance scoring model, it ascertains

similarity between the theoretical distribution and the candidate envelope's distribution.

The detect and validate peak envelopes step employs two algorithms, Get Peak Envelopes and Validate as Poisson. They are described in detail using pseudocode in the Algorithm 3: Get Peak Envelopes (Part 1), Algorithm 3: Get Peak Envelopes (Part 2), and the Algorithm 5: Validate as Poisson listing. In addition, Figure 6.5 depicts a single MS^1 scan with three peak envelopes (shaded boxes), a vector of peak envelope (box with dotted outline), and a vector of used peaks (box with the dashed outline).

Algorithm 3 Get Peak Envelopes (Part 1)

Input $\Leftarrow$ ω : An MS^1 scan
Π : A vector of user supplied charges as integer
e : User supplied envelope peak spacing tolerance
m : User supplied minimum number of peaks in envelope
Output $\Rightarrow$ ω : An MS^1 scan

1: **function** GETPEAKENVELOPES(ω, Π, e, m)
2: $\Upsilon \leftarrow initialized$ ▷ peak envelopes vector
3: $a \leftarrow 0$ ▷ index for Υ
4: $\Phi \leftarrow initialized$ ▷ used peaks vector
5: $b \leftarrow 0$ ▷ index for Φ
6: $\Delta' \leftarrow \omega.\Delta$ ▷ temporary peak list vector
7: sort Δ' such that $\Delta'_x.mz < \Delta'_{x+1}.mz$ ▷ sort in ascending m/z order
8: sort Π such that $\Pi_x > \Pi_{x+1}$ ▷ sort in descending charge order
9: **for** $i \leftarrow 0, \Pi.size$ **do** ▷ process once for each charge
10: $y \leftarrow \frac{1}{\Pi_i}$ ▷ compute inverse charge spacing
11: **for** $j \leftarrow 0, \Delta'.size - 1$ **do** ▷ process peak list in ascending m/z order
12: $\delta' \leftarrow \Delta'_j$ ▷ copy peak to current peak
13: **if** δ' !Found in Φ **and** ▷ not already used?
$\delta'.intenisty > \omega.snThresh$ **then** ▷ above scan's S/N Threshhold
14: $\Lambda \leftarrow initialized$ ▷ new peak envelope vector
15: $\Lambda_0 \leftarrow \delta'$ ▷ first peak in peak envelope
16: $c \leftarrow 1$ ▷ index for Λ
17: **for** $k \leftarrow j + 1, \Delta'.size - 1$ **do** ▷ start at next peak in peak list
18: $\delta'' \leftarrow \Delta'_k$ ▷ copy peak to next peak
19: **if** δ'' !Found in Φ **then** ▷ not already used?
20: **if** $\delta''.mz > \Lambda_0.mz + d * 7$ **then** ▷ greater than max possible?
21: *break*; ▷ get out of loop
22: **end if**

continued on next page...

Algorithm 4 Get Peak Envelopes (Part 2)

```
    continued from previous page...
23:                     if δ''.mz > δ'.mz + y − e and          ▷ correct inverse charge spacing?
24:                         δ''.mz < δ'.mz + y + e then
25:                         Λ_c ← δ' ← δ''                      ▷ add peak to peak envelope
26:                         c ← c + 1
27:                         δ' ← δ''                            ▷ next peak becomes current peak
28:                     end if
29:                 end if
30:             end for
31:             if Λ.size > m and                               ▷ meets minimum criteria?
32:                 validPoisson(Λ) then                        ▷ is pseudo Poisson distribution?
33:                 Υ_b ← Λ                                     ▷ add envelope to peak envelop vector
34:                 b ← b + 1
35:                 for k ← 0, Λ.size − 1 do                    ▷ process peak envelope peaks
36:                     Φ_a ← Λ_k                               ▷ add peak to peaks used vector
37:                     a ← a + 1
38:                 end for
39:             end if
40:         end if
41:     end for
42: end for
43: ω.peakEnvelopes ← Υ                                         ▷ update scan's peak envelope vector
44: ω.usedPeaks ← Φ                                             ▷ update scan's used peaks
45: return ω
46: end function
```

The Get Peaks Envelopes algorithm operates on a single MS^1 scan, detecting and validating its peak envelopes. The first few lines perform some housekeeping tasks, including initializing temporary objects (lines 2 - 6) and sorting vectors (lines 7 - 8). The main outer loop (line 9) controls how many times the algorithm processes a peak list, once for each requested charge. The second loop (line 11) is an inner loop that controls the processing of the temporary peak list in increasing m/z order. As it processes each peak, it copies to current peak (line 12). Next, the algorithm checks if the peak has already been used in a peak envelope and compares its intensity to the S/N threshold (line 13). If it has not been used, the algorithm seeds the new peak envelope with the current peak (lines 14-16). The third loop (line 17) is an inner loop that looks ahead in the temporary peak list for the next peak (line 18) that is not already used (line 19). If the

next peak's m/z value is greater than the maximum possible m/z value that could be in the seeded peak's envelope, the loop finishes (lines 20 - 22). If the next peak has an inverse charge m/z spacing from the current peak's m/z charge (line 22), the algorithm adds the peak to the envelope (lines 24-25), and the next peak becomes the current peak (line 26). After the third loop finishes (line 29), the algorithm then checks the size of the peak envelope (line 30) and validates it as a pseudo Poisson distribution (line 31). If it passes these checks it adds the peak envelope to the scan's peak envelope and the used peak lists (lines 32 - 26 and 42 - 43). Finally, the algorithm returns the MS^1 scan to the caller.

Algorithm 5 Validate as Poisson

Input $\Leftarrow$ Λ : Peak envelope as a vector of peaks
Output $\Rightarrow$ Boolean : true / false

1: **function** VALIDPOISSON(Λ)
2: $H \leftarrow 1.0078250 - 5.485e-4$ ▷ mass of hydrogen - 1 electron
3: $z \leftarrow \frac{1}{\Delta_1.mz - \Delta_0.mz}$ ▷ compute charge from peak spacing
4: $m \leftarrow \Lambda_0.mz \times z - H \times z$ ▷ compute mass of seed peak
5: $n \leftarrow \Lambda.size - 1$ ▷ number of peaks in envelope
6: $\Lambda' \leftarrow \frac{\Lambda}{\sum_{i=1}^{n} \Lambda_i}$ ▷ normalize so sum of distribution = 1
7: $\Psi' \leftarrow$ POISSONDIST(m) ▷ get theoretical Poisson Distribution
8: **if** KL(Λ', Ψ') $< .4$ **then** ▷ score observed vs. theoretical distribution?
9: **return** *true* ▷ KL distance score low enough
10: **else** ▷
11: **return** *false* ▷ KL distance score not low high enough
12: **end if**
13: **end function**

The Validate as Poisson algorithm compares the detected peak envelope to a theoretical isotopic distribution, modeled as a pseudo Poisson distribution. The first few lines of the algorithm performs several tasks. First, the algorithm computes the mass of a single hydrogen atom, minus a single electron (line 2), the peak envelope charge based on peak spacing (line 3), and the mass of the monoisotopic (seed) peak (line 4). Next, it normalizes the peak envelopes intensities so that their sum is equal to one (line 6) so that it has

the same scale as the theoretical isotopic distribution returned from the poissonDistribution function (line 7), which is described in [75]. Finally, the algorithm uses the kL function (line 8) to score the match between the detected peak envelope and the theoretical isotopic distribution. If the function returns a score less than 0.4, the algorithm deems the peak envelope valid and the algorithm returns true to the caller (line 9). If the score is greater or equal to 0.4, the algorithm returns false to the caller (line 11). The value of 0.4 is chosen based msInspect's observation that a KL > 1.0 threshold is too conservative, meaning that it does not remove peak envelopes that visually do not match an isotopic distribution, and a KL < 0.1 threshold is too strict, meaning that real peak envelopes are excluded [75]. Furthermore, empirical evidence (results from varying the KL score threshold between 0.1 and 1.0 by 0.1 increments) showed that a KL < 0.4 threshold produced good results (based on visual inspection).

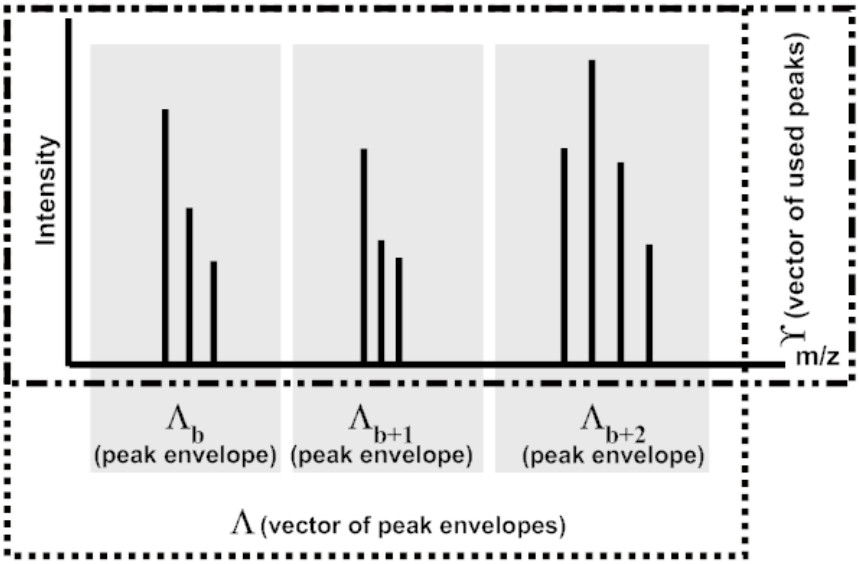


Figure 6.5: Cartoon of a single MS^1 scan with three peak envelopes (denoted by Λb, $\Lambda b + 1$, $\Lambda b + 2$ in the shaded boxes), a vector of peak envelopes (box with dotted outline, denoted by Λ), and a vector of used peaks (box with dashed outline, denoted by Υ).

6.5.2.3 Sub-step 2.c: Deisotope Isotopic Peak Envelopes

The third sub-step deisotopes validated isotopic peak envelopes. It takes in a MS^1 scan containing a list of peak envelopes, which were detected and validated in Sub-step 2.b (Section 6.5.2.2). For each peak envelope, the algorithm deisotopes it by aggregating it to single, deisotoped, peak. It does so by first selecting the envelope's monoisotopic peak (the lowest m/z value) and then summing the envelope's peak's intensities. The sub-step then constructs a new deisotoped peak as a monoisotopic peak m/z – summed peak intensity pair and adds it to the scan's list of deisotoped peaks. After all peak envelopes are processed, this sub-step outputs the MS^1, now with its list of deisotoped peaks.

The detect and validate peak envelopes step employs a single algorithm, Deisotope Peak Envelop, described in detail using using pseudocode in the Algorithm 6: Deisotope Peak Envelope listing. In addition, Figure 6.6 depicts a single MS^1 scan with three peak envelopes (shaded boxes), three deisotoped peaks (denoted by γ_b, γ_{b+1}, and γ_{b+2}), and a vector of deisotoped peaks (box with the dotted outline).

Algorithm 6 Deisotope Peak Envelope

```
    Input ⇐   ω : An MS^1 scan
    Output ⇒ ω : An MS^1 scan
 1: function DEISOTOPE(ω)
 2:     Γ ← initialized                          ▷ deisotoped peak vector
 3:     Λ′ ← ω.Λ                                 ▷ copy vector of peak envelopes
 4:     for i ← 0, Λ′.size − 1 do                ▷ process vector of peak envelopes
 5:         γ ← Λ′_i0                            ▷ deisotoped peak = first peak in envelope
 6:         x ← 0
 7:         for j ← 0, Λ′_i.size do              ▷ process peak envelope vector
 8:             x ← x + Λ′_i.intensity           ▷ sum intensities
 9:         end for
10:         γ.intensity ← x                      ▷ deisotoped peak intensity = sum intensities
11:         Γ_i ← γ                              ▷ add to vector of deisotoped peaks
12:     end for
13:     ω.deisotopedPeaks ← Γ                    ▷ update scans's deistoped peaks
14:     return ω
15: end function
```

The Deisotope Peak Envelope algorithm first initialized a new vector to contain deisotoped peaks (line2) and copies the MS^1 scan's peak envelopes vector (line3). The outer loop processes a scan's peak envelope vector (line 4). For each peak envelope, it selects the first peak and copies it to a new deisotoped peak (lines 5 - 6). Next, the algorithm sums the peak envelopes intensities (lines 7 - 10), updates the deisotoped peak's intensity with the summed intensity (line 11), and adds the deisotoped peak the scan's deisotoped peak vector (line 12). After it deisotopes all peak envelopes, the new deisotoped peak vector is added to the MS^1 scan (line 13). Finally, the MS^1 scan is returned to the caller (line 14).

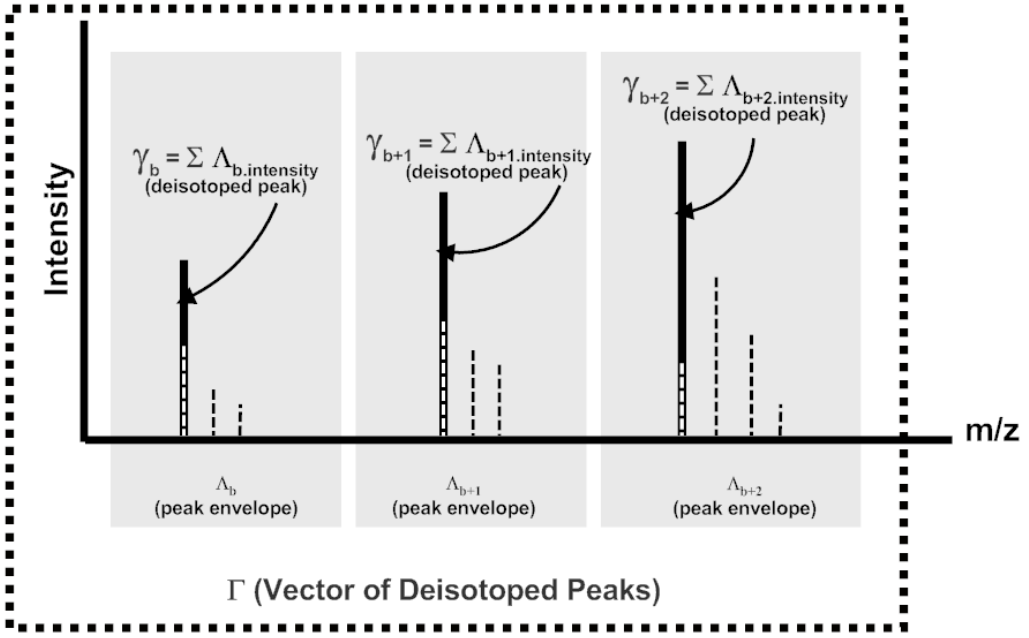


Figure 6.6: Cartoon of a single MS^1 scan with three deisotoped peaks (denoted by υ_x, υ_{x+1}, υ_{x+2} a vector of peak envelopes (box with dotted outline, denoted by Λ), and a vector of used peaks (box with dashed outline, denoted by Υ). A desisotoped peak is constructed from a peak envelope's (shaded boxes, denoted by (denoted by Λb, $\Lambda b + 1$, $\Lambda b + 2$ in the shaded boxes). The deisotoped peak's m/z value is that of the monoisotopic peak, the lowest m/z valued peak in the peak envelope and the intensity is the sum of the peak envelope's constituent peak intensities.

6.5.2.4 Sub-step 2.d: Extract Ion Chromatograms

The fourth sub-step extracts XICs. Because I chose to extract peak envelopes with m/z and intensity dimensions, the algorithm for extracting XICs had to assemble deisotoped peaks along the time dimension. This sub-step extracts XICs similar to one mentioned in the literature for SuperHirn[121]. It does so by combining deisotoped peaks from neighboring scans and then selecting deisotoped peaks with similar m/z within a short retention time window. Unlike SuperHirn, however, this sub-step additionally trims each XIC based on user supplied parameters. I chose to trim XICs because I had previously observed that XICs can have a long right tail. This tail seems to be erroneous, possibly caused by instrumentation. The sub-step produces a vector of XICs, and a vector of valid XIC deisotoped peaks.

To extract XICs, this sub-step employs two algorithms, Extract Ion Chromatogram and Trim XIC. The Extract Ion Chromatogram algorithm takes in a list of MS^1 scan as well as several user supplied parameters, and outputs two lists, one containing the trimmed XICs and the other containing all deisotoped peaks used in the trimmed XICs. To remove extraneous deisotoped peaks form XICs, the algorithm employs another algorithm, Trim XIC. It takes in a XIC and outputs a trimmed XIC. The two algorithms are described in detail using pseudocode in the Algorithm 7: Extract Ion Chromatogram (Part 1), Algorithm 8: Extract Ion Chromatogram (Part 2), and Algorithm 9: Trim XIC listings. In addition, Figure 6.7 depicts two trimmed XICs (the two shaded boxes, denoted by Ξ_x and Ξ_{x+1}). Note that in this figure, compared to previous figures, the graph is now three-dimensional and rotated. While the vertical axis remains the intensity dimension, the horizontal axis is now retention time, denoted by scan Ω. The m/z dimension is along the axis extending from the figure. The vertical lines within the shaded boxes are deisotoped peaks included in the trimmed XIC; vertical lines outside the shaded boxes are not included. The horizontal dashed lines delineate m/z ranges for each XIC and the vertical dashed lines delineate retention time boundaries for each

XIC. The incoming deisotoped peak list (denoted by Γ') contains all deisotoped peaks within a run.

Algorithm 7 Extract Ion Chromatograms

Input $\Leftarrow$ Ω : A vector MS^1 scans
e : User supplied m/z tolerance
m : User supplied minimum number of deisotoped peaks
r : User suppled maximum XIC m/z range
t : User suppled maximum XIC rt range
Output $\Rightarrow$ Ξ : A vector of XIC's

1: **function** GETXICS(Ω)
2: $\Xi \leftarrow$ *initialized* ▷ vector of XICs
3: $y \leftarrow 0$ ▷ index for vector of XICs
4: $\Gamma' \leftarrow$ GETALLDESISOTOPEDPEAKS(Ω) ▷ combine into 1 vector
5: sort Γ'such that$\Gamma'_x.mz < \Gamma'_{x+1}.mz$ ▷ sort ascending
6: **for** $i \leftarrow 0, \Gamma'.size - 1$ **do** ▷ process all deisotoped peaks
7: $\gamma' \leftarrow \Gamma'_i$ ▷ copy deisotoped peak
8: **if** γ'not in Φ **then** ▷ not already used?
9: $\Gamma'' \leftarrow$ *initialized* with γ' ▷ temporary peak vector
10: $x \leftarrow 1$ ▷ index for temporary peak vector
11: **for** $j \leftarrow i + 1, \Gamma'.size - 1$ **do** ▷ process next deisotoped peaks
12: $\gamma'' \leftarrow \Gamma'_j$ ▷ copy next deisotoped peaks
13: **if** Γ'_i not in Φ **then** ▷ not already used?
14: **if** $\gamma''.mz > \gamma'.mz + r$ **then** ▷ outside m/z range?
15: *break* ▷ get out of loop
16: **else**
17: $\Gamma''_x \leftarrow \gamma'$ ▷ add o temporary vector
18: **end if**
19: **end if**
20: **end for**
21: **end if**
22: **if** $\Gamma''.size > m$ **then** ▷ enough peaks in temporary vector?
23: sort Γ'' such that $\Gamma''_x.rt < \Gamma''_{x+1}.rt$ ▷ sort by ascending rt
24: $\Psi \leftarrow$ *initialized* ▷ XIC vector
25: $z \leftarrow 0$

continued on next page...

Algorithm 8 Extract Ion Chromatograms (Part 2)

```
    continued from previous page...
26:             for j ← 0, Γ''.size − 1 do                    ▷ process temporary peak vector
27:                 if Γ''_j.rt > Γ''_0.rt + t then            ▷ outside rt range?
28:                     if Ψ.size > m then                     ▷ enough peaks in XIC?
29:                         TRIM(Ψ, t)                         ▷ trim XIC
30:                         Ξ_y ← Ψ                            ▷ add to vector of XICs
31:                         y ← y + 1
32:                         Ψ ← initialized                    ▷ XIC vector
33:                         Ψ_0 ← Γ''_j                        ▷ add to XIC
34:                         z ← 1
35:                     end if
36:                 end if
37:                 Ψ_z ← Γ''_j                                ▷ not outside of range, add to XIC
38:                 z ← z + 1
39:             end for
40:             if Ψ.size > m then                             ▷ enough peaks in XIC?
41:                 TRIM(Ψ, t)                                 ▷ trim XIC
42:                 Ξ_y ← Ψ                                    ▷ add to vector of XICs
43:             end if
44:         end if
45:     end for
46:     return Ξ
47: end function
```

The first few lines of the algorithm perform some housekeeping tasks, including combining all deisotoped peaks from the incoming MS^1 scans and sorting it in ascending m/z order (lines 2 - 5). The main outer loop (line 6) processes the deisotoped peak vector peak list. Briefly, within this loop, the first part of the algorithm selects deisotoped peak clusters with similar m/z values (lines 7 - 21), storing them in a temporary vector, and the second part of the algorithm selects XICs as peak clusters within the temporary vector having similar retention time values (lines 22 - 37). In the first part of the algorithm, to select similar m/z valued deisotoped peak clusters, it copies the deisotoped peak (line 7), and then if the deisotoped peak has not already been used (line 8), it then initializes a temporary vector and adds to it the deisotoped peak (line 9). An inner loop looks ahead in the deisotoped peak vector, continuing to add deisotoped peaks to the temporary vector until the user supplied maximum m/z range has been met (lines 11 -

20). The resulting temporary deisotoped peak vector can contain more than one XIC. This is because two different peptides can have a similar m/z but separated by gaps in retention time. Therefore, after the first inner loop finishes, the algorithm's second part first checks the size of the temporary deisotoped peak vector, making sure that it meets minimum requirements (line 22). Then, it sorts the vector in ascending retention time order (line 23) and initializes a new XIC vector (line 24). The second inner loops process the vector (line 26), and if successive deisotoped peaks' retention times differ by more than the user supplied parameter (line 27), then the XIC vector contains a candidate XIC. The algorithm then checks XIC vector's size (line 28), trims the XIC vector using the TRIM function (line 29), and adds the trimmed XIC vector to the vector of XICs (line 30). Next, it initializes a new XIC vector, adds the deisotoped peak to the first position (line 32 - 34) and the loop continues, adding deisotoped peaks to the new XIC vector (line 37). Once the second inner loop finishes, the algorithm checks the final XIC vector's size (line 40), trims the XIC vector (line 41), and adds the trimmed vector to the vector of XICs (line 42).

Algorithm 9 Trim XIC

Input $\Leftarrow$ Ψ : An XIC peak list
t : User suppled maximum XIC rt range
Output $\Rightarrow$ Ψ : A trimmed XIC peak list

```
 1: function TRIM(Ψ, t)
 2:     Ψ' ← initialized                          ▷ trimmed XIC vector
 3:     x ← 0                                     ▷ index for trimmed XIC
 4:     a ← 0                                     ▷ apex intensity
 5:     for i ← 0, Ψ.size − 1 do                  ▷ process XIC vector - look for apex
 6:         if Ψ_i.intensity > a then             ▷ apex?
 7:             a ← Ψ_i.intensity                 ▷ update apex intensity
 8:             r ← Ψ_i.rt                        ▷ update apex rt
 9:         end if
10:     end for
11:     for i ← 0, Ψ.size − 1 do                  ▷ process XIC vector
12:         if Ψ_i.rt > r − t/2 and Ψ_i.rt < r + t/2 then   ▷ within rt range of apex?
13:             Ψ'_x ← Ψ_i                        ▷ add peak to trimmed XIC
14:         end if
15:     end for
16:     returnΨ'
17: end function
```

The first few lines of the algorithm perform some housekeeping tasks, including initializing a new trimmed XIC vector (lines 2 - 4). The first loop processes incoming XIC vector, looking for the XIC apex, which is defined as the deisotoped peak with the largest intensity (lines 5 - 10). The second loop processes the incoming XIC vector again, but this time, if a deisotoped peak's retention time is within the retention time window (defined as the apex retention time plus or minus one-half the user supplied maximum XIC retention time range), it adds it to the trimmed XIC vector (lines 11 - 15).

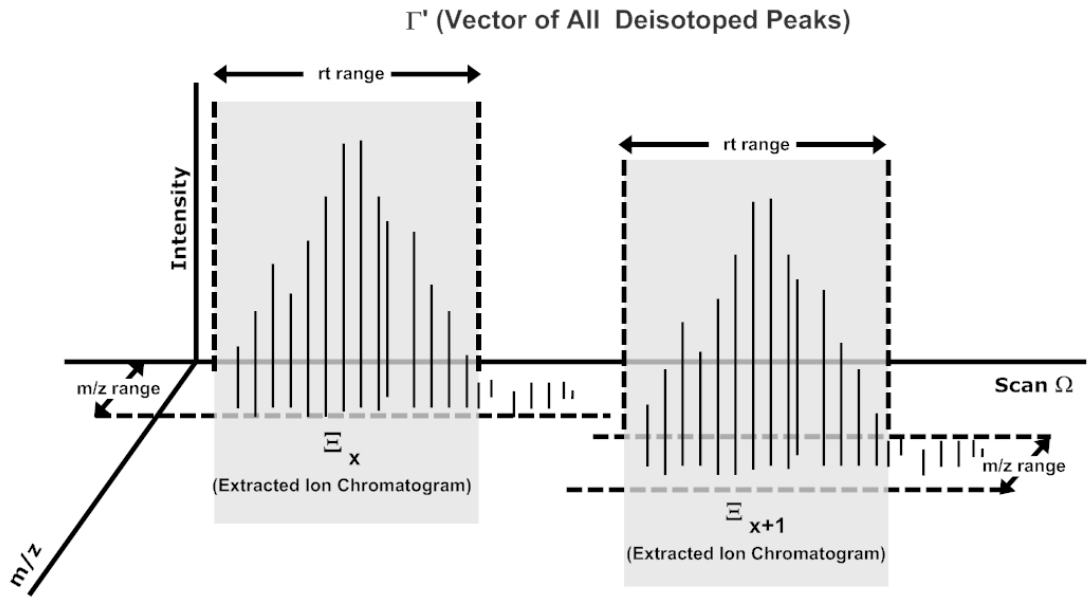


Figure 6.7: Cartoon of two trimmed XICs (the two shaded boxes, denoted by Ξ_x and Ξ_{x+1}). The vertical axis is the intensity dimension, the horizontal axis is the retention time, denoted by scan Ω, and the m/z dimension is along the axis extending from the figure. The vertical lines within the shaded boxes are deisotoped peaks included in the trimmed XIC; vertical lines outside the shaded boxes are not included. The horizontal dashed lines delineate m/z ranges for each XIC and the vertical dashed lines delineate retention time boundaries for each XIC. The deisotoped peak list (denoted by Γ') contains all deisotoped peaks within a run.

6.5.2.5 Sub-step 2.e: Construct Peptide Signals

The fifth and final sub-step constructs peptide signals from XICs. It takes in a list of XICs, aggregates each XIC to construct a peptide signal, and outputs a vector of peptide signals. The algorithm is described in detail using pseudocode in the Algorithm 10: Construct Peptide Signal listing. In addition, Figure 6.8 depicts a trimmed XIC list (denoted by Ξ), two trimmed XICs (the two shaded boxes, denoted by Ξ_x and Ξ_{x+1}), and two newly constructed peptide signal peaks (denoted by K_x and K_{x+1}). Note that the peptide signal overlays the XIC's apex, that is, it is located at the m/z value of the XIC peak with the largest intensity. Also, the peptide signal's intensity is the sum of

the XIC peaks' intensities.

Algorithm 10 Construct Peptide Signals

Input $\Leftarrow$ Ξ : A vector of XICs
Output $\Rightarrow$ K : A vector of Peptide Signals

```
 1: function constructPeptideSignals(Ξ)
 2:     K ← initialized                                  ▷ vector of peptide signals
 3:     x ← 0
 4:     for i ← 0, Ξ.size − 1 do                         ▷ vector of XICs
 5:         Ψ′ ← Ξ_i                                     ▷ copy XIC to temporary XIC
 6:         a ← 0
 7:         y ← 0
 8:         for j ← 0, Ψ′.size − 1 do                    ▷ XIC vector
 9:             if Ψ′_j.intensity > a then               ▷ apex?
10:                 y ← j                                ▷ save index of apex
11:             end if
12:         end for
13:         κ ← Ψ′_y                                     ▷ new peptide signal
14:         n ← Ψ′.size
15:         κ.intensity ← Σ_{k=1}^{n} Ψ_k                ▷ sum XIC intensity
16:         κ.peaks ← Ψ′
17:         K_x ← κ                                      ▷ add to vector of peptide signals
18:         x ← x + 1
19:     end for
20:     return K
21: end function
```

The first two lines of the algorithm initialize a new vector to hold peptide signals (lines 2 and 3). The main outer loop (line 4) the processes the vector of XICs and the inner loop processes a single XIC's vector (line 6), detecting the XIC peak with the largest intensity and saving its index (lines 10 - 11). After the inner loop finishes, the algorithm copies the apex peak, identified with the saved index, to a new peak (line 13), thus constructing a new peptide signal. It then updates the new peptide signal's intensity with the sum of the XIC's constituent peaks' intensities (line 15) and the XIC (line 16). The last step within the algorithm's outer loop adds the new peptide signal to the vector of peptide signals (line 17 - 18). After constructing a peptide signal for each XIC, the algorithm returns the vector of peptides signals to the caller (line 20).

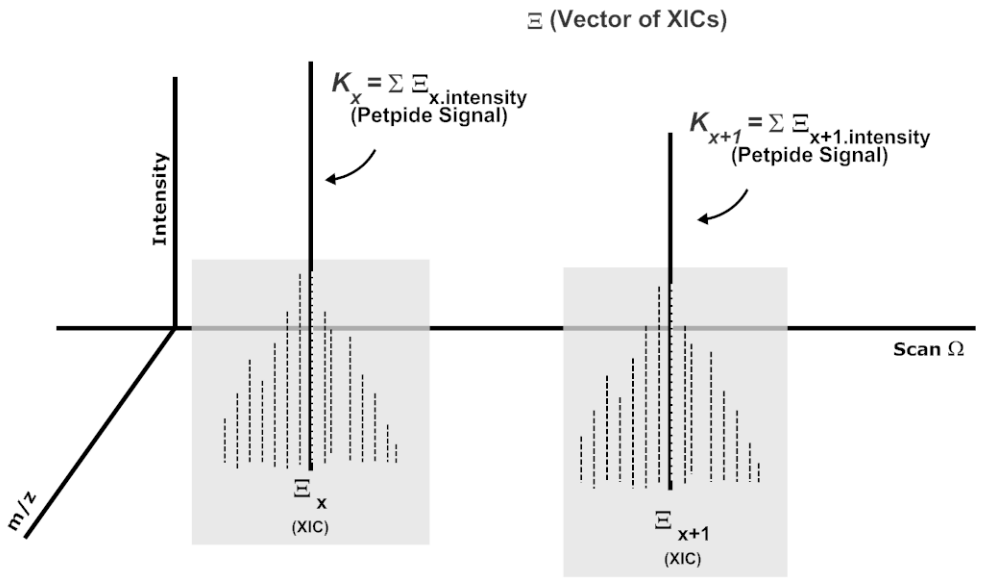


Figure 6.8: Cartoon of two trimmed and validated XICs (the two shaded boxes, denoted by Ξ_x and Ξ_{x+1}). The vertical axis is the intensity dimension, the horizontal axis is the retention time, denoted by scan Ω, and the m/z dimension is along the axis extending from the figure. Each XIC has a vertical line at its apex representing the newly constructed peptide signs (denoted by K_x and K_{x+1}). The apex is the XIC peak with the largest intensity. The peptide signal takes on the apex's m/z and retention time.

6.5.3 Step 3: Filter Peptide Signals

RIPPER's third step filters and removes poor quality peptide signals. As described in Chapter 3, XICs can be filtered prior to peptide signal construction. These filters work on XICs constructed from desisotoped peaks rather than XICs constructed from peptide signals. Furthermore, they compare XIC's distribution profile to a pseudo Gaussian distribution and invalidate those XICs not meeting minimum criteria. Not only is this a computationally intensive task, an XIC's distribution tends to be right tailed, which does not fit a Gaussian distribution. I chose instead to apply a filter after peptide signal construction. This filter works on XICs constructed from peptide signals and assumes that valid peptide signals have associated MS^2 scan(s) in at least one of the mzXML files

processed together in a single RIPPER analysis. Simply counting the number a peptide signal's associated MS^2 scans is computationally inexpensive. The downside with using this filter is that it requires tandem mass spectrometry, thus RIPPER disallows analysis of HPLC-ESI-MS runs where no MS^2 scans are taken. Fortunately, because RIPPER was designed to be extensible, modifying or replacing this filter will be straight forward.

After RIPPER processes each mzXML files through Steps 1 and 2, it filters the peptide signals. RIPPER combines the individual MS^2 scan object lists constructed in Step 1into a single MS^2 scan list prior to executing this sub-step. The algorithm takes in a vector of peptide signals, a vector of MS^2 scans, and user supplied filtering criteria and outputs a vector of filtered peptide signals. The algorithm is described in detail using pseudocode in the Algorithm 11: Quality Check listing.

Algorithm 11 Quality Check

Input $\Leftarrow$ K : A vector of Peptide Signals
Ω^2 : A vector of MS^2 scans from all runs
m : user supplied minimum number of MS^2 scans
t : user supplied maximum MS^2 retention time tolerance
e : user supplied maximum MS^2 m/z tolerance
Output $\Rightarrow$ K' : A vector of filtered Peptide Signals

```
1:  function QualityCheck(Ψ, K, Ω², m, t, e)
2:      K' ← initialized                          ▷ vector of filtered peptide signals
3:      x ← 0
4:      sort Ω² such that Ω²_x.mz < Ω²_{x+1}.mz and
            Ω²_x.charge ≤ Ω²_{x+1}.charge         ▷ sort ascending charge and m/z order
5:      for i ← 0, K.size − 1 do                  ▷ process vector of peptide signals
6:          κ ← K_i                               ▷ copy peptide signal
7:          y ← 0
8:          for j ← 0, Ω².size − 1 do             ▷ process vector of MS² scans
9:              if κ.charge == Ω².charge and      ▷ meets minimum criteria?
                  κ.mz > Ω².mz − e and κ.mz < Ω².mz + e and   ▷
                  κ.rt > Ω².rt − t and κ.rt < Ω².rt + t then  ▷
10:                 y ← y + 1
11:             end if
12:         end for
13:         if y > m then
14:             K'_x ← κ
15:             x ← x + 1
16:         end if
17:     end for
18:     return K'
19: end function
```

The first few lines of the algorithm perform housekeeping tasks. They initialize a new vector to hold validated peptide signals (lines 2 and 3) and sort the MS^2 scan vector in ascending charge and m/z order (line 4). The main outer loop processes the vector of peptide signals (line 5), and the inner loop processes searches through the MS^2 (line 8). Within the inner loop, the algorithm counts the number of MS^2 scans that match a single XIC's vector (line 6), detecting the XIC peak with the largest intensity and saving its index (lines 10 - 11). After the inner loop finishes, the algorithm copies the apex peak, identified with the saved index, to a new peak (line 13), thus constructing a new peptide signal. It then updates the new peptide signal's intensity with the sum of

the XIC's constituent peaks' intensities (line 15) and the XIC (line 16). The last step within the algorithm's outer loop adds the new peptide signal to the vector of peptide signals (line 17 - 18). After constructing a peptide signal for each XIC, the algorithm returns the vector of peptides signals to the caller (line 20).

6.5.4 Step 4: Normalize Peptide Signal Intensities

Step 4 implements Proximity-based Intensity Normalization (PIN) a new normalization method embodying the proportionality paradigm. Recall from Chapter 5 that PIN scales each peptide signal's intensity by the sum of neighboring peptide signal intensities within a temporal window. However, a neighboring peptide signal's parent XIC can straddle the temporal window boundary; thus, only a portion its intensity contributes to the normalization scaling factor. Fortunately, within RIPPER, Step 3 produces a vector of filtered peptide signals with vectors of their XICs.

The algorithm is described in detail using pseudocode in the Algorithm 12: PIN listing. In addition, Figure 6.9 depicts three peptide signals (vertical bold lines denoted by K_x,K_{x+1}, and K_{x+2} with their XICs deisotoped peaks are represented by vertical dashed lines. To compute K_{x+1}'s normalized intensity, PIN computes a scaling factor using peaks within a retention time window centered at K_{x+1}. Two horizontal bold lines indicate the boundaries for K_{x+1}'s retention time window. Deisotoped peaks within the retention time window are circled. K_{x+1}'s scaling factor is computed as the sum of the circled deisotoped peaks' intensities and K_{x+1}'s normalized intensity is computed as its intensity divided by its scaling factor. Note that the graph is three dimensional, but compared to Figure 6.8, it is rotated. Here, the intensity dimension remains the same, but m/z dimension returns to the horizontal axis and the retention time (scan) dimension is the axis extending out of the figure.

Algorithm 12 PIN

Input $\Leftarrow$ K : A vector of Peptide Signals
w : user supplied retention time window
Output $\Rightarrow$ K : A vector of normalized Peptide Signals

```
1:  function PIN(K, w)
2:      Γ' ← getAllXICPeaks(K)          ▷ put all XICs' deisotoped peaks in a single vector
3:      sort Γ' such that Γ'_x.rt < Γ'_{x+1}.rt          ▷ sort peaks in ascending rt order
4:      for i ← 0, K.size − 1 do          ▷ process vector of peptide signals
5:          κ ← K_i          ▷ copy peptide signal
6:          n ← 0          ▷ initialize scaling factor
7:          for j ← 0, Γ'.size − 1 do          ▷ process vector of XIC's deisotoped peaks
8:              γ' ← Γ'_j          ▷ copy peak
9:              if γ'.rt ≥ κ.rt + w/2 then          ▷ outside max rt window?
10:                 break          ▷ get out of loop
11:             end if
12:             if γ'.rt > κ.rt − w/2 and γ'.rt < κ.rt + w/2 then          ▷ in rt window?
13:                 n ← n + γ'.intensity          ▷ add peak intensity to scaling factor
14:             end if
15:         end for
16:         K.normalizedIntensity ← K.intensity / n          ▷ NORMALIZE
17:     end for
18:     return K
19: end function
```

The first line of the PIN algorithm calls the function getAllXICPeaks (line 2). This function extracts all the XICs' deisotoped peaks from the vector of peptide signals and returns a vector of these peaks. The next line sorts the vector of all XIC deisotoped peaks in ascending retention time order (line 3). The main outer loop processes each peptide signal vector (line 4). For each peptide signal, the algorithm first initializes the normalization scaling factor (line 6). Then, the inner loop processes the vector of all XIC deisotoped peaks (line 7). If the retention time of the deisotoped peak is greater than the maximum retention time window boundary, the algorithm breaks out of the inner loop (lines 9 - 11). It does so to save processing time. Next, the algorithm checks if the deisotped peak's retention time is within the peptide signals retention time window (line 14). If it is, the algorithm adds the deisotoped peak's intensity to the normalization scaling factor (lines 13). After exiting the inner loop, the algorithm then normalizes the peptide signal's intensity (line 16). It does so by simply dividing its intensity by the

scaling factor. After processing all peptide signals, the algorithm returns the peptide signal vector to the caller (line 18).

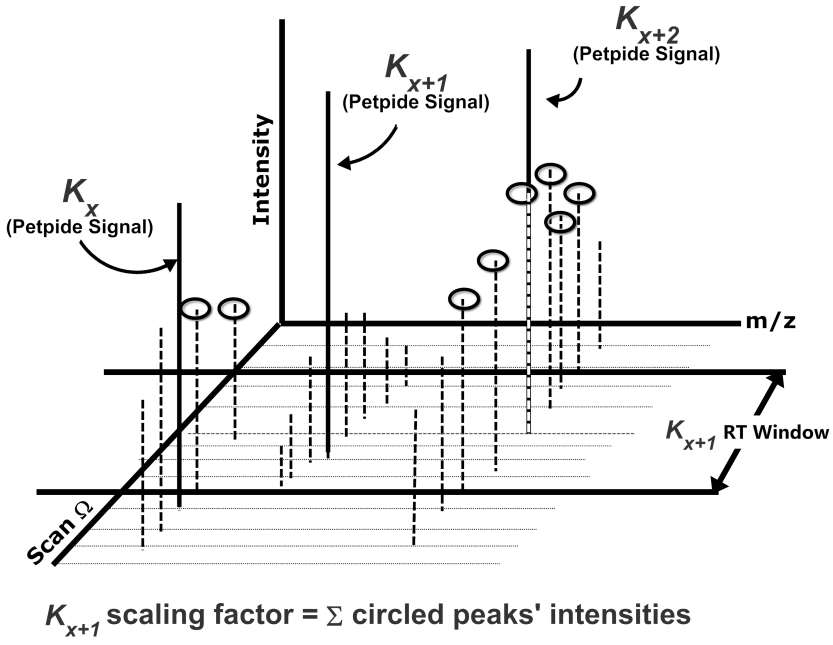


Figure 6.9: Cartoon of computing the scaling factor for normalizing a single peptide signal with PIN. Depicted are three peptide signals (vertical bold lines denoted by K_x,K_{x+1}, and K_{x+2}. Their XICs deisotoped peaks are represented by vertical dashed lines. To compute K_{x+1}'s normalized intensity, PIN computes a scaling factor using peaks within a retention time window centered at K_{x+1}. Two horizontal bold lines indicate the boundaries for textitK$_{x+1}$'s retention time window. Deisotoped peaks within the retention time window are circled. K_{x+1}'s scaling factor is computed as the sum of the circled deisotoped peaks' intensities and K_{x+1}'s normalized intensity is computed as K_{x+1}.intensity / K_{x+1}.scalingfactor.

6.5.5 Step 5: Group Peptide Signals Across Multiple Runs

RIPPER's fifth processing step groups peptide signals across runs. As described in Chapter 3, retention times drift due to changes in experimental conditions, for example, column performance or column overloading, impedes accurate grouping of peptide signals across LC-MS runs. Most existing frameworks overcome this obstacle by using

retention time alignment to correct retention time drift prior to peptide signal extraction. However, some frameworks, such as msInspect [75] group peptide signals across runs within retention time windows. I chose an approach similar to msInspect because the data sets analyzed via RIPPER thus far have similar temporal dimensions. The algorithm is described in detail using pseudocode in the Algorithm 13: Group Peptides listing. To group peptides across multiple runs, this algorithm takes in a vector of all peptide signals from all runs processed together in a single RIPPER run. The algorithm then groups the peptide signals using and indexing scheme, and then outputs the incoming vector of peptide signals, but with their group indexes updated.

Algorithm 13 Group Peptides

```
    Input ⇐  K : A vector of all Peptide Signals from all runs
             t : user supplied retention time tolerance
             e : user supplied m/z tolerance
    Output ⇒  K : A vector of grouped Peptide Signals
 1: function GROUPPEPTIDESIGNALS(K, t, e)
 2:     sort K such that K_x.mz < K_{x+1}.mz and      ▷ sort in charge and m/z ascending order
        K_x.charge ≤ K_{x+1}.charge
 3:     y ← 0                                                          ▷ group index
 4:     for i ← 0, K.size − 1 do                       ▷ process vector of peptide signals
 5:         if K_i.groupNumber == 0 then                               ▷ not already used?
 6:             K_i.groupNumber ← y                                    ▷ group number
 7:             for j ← i + 1, K.size − 1 do           ▷ process next peptide signals
 8:                 if K_i.groupIndex == 0 then                        ▷ already used?
 9:                     if K_j.mz > K_i.mz + e or                      ▷ outside m/z range?
                        K_j.charge ≠ K_i.charge then
10:                         break                                      ▷ get out of loop
11:                     else
12:                         if K_j.rt < K_i.rt + t then                ▷ within rt range?
13:                             K_i.groupNumber ← y                    ▷ group number
14:                         end if
15:                     end if
16:                 end if
17:             end for
18:             y ← y + 1‘
19:         end if
20:     end for
21:     return K’
22: end function
```

The Group Peptides algorithm begins by sorting the vector of all peptide signals in ascending charge and m/z order (line 2). It continues by initializing the group number index (line 3). The main outer loop processes peptide signal vector (line 4). Within the main outer loop, the algorithm first checks if the peptide signal has already been used in a group (line 5). If not, this peptide signal becomes the first member of a group. To identify the group number, the algorithm assigns a group index to the peptide signal (line 6). The inner loop looks ahead in the peptide signal vector (line 7). If the next peptide signal has already been used, the loop continues to look ahead for the next peptide signal (line 8). If the next peptide signal's m/z value is outside of the first group member's m/z range (within the user supplied m/z tolerance), the algorithm breaks out of the inner loop (lines 10 and 11). Next, the algorithm checks to see if the next peptide signal's retention time is greater than the first member's retention time plus the user supplied retention time tolerance (line 13). After the inner loop finishes, the group is complete. The algorithm then increments the group number index (line 18) and continues processing in the main outer loop. After the main outer loop finishes, the algorithm returns the peptide signal vector, with its updated group indexes, to the caller.

6.5.6 Step 6: Optionally Identify Peptides/Proteins

RIPPER does not identify peptides and proteins associated with m/z values directly. However, it can match a peptide's m/z, charge, and retention time values (within user specified tolerances) to output generated by external protein and peptide identification software, here Scaffold [108]. Most often, m/z and charge values are sufficient to match a RIPPER peptide signal to a Scaffold identified peptide. But, on occasion, two peptides will have nearly identical m/z and charge values, but elute at different times. Thus, using only m/z and charge values would result in erroneous RIPPER to Scaffold matches.

Unfortunately, matching RIPPER to Scaffold using retention times, in addition to m/z and charge values, is not straight forward. While peptide signals within RIPPER

have a retention time, identified peptides within Scaffold do not. But, as described in Chapter 3 Scaffold's takes in scored peptide identifications generated from other software packages. Fortunately, when peptides are identified by the software package SEQUEST [90] prior to being input into Scaffold, each MS^2 scan is assigned text description that has its scan number embedded. Furthermore, Scaffold's spectrum report includes this text. This scan number for this report, combined with a run identifier, is used to locate the MS^2 in RIPPER's vector of all MS^2. Then, the matching RIPPER MS^2 scan's retention time are used as a surrogate for the missing identified peptide's retention time.

To implement this algorithm, I turned to a student working under the Undergraduate Research Opportunity (UROP) grant, Kathryn Doroschak. Together, we fleshed out the details of the algorithm, which is described in detail using pseudocode in the Algorithm 14: Identify Peptides listing.

Algorithm 14 Identify Peptides

Input $\Leftarrow$ K : A vector of all Peptide Signals from all runs
Ω^2 : A vector of all MS^2 scans
$fileName$: Scaffold extract file name
t : user supplied retention time tolerance
e : user supplied m/z tolerance
Output $\Rightarrow$ K : A vector of identified Peptide Signals

```
1:  function matchToScaffold(K, fileName, t, e)
2:      S ← getScaffoldScans(fileName)                    ▷ parse scans from Scaffold
3:      hashMap ← getScanRT(Ω², S)                        ▷ get retention time / scan has map
4:      for i ← 0, K.size − 1 do                          ▷ process vector of peptide signals
5:          for j ← 0, S.size − 1 do                      ▷ process vector of Scans
6:              if S_j.mz > K_i.mz + e or                 ▷ outside m/z range?
                        S_j.charge ≠ K_i.charge
7:                  break                                 ▷ get out of loop
8:              else
9:                  v ← scan RT from hashMap              ▷ lookup Scaffold scan retention time
10:                 if K_j.rt > v − t and then            ▷ within rt range?
                        K_j.rt > v − t and
                        S_j.mz > K_i.mz − e and           ▷ within m/z range?
                        S_j.mz ≤ K_i.mz + e and
                        S_j.charge == K_i.charge then
11:                     K_i.peptideID ← S_j.peptideID     ▷ peptide identificaiton
12:                 end if
13:             end if
14:         end for
15:     end for
16:     return K
17: end function
```

The algorithm begins by calling the getScaffoldScans function which parses the Scaffold spectrum report extract (line 2) and returns them in a vector of Scaffold scans. This function returns the vector of Scaffold scans in ascending m/z and charge order. Next, it creates a hash map containing the Scaffold scan number and RIPPER's matching MS^2 scan's retention time. It does so by calling the getScanRT function (line 3). The main outer loop processes the peptide signal vector (line 4), and the inner loop processes the Scaffold scans (line 5). If the retention time of the Scaffold is greater than the maximum retention time window boundary, the algorithm breaks out of the inner loop (lines 6 - 8). It does so to save processing time. Next, the algorithm looks up the Scaffold

scan's retention time in the hash map (line 9). If the Scaffold scan's m/z value, charge, and retention times are similar (within tolerances) to the peptide signal's corresponding values (line 10), the algorithm assigns the Scaffold scan's peptide identification to the peptide signal (line 11). After the main outer loop finishes, the algorithm returns the vector of peptide signals to the caller.

6.6 Peptide Signal Intensity and Optional Reports

RIPPER generates several reports in a comma delimited format (.csv). This format allows users to import the reports into a spreadsheet application, for example, Microsoft Excel^{TM}. By default, RIPPER generates two quantitative peptide signal reports. Additionally, in debug mode, RIPPER generates several intermediate reports.

6.6.1 Peptide Signal Intensity Reports

RIPPER generates two listings containing peptide signal reports: un-normalized peptide signal quantities and normalized peptide signal quantities. Both reports share the same format. As shown in Figure 6.10, each row contains a unique peptide signal identified by the first two columns, charge and peptide signal (m/z). The remaining columns contain intensity values for each peptide signal; one column for each processed mzXML file.

Un-normalized Peptide Signal Intensity Report

Charge	Peptide Si	cncr_109	cncr_120	cncr_130	cncr_135	cncr_140	cncr_141	cncr_147	cncr_150	cncr_154	cncr_157	cncr_158
2	360.2015	0	0	0	0	0	0	0	0	0	0	51064.69
2	360.2067	0	0	0	0	7339050	0	0	0	0	0	0
2	360.2161	0	0	297702.9	0	1048419	0	0	0	1440261	0	0
2	360.2556	0	0	0	0	0	0	0	0	0	0	0
2	360.6115	0	0	0	0	0	0	0	0	0	0	0
2	360.666	0	0	0	1457061	2806564	71585.36	0	290802.6	0	950019.7	165649.9
2	360.6836	0	0	0	0	0	0	12797.42	0	168855.1	0	0
2	360.6902	0	0	0	0	0	239014.1	0	0	0	0	0
2	360.6937	0	64516.26	0	0	0	0	0	0	0	0	0
2	360.7029	0	630573	0	0	159775.3	533719.3	0	0	0	26451.24	36111.36
2	360.7058	1186450	0	1210341	63106.11	1024735	0	1706988	2002876	386365.4	1391109	0
2	360.7255	1550884	7322439	8370415	1485801	774483.5	457240.7	1019886	1444651	0	348367.6	1281818
2	361.1295	0	0	0	124553.4	733463.3	0	0	0	0	0	0

Normalized Peptide Signal Intensity Report

Charge	Peptide Si	cncr_109	cncr_120	cncr_130	cncr_135	cncr_140	cncr_141	cncr_147	cncr_150	cncr_154	cncr_157	cncr_158
2	360.2015	0	0	0	0	0	0	0	0	0	0	0.003675
2	360.2067	0	0	0	0	0.021947	0	0	0	0	0	0
2	360.2161	0	0	0.01044	0	0.006768	0	0	0	0.007105	0	0
2	360.2556	0	0	0	0	0	0	0	0	0	0	0
2	360.6115	0	0	0	0	0	0	0	0	0	0	0
2	360.666	0	0	0	0.008032	0.03087	0.004984	0	0.002982	0	0.003354	0.012624
2	360.6836	0	0	0	0	0	0	0.002917	0	0.00454	0	0
2	360.6902	0	0	0	0	0	0.003614	0	0	0	0	0
2	360.6937	0	0.607788	0	0	0	0	0	0	0	0	0
2	360.7029	0	0.007637	0	0	0.003214	0.005631	0	0	0	0.002405	0.003314
2	360.7058	0.002467	0	0.001598	8.84E-04	0.001644	0	0.001512	0.002024	0.001223	0.002277	0
2	360.7255	0.002726	0.010399	0.005916	0.002948	0.002583	0.003811	0.001184	0.001665	0	0.002807	0.008941
2	361.1295	0	0	0	0.014198	0.020766	0	0	0	0	0	0

Figure 6.10: RIPPER produces two reports, each containing the same peptide signals. The first report contains un-normalized peptide signal intensities (top), and the second report contains normalized peptide signal intensities (bottom).

6.6.2 Optional Intermediate Reports

RIPPER generates two types of intermediate reports. The first type contains information from each individual mzXML file (see Table 6.1). The second type contains combined information from all mzXML files processed in a single RIPPER run (see Table 6.2).

Report	**Description**
MS^2 Scans	MS^2 scan data, excluding peak data, but including run identifier, scan number, retention time, precursor m/z value, and precursor charge
MS^1 Scans	MS^1 scan data, excluding peak data, but including run identifier, scan number, retention time, number of peaks, low m/z value, high m/z value, and total ion current
Extracted Deisotoped Peaks	MS^1 desistoped peak data, including run identifier, scan number, retention time, m/z value, and intensity
Extracted XIC Peaks	XIC peak data, including run identifier, scan number, retention time, m/z value, intensity, apex peptide signal intensity, and apex retention time
Extracted Peptide Signals	Peptide signal data, including run identifier, apex scan number, apex retention time, apex m/z value, summed intensity, apex peptide signal intensity, and apex retention time

Table 6.1: RIPPER Peptide Signal Intensity Reports Column Detail

Report	Description
MS^2 Scans	MS^2 scan data, excluding peak data, but including run identifier, scan number, retention time, precursor m/z value, and precursor charge
MS^1 Scans	MS^1 scan data, excluding peak data, but including run identifier, scan number, retention time, number of peaks, low m/z value, high m/z value, and total ion current
Extracted Deisotoped Peaks	MS^1 desistoped peak data, including run identifier, scan number, retention time, m/z value, and intensity
Extracted XIC Peaks	XIC peak data, including run identifier, scan number, retention time, m/z value, intensity, apex peptide signal intensity, and apex retention time
Extracted Peptide Signals	Peptide signal data, including run identifier, apex scan number, apex retention time, apex m/z value, summed intensity, apex peptide signal intensity, and apex retention time
Peptide Signals with MS^2 Data	Peptide signals with their associated MS^2 data, if present. Includes data contained in the Peptide Signals report as well as the MS^2 precursor m/z value, the MS^2 precursor minimum retention time, and the MS^2 maximum retention time
XIC Deisotoped Peaks	XIC peak data after applying filters. Same format as XIC Peaks Report.

Table 6.2: Optional Reports, One Per mzXML File Processed

6.7 Discussion

RIPPER's was born out of frustration with existing iLFRQ software frameworks' inability to meet my needs for analyzing peptidomic data. Therefore, RIPPER's implementation addresses several limitations present in existing frameworks. Of course, it also comes with its own set of limitations.

RIPPER addresses several limitations in existing iLFRQ software frameworks. First, RIPPER reports normalized intensities for peptide signals rather than proteins. This allows users, such as the Griffin Lab members, to conduct large scale comparative studies on endogenous peptides using iLFRQ. Second, RIPPER optionally provides intermediate reports. These reports provide users with details on RIPPERs peptide signal extraction steps. In essence, these reports provide an audit trail that tracks each of RIPPER's steps, from isotopic peak envelope detection and validation through filtering and normalizing peptide signals. Thus, if a user has questions on the origin of a peptide signal or normalized intensity, the questions can be easily answered by perusing the intermediate reports.

As described in Chapter 3, other frameworks do not provide this capability. Third, RIPPER only requires one peak in an isotopic envelope above the signal to noise threshold. Other frameworks require all peaks in an isotopic envelope above the signal to noise threshold. This allows RIPPER to detect low abundance peptide signals from peptides other frameworks miss. Fourth, and perhaps most importantly, RIPPER is the only software framework implementing PIN. As demonstrated in the following chapter, PIN dominates other normalization methods.

Of course, RIPPER also has its own limitation. First, RIPPER is not well suited for experiments using pre-fractionation, for example, MUDPit experiments. As described in Chapter 3, researchers often prefractionate to reduce a sample's dynamic range which allows detection of lower abundance peptides. However, one of RIPPER's advantages is that it can detect lower abundance peptide signals. Second, RIPPER takes in standard

formatted mzXML files with centroid peak data; it currently does not support the newest standard format for MS data, mzML. However, I have observed that adoption by the proteomics community of the newer mzML file format has been slow. Therefore, the vast majority of standardly formatted MS data deposited in public repositories is still mzXML data. Furthermore, the metabolomics community is only beginning to discuss mzML's adoption[181]. Third, RIPPER requires MS/MS data. As described in Section 5.3, the peptide signal filtering step requires at a minimum number of MS^2 precursors for each peptide signal. However, most MS/MS experiments use data dependent acquisition mode. This mode only selects the most intense ions for subsequent MS^2 scans. Hence, low abundance peptide signals found in MS^1 scans often do not trigger an MS^2 scan. Therefore, while RIPPER is capable of detecting low abundance peptide signals, the peptide signal filtering removes them.

Software that lasts evolves. For RIPPER to survive, it must overcome limitations, enhance capabilities, and meet new challenges [182]. The first order of business is to remove limitations where possible. Unfortunately, the first limitation described, no prefractionation, would require an impractical amount of time to address software. Therefore, it will not be addressed in the near future. However, the other two limitations described previously, requiring MS^2 scans and only processing mzXML files, can be solved by software enhancements in a reasonable amount of time. Of the two, removing RIPPER's requirement for MS^2 scans is a higher priority because it limits RIPPER's user base more than only processing mzXML files. To remove the MS^2 scan requirement, I plan to implement a replacement for the peptide signal filtering algorithm. The new algorithm will validate XICs prior to peptide signal construction. It will evaluate an XIC's similarity to a generalized extreme value distribution [183] using a pattern matching scoring model, for example, KL. Integrating the new validation algorithm into RIPPER should be straightforward because of RIPPER designed extensibility.

The next order of business is to extend RIPPER's capabilities. First, researchers are

ultimately looking for a list of statistically significant differences between two populations' peptide signal intensities. Ideally, RIPPER would provide this capability. Currently users must use with third party software, for example, R or Matlab. They must manually import peptide signal reports generated by RIPPER into these statistical packages for analyses. Although the first steps toward providing this capability have been taken, and initial testing confirms that RIPPER can interface with R and run statistical analyses, higher priority activities precluded further work. Second, RIPPER lacks visualization capabilities commonly found in other highly used frameworks. While this is not a RIPPER limitation, per se, however, visualization is appealing to many users. Thus, I believe that integrating visualization, such as three-dimensional peaks, XICs, heatmaps, and volcano plots, would attract a larger user base.

The fast pace of mass spectrometry advances will increasingly present challenges for software processing their generated spectral data. While these challenges are unpredictable, I believe that because its architecture is designed for success, RIPPER can be extended with a reasonable amount of effort to meet those challenges.

Chapter 7

Evaluation

"Quod erat demonstrandum." [1] - Euclid

7.1 Introduction

This chapter evaluates the proportionality paradigm (as applied to HPLC-ESI-MS/MS data), Proximity-based Intensity Normalization (PIN), and RIPPER. It does so using datasets from HPLC-ESI-MS/MS analyses of complex peptide mixtures. The evaluation results demonstrate that PIN dominates current normalization methods in reducing systematic bias and complex variability. Furthermore, it does so while retaining the ability to statistically significant biological variation. The relationships between the evaluation experiments and the data sets used is complex. Here, I provide two tables, Table 7.1 and Table 7.2, describing these relationships.

The remainder of this chapter is organized as follows. Section 7.2 motivates the experiments for evaluating RIPPER/PIN. Section 7.3 describes the five data sets used in the seven experiments listed in Section 7.4. Section 7.5 discusses each experiment, providing observations as warranted. Finally, Section 7.6 provides a brief summary of the evaluation of RIPPER/PIN.

[1] Translation "Which was to be proved."

Exp. No.	Data Set Name	Sample Name	Overview Section	Methods Appendix
1	Serial Dilution	Salivary Peptides	7.3.1	C.3.1
2	Instrument Variability	Salivary Peptides	7.3.2	C.3.1
3	Sample Variability	Salivary Peptides	7.3.3	C.3.1
4	CPTAC Study 6	UPS1 & Yeast	7.3.4	C.3.2
5	OPML vs. OSCC	Salivary Peptides	7.3.5	C.3.1

Table 7.1: Data Sets, Sample Types, and Description Locations - Lists the five data sets used as input into the PIN evaluation experiments. The first two columns number and name the data set and the third column lists the sample type used to generate the data set. The fourth column provides a section reference for the data set's general description, and the fifth column provides an appendix reference for the data set's generation detailed methods.

No.	Name	Description	Data Sets	Overview	Methods
A	SN vs Peptide Signal	Demonstrates the discrepancy in recorded signal above the signal to noise threshold and the record signal stemming from peptides.	1	Section 7.4.1.2	Appendix C.4.1
B	Systematic Bias	Demonstrates the systematic bias portion of extraneous variability.	1,2,3,4	Section 7.4.2	Appendix C.4.2
C	Complex Variability	Demonstrates the complex variability portion of extraneous variability.	4	Section 7.4.3	Appendix C.4.2
D	Internal Standard (Spike-in)	Demonstrates absolute abundance estimation	1	Section 7.4.4	Appendix C.4.4
E	Repeatability	Measures PIN's ability to reduce variance in complex biological samples vis-a-vis common normalization methods.	1,2,3,4	Section 7.4.5	Appendix C.4.5
F	Overfitting	Measures PIN's ability to detect biological variation after normalization.	4	Section 7.4.6	Appendix C.4.6
G	Biomarker Discovery	Applies PIN to a biomarker discovery set.	5	Section 7.4.7	Appendix C.5

Table 7.2: Experiments, Data Sets, and Description Locations - Lists the seven PIN evaluation experiments. The first two columns number and name the data set and the third column briefly describes each experiment. The fourth column lists the data sets I used in the experiment. The fifth column provides a section reference for the data set's general description and the sixth column provides an appendix reference for the experiment's detailed methods.

7.2 Motivation

Once the RIPPER framework was completed, I sought to evaluate its PIN normalized results. The obvious choice for evaluating PIN (and inherently the proportionality paradigm) was to design experiments comparing PIN's results to results from existing software frameworks. However, as described in Chapter 6, existing software frameworks report protein level quantification rather than peptide level quantification. Therefore, comparing PIN's results, which report results at the peptide level, to existing software frameworks results is an untenable experimental design. In lieu of evaluating PIN against existing software frameworks, I sought to demonstrate PIN's dominance by building a case where I first show examples of extraneous variability, including systematic bias and complex variability. Second, I compare PIN against common normalization methods in reducing extraneous variability. Third, I prove that PIN does not overfit the data. Finally, I apply PIN to a biomarker discovery study, showing that I find biological variation otherwise missed.

7.3 Data Sets

The Griffin Lab at the University of Minnesota maintains several archived data sets from various experiments analyzing saliva via HPLC-ESI-MS/MS. I use three archived data sets from a saliva sample handling experiment [27] and an archived data set from a oral premalignant lesion (OPML) to oral squameous cell carcinoma (OSCC) transition biomarker discovery experiment (data unpublished). In addition, I use a standard reference data set from the Clinical Proteomic Tumor Analysis Consortium. The following subsections provide an overview of each data set used in the evaluation experiments.

7.3.1 Serial Dilution

To measure the correlation between protein (or in this case peptide) concentrations and measurements by an analytical technique, researchers often conduct serial dilutions. A serial dilution protocol stepwise dilutes one mixture (substance or solution) into a second mixture (again, a substance or solution). In mass spectrometry, it characterizes the relationships between signals in successive dilution steps [184]. Figure 7.1 depicts a series of serial dilution mixtures and the resulting calibration curve.

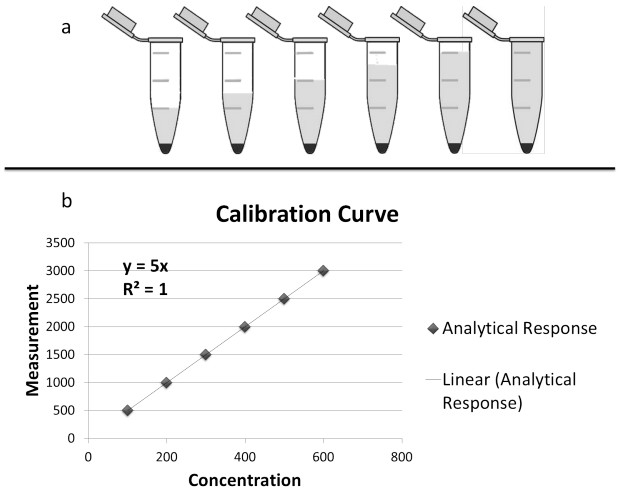


Figure 7.1: Cartoon of serial dilution mixtures and the resulting calibration curve. a) Each tube contains a constant amount of a standard mixture (dark area at the bottom of each tube) and incremental changes in amount of a second mixture (grey area in each tube). b) An ideal calibration curve as a result of analyzing the mixtures in panel a. Note that here, the correlation coefficient (R^2) = 1.0, which is perfect correlation.

Appendix C.3.1 describes in detail the protocol to generate the Serial Dilution data set used in evaluating PIN. Briefly, we placed incremental amounts (0.5μg, 1.0μg, 1.5μg, 2.0μg, 2.5μg, and 3μg) of salivary endogenous peptides from the same mixture into individual eppendorf tubes. To each tube, we added a constant amount (500 fmol) of bradykinin. We then analyzed each mixture once via HPLC-ESI-MS/MS and identified peptide sequences and their predecessor proteins using SEQUEST and Scaffold.

Run ID	Amt. of Salivary Endogenous Peptides
ssd_0_5	0.5μg
ssd_1_0	1.0μg
ssd_1_5	1.5μg
ssd_2_0	2.0μg
ssd_2_5	2.5μg
ssd_3_0	3.0μg

Table 7.3: Run Identifiers for Serial Dilution Data Set

7.3.2 Instrument Variability

To measure the amount of inter-run variability directly stemming from changes between HPLC-ESI-MS/MS runs, we conducted an Instrument Variability experiment. Appendix C.3.1 describes in detail the protocol used to generate the Instrument Variability data set used in evaluating PIN. Briefly, we placed salivary endogenous peptide mixture from a single donor into an autosampler vial. We then analyzed the mixture three times in succession via HPLC-ESI-MS/MS and identified peptide sequences and their predecessor proteins using SEQUEST and Scaffold.

Run ID	Description
ir1	Technical Replicate 1
ir2	Technical Replicate 2
ir3	Technical Replicate 3

Table 7.4: Run Identifiers for Instrument Variability Data Set

7.3.3 Sample Variability

To measure the amount of variability stemming from changes between sample preparations I conducted a Sample Variability experiment. Note that sample variability, when analyzed via HPLC-ESI-MS/MS inherently comprises variability directly stemming from differences in sample preparation and underlying instrument variability. Appendix C.3.1

describes in detail the protocol used to generate the Sample Variability data set used in evaluating PIN. Briefly, we prepared salivary endogenous peptide mixtures from a single donor three times. We then analyzed each mixture once via HPLC-ESI-MS/MS and identified peptide sequences and their predecessor proteins using SEQUEST and Scaffold.

Run ID	Description
sr1	Technical Replicate 1
sr2	Technical Replicate 2
sr3	Technical Replicate 3

7.3.4 CPTAC Study 6

"The Clinical Proteomic Technologies for Cancer Initiative, CPTAC Phase I 2006 - 2011, was developed to address the pre-analytical and analytical variability issues that are major barriers to the field of proteomics." - https://cptac-data-portal.georgetown.edu/cptac /study/list?scope=Phase+I. The resulting data sets were generated from various laboratories using the same standard operating procedure protocols [111, 176, 28]. Here, I use a subset of the data generated in CPTAC Study 6 where I observed complex variability in one of three replicates. Appendix C.3.2 describes data procurement and peptide and protein identification details. Briefly, I downloaded the CPTAC Study 6 data from Tranche and identified proteins using SEQUEST and Scaffold.

Table 7.5: Run Identifiers for Sample Variability Data Set

7.3.5 OPML vs. OSCC

The OPML vs. OSCC data set was generated as a result of a biomarker study conducted in the Griffin Lab. The goal of this study was to discover salivary endogenous peptides predictive for transition from benign lesions to malignant dysplasia in oral cancer. Appendix C.3.1 describes in detail the protocol used to generate the OPML vs. OSCC data set used in evaluating PIN. Briefly, I prepared salivary endogenous peptide mixtures from 17 patients with OPMLs and 18 patients with OSCC. We then analyzed each mixture once via HPLC-ESI-MS/MS and identified peptide sequences and their predecessor proteins using SEQUEST and Scaffold.

Run ID	Description
CPTAC_ORBI_UPS1_C_REP1	Sample C - Technical Replicate 1
CPTAC_ORBI_UPS1_C_REP2	Sample C - Technical Replicate 2
CPTAC_ORBI_UPS1_C_REP2	Sample C - Technical Replicate 3
CPTAC_ORBI_UPS1_E_REP1	Sample E - Technical Replicate 1
CPTAC_ORBI_UPS1_E_REP2	Sample E - Technical Replicate 2
CPTAC_ORBI_UPS1_E_REP2	Sample E - Technical Replicate 3

Table 7.6: Run Identifiers for CPTAC Study 6 Data Set

7.4 Experiments

This section describes the evaluation experiments and their results. (Here I use the term experiments for the specific evaluation tasks even though they are not hypothsis-driven tests.) The first six subsections RIPPER/PIN's ability to mitigate systematic bias and complex variability, improving repeatability and reproducibility without overfitting. The seventh and last subsection describes a biomarker discovery study to which I applied RIPPER/PIN.

7.4.1 SN vs Peptide Signal

As a first step, I wanted to demonstrate that chromatograms contain a large amount of signal not attributable to peptides. One way to demonstrate this is to plot the signal recorded above the signal to noise threshold and the record signal stemming from peptide signals in a chromatogram. The following two sections provide 1) an overview of the experiment and methods, and 2) the results of the experiment.

7.4.1.1 Overview

I use the Serial Dilution (1) data set to demonstrate the difference between signal recorded over the signal to noise threshold and signal stemming from peptide detection. The Serial Dilution data set is described in Section 7.3.1 and its generation detailed in

Appendix C.3. I chose this data set because I was able to construct a calibration curve and calculate the original amount of peptide mixture analyzed via HPLC-ESI-MS/MS. Furthermore, each of the six mixtures differed only in the amount loaded and the concentration of bradykinin. The difference between signal above signal to noise threshold and peptide signal is best demonstrated via visual inspection. Using reports generated by RIPPER, I used R to plot the extracted chromatograms in the following section (see Appendix C.2.1 for plot generation details).

7.4.1.2 Results (A1)

The results presented here allow for visual inspection of signal above the SN threshold and peptide signals for the Serial Dilution data set. First, Figure 7.2a plots peptide signal XCs for each of the six runs in the Serial Dilution data set. I expected to see monotonically increasing regression lines corresponding to the increasing amounts of salivary endogenous peptides in each run. Four of the six regression lines (grey lines) follow expectations. However, two regression lines (annotated black lines - 2.0 μg and 3.0μg runs) do not follow expectations. In fact, the run with the greatest amount of salivary peptide mixture (3.0 μg run) recorded less peptide signal than four of the five other runs. First, Figure 7.2b plots signal greater than SN threshold XCs for each of the six runs in the Serial Dilution data set. Again, I expected to see monotonically increasing regression lines corresponding to the increasing amounts of salivary endogenous peptides in each run. Again, four of the six regression lines (grey lines) follow expectations and the other two (annotated black lines - 2.5 μg and 3.0μg runs) do not. However, here, the 2.5 μg run regression line replaces the 2.0 μg run regression line.

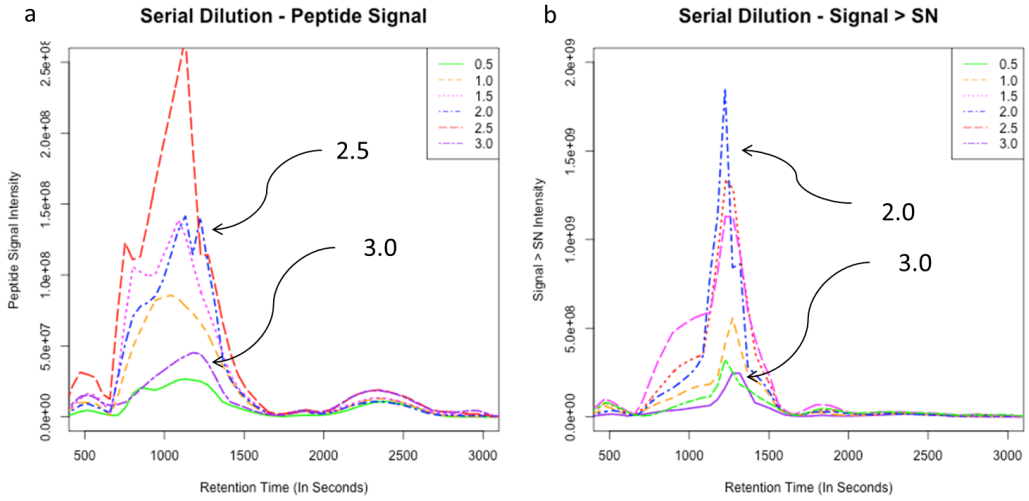
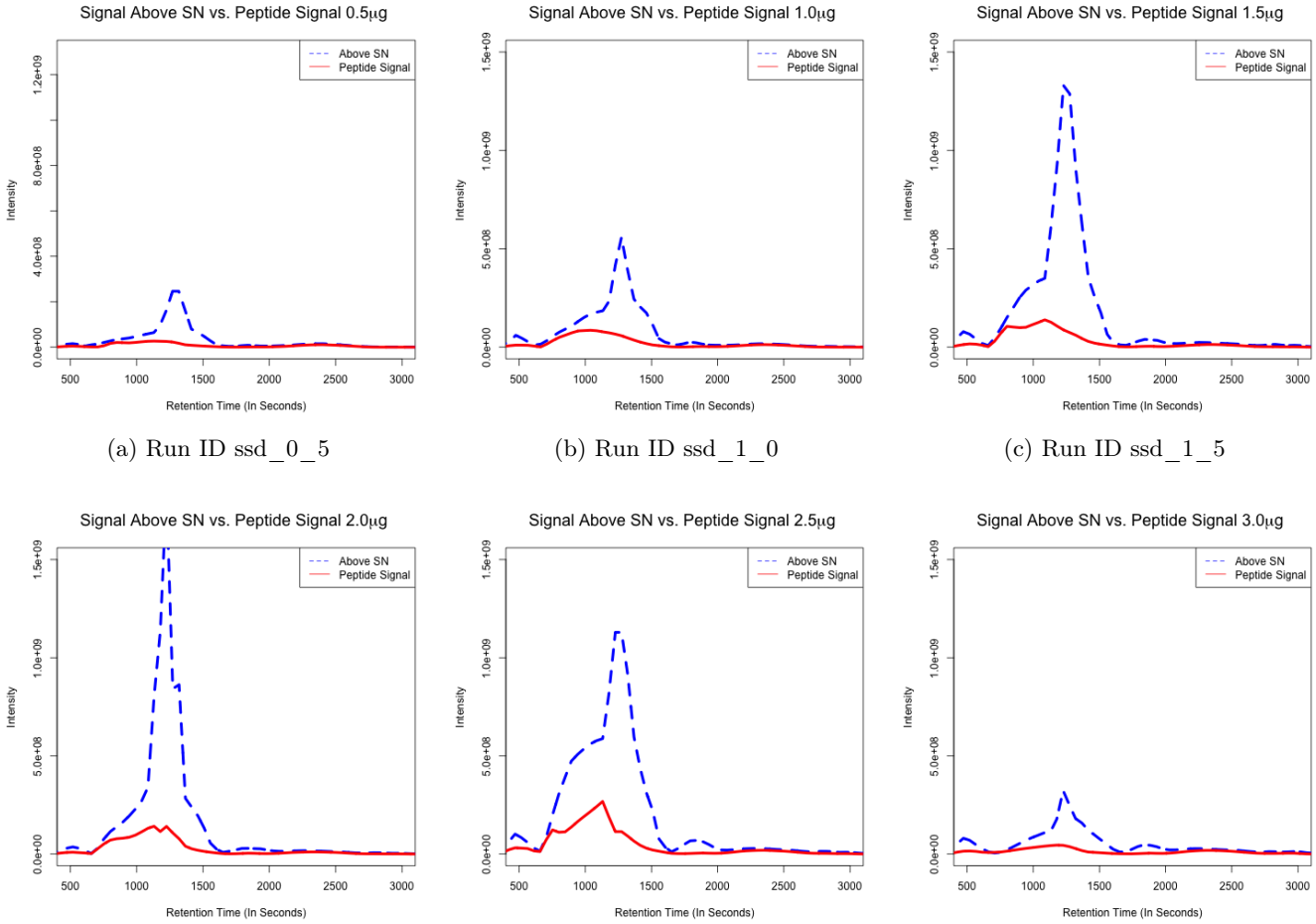


Figure 7.2: Serial Dilution Signal > SN Threshold and Peptide Signal XCs by XC type - the black lines depict XCs not meeting monotonically increasing expectations.

Figures 7.3 shows the Serial Dilution data set's peptide signal XCs overlaid with the signal greater than SN threshold XC for each run. Note that, visually, the proportion of signal above SN threshold that stems from peptide signal varies within runs as well as between runs.



(a) Run ID ssd_0_5 (b) Run ID ssd_1_0 (c) Run ID ssd_1_5

(d) Run ID ssd_2_0 (e) Run ID ssd_2_5 (f) Run ID ssd_3_0

Figure 7.3: Serial Dilution Signal > SN Threshold and Peptide Signal XCs by Run

7.4.2 Systematic Bias

Next, I sought to analyze PIN's effect on systematic bias compared to common normalization methods. Figures 7.2 and 7.3 in the previous section clearly show systematic bias resulting from loading amount differences from an unknown source. Another way to visualize systematic bias is through Minus vs Average (MA) plots (see Appendix C.2.2). First developed for microarray studies [151], MA plots are also useful in MS-based proteomic studies to see differences in between run variations and intensity dependent trends when comparing normalization methods [117]. The following two sections provide 1) an overview of the experiment and methods, and 2) the results of the experiment.

7.4.2.1 Overview

I designed this experiment to allow visualization via MA plots of 1) systematic bias in un-normalized data and 2) the effect on systematic bias after applying PIN and common normalization methods.

MA Plots

While MA plots are described in detail in Appendix C.2.2, I briefly describe them here. MA plots are similar to ratio vs. intensity plots. However, in MA plots, the ratio versus intensity ordinate system is re-scaled ($\log_2$ transformed) and rotated clockwise in the x versus y coordinate system [151]. Then plotting a locally weighted regression line (LOESS) allows the user to observe linear and nonlinear trends resulting from biases [160] as deviations from a flat line located at $y = 0$. These trends are due to the dependency of the ratio of abundances for a peptide signal intensity, from both runs rather than just one run [117].

MA plots traditionally plot two runs. However, the data sets used to evaluate PIN contain more than two runs. Rather than producing an MA plot for each pairwise combination of runs within a data set, I arbitrarily selected one run as the reference run

and plot MA plots for each other run against the reference run. The resulting MA plots thus have multiple LOESS regression lines, one for each reference run - non-reference run pair.

Data Sets

I demonstrate systematic bias using multiple data sets. The most obvious choice to evaluate the effect of normalization methods on systematic bias is the Serial Dilution data set. This data set has known system biases resulting from loading amount differences. However, Instrument Variability, Sample Variability, and CPTAC Study 6 data sets are also susceptible to systematic bias. Therefore, I show results from each of these four data sets.

Normalization Methods

While Kultima et al. describe ten normalization methods, I present here from normalizing data data sets using PIN and the top five performing common normalization methods. The five common normalization methods are regression, LOESS, quantile, reference run, and median scale (see Appendix C.2.3 for detailed descriptions of these normalization methods).

7.4.2.2 Results (B1 - B4)

The sub-section provides MA plots for Instrument Variability, Sample Variability, CPTAC Study 6, and Serial Dilution data sets. For each data set, I show the MA plots for 1) un-normalized data, 2) data normalized by median scale, and 3) data normalized by PIN. Median intensity normalization is representative of the other normalization methods.

Instrument Variability (B2)

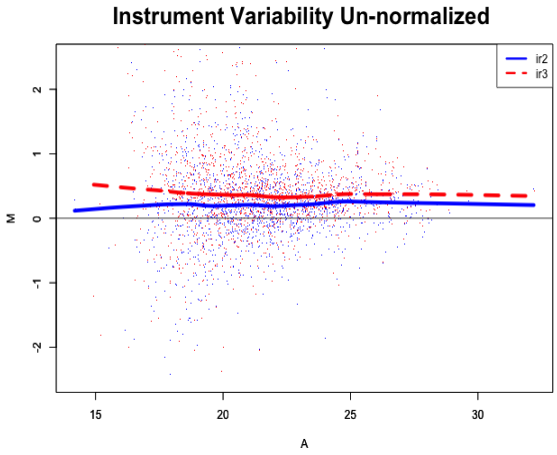
Figure 7.4 depicts three MA plots. I constructed the first MA plot from the three runs

within the Instrument Variability data set. I arbitrarily selected run identifier *ir1* as the reference run. The solid LOESS regression lines were generated from run *ir2*, and the dashed LOESS regression lines were generated from run *ir3*. In Figure 7.4a, note that the LOESS regression lines drawn for the un-normalized MA plot lie above the horizontal flat line positioned at 0 on the y-axis. This indicates that systematic bias exists between runs *ir1* and *ir2* as well as runs *ir1* and ir3. Furthermore, note that the two LOESS regression lines are distinct, that is, they do not overlay one another. I conclude from this observation that the systematic biases in *ir2* and *ir3* are unequal. Figure 7.4b shows the MA plot after the data were normalized using median scale. Note that the solid LOESS regression line generated from run *ir2* and the dashed LOESS regression line generated from run *ir3* have moved down and are now, for the most part, positioned on or very near the horizontal flat line positioned at 0 on the y-axis. However, also note that the lines diverge in different directions on the left end of the plot. From these observations, I conclude that median scale normalization performs well to mitigate the systematic bias in the Instrument Variability data set, with only a small amount of systematic bias remaining.

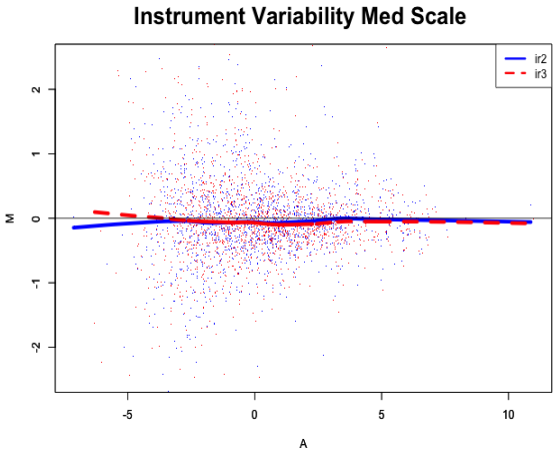
Figure 7.4c shows the MA plot after the data were normalized using PIN. Again, note that the solid LOESS regression line generated from run *ir2* and the dashed LOESS regression line generated from run *ir3* have moved down and are now, for the most part, positioned on or very near the horizontal flat line positioned at 0 on the y-axis. However, also note that the lines dip below the horizontal flat line on the left end of the plot. Unlike median scale, they both change in the same direction, making the two regression lines nearly indistinguishable. From these observations, I conclude that PIN also performs well to mitigate the systematic bias in the Instrument Variability data set, with only a small amount of systematic bias remaining.

Sample Variability (B3)

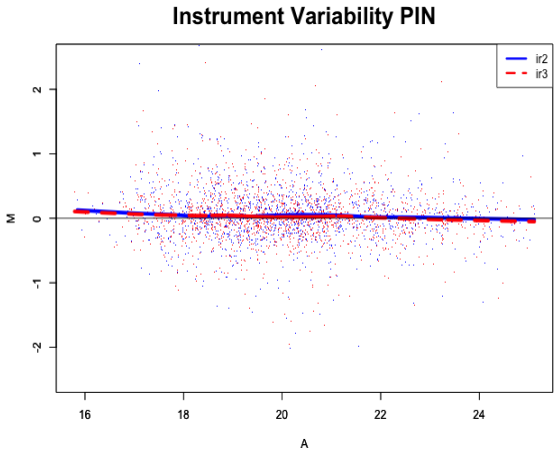
Figure 7.4b shows the MA plot after the data were normalized using median scale.



(a) Un-normalized Data



(b) Median Scale Normalized Data



(c) PIN Normalized Data

Figure 7.4: Minus vs. Average Plots for Instrument Variability dataset

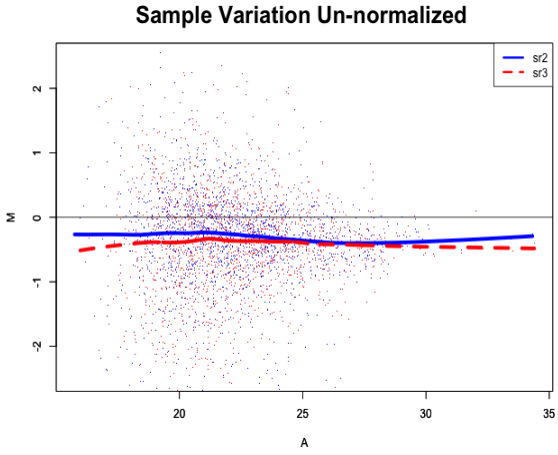
Note that the solid LOESS regression line generated from run *sr2* and the dashed LOESS regression line generated from run *sr3* have moved up and are now, for the most part, positioned on or very near the horizontal flat line positioned at 0 on the y-axis. However, also note that the lines diverge in different directions on the left end of the plot. From these observations, I conclude that median scale normalization performs well to mitigate the systematic bias in the Sample Variability data set, with only a small amount of systematic bias remaining.

Figure 7.5c shows the MA plot after the data were normalized using PIN. Note that, again, the solid LOESS regression line generated from run *sr2* and the dashed LOESS regression line generated from run *sr3* have moved up and are now, for the most part, positioned on or very near the horizontal flat line positioned at 0 on the y-axis. However, also note that the lines dip below the horizontal flat line on the left end of the plot. Unlike median scale, they both dip in the same direction and are distinct. From these observations, I conclude that PIN also performs well to mitigate the systematic bias in the Sample Variability data set, with only a small amount of systematic bias remaining.

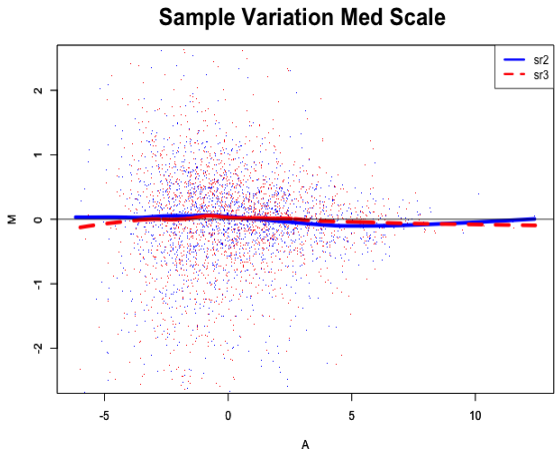
Serial Dilution (B1)

Figure 7.6 depicts three MA plots. The first MA plot was constructed from the six runs within the Serial Dilution data set. I arbitrarily selected run identifier *ssd_0_5* as the reference run. In Figure 7.6a, note that LOESS regression lines drawn for the un-normalized MA generally follow the same trends. However, we also see more than one trend within each LOESS regression line, that is, they first start trending down, then increase, and then tend to flatten out. Furthermore, note that the two LOESS regression lines are distinct, that is, they do not overlay one another. I conclude from this observation that the systematic biases in the runs are unequal and non-linear.

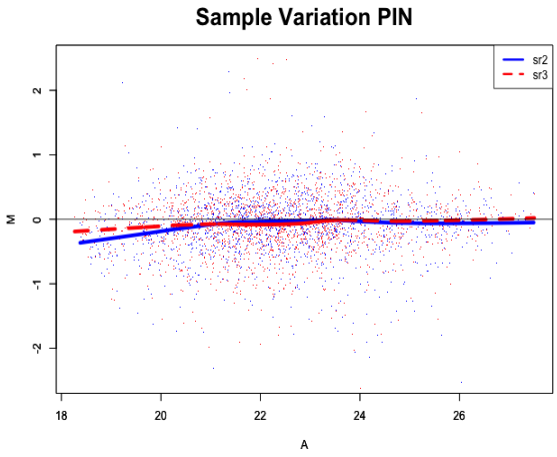
Figure 7.6b shows the MA plot after the data were normalized using median scale. Note that the solid LOESS regression lines generated from the runs have moved up.



(a)



(b)



(c)

Figure 7.5: Minus vs. Average Plots for Sample Variability data set- a) Un-normalized data b) Normalized by median scale c) Normalized by PIN

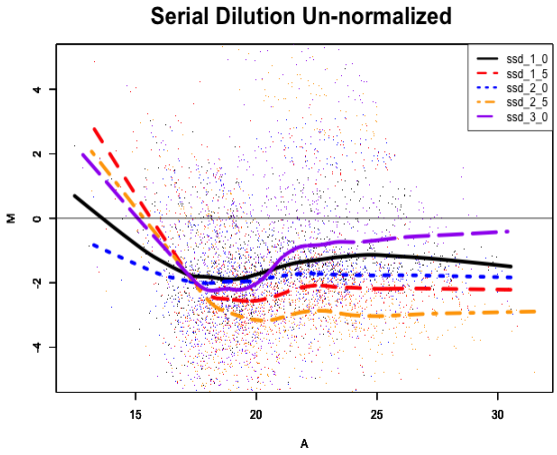
Furthermore, the portions that were below the horizontal flat line positioned at 0 on the y-axis are now centered slightly above and below the horizontal flat line positioned at 0 on the y-axis. However, also note that the lines on the left end of the plot also shift up, but are now located above the horizontal flat line positioned at 0 on the y-axis. Furthermore, note that the LOESS regression lines while they converge slightly continue to have the same trends within the LOESS regression lines. From these observations, I conclude that median scale normalization mitigates the systematic bias slightly in the Serial Dilution data set. However, because the systemic bias is non-linear, much of the systematic bias remains.

Figure 7.6c shows the MA plot after the data were normalized using PIN. Note that the LOESS regression lines on the left end of the plot have shifted up and now lie positioned on or very near the horizontal flat line positioned at 0 on the y-axis. However, also note that the lines remain below the horizontal flat line on the left end of the plot. Unlike median scale, the LOESS regression lines converge, making the regression lines nearly indistinguishable. From these observations, I conclude that PIN performs well to mitigate the systematic bias in the Serial Dilution data set, with only a small amount of systematic bias remaining. Furthermore, I conclude that PIN outperforms median scale normalization in making systematic bias consistent between runs.

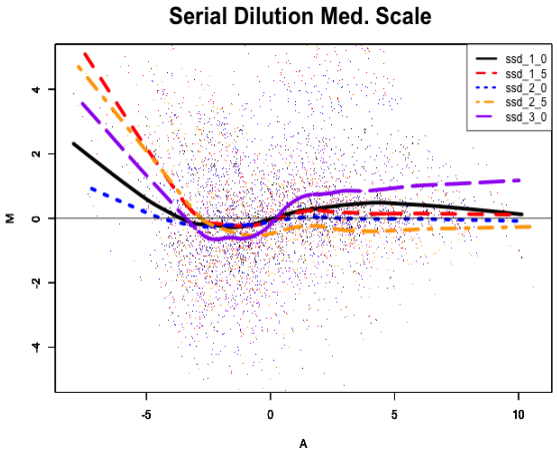
CPTAC Study 6 (B4)

Figure 7.7 depicts three MA plots. The first MA plot was constructed from the six runs within the Serial Dilution data set. I arbitrarily selected run identifier *C_Rep1* as the reference run.

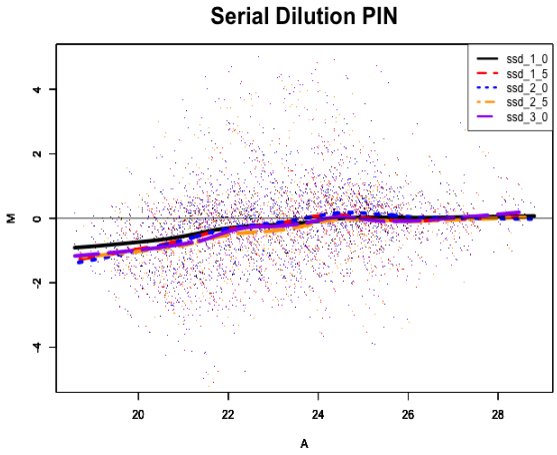
In Figure 7.7a, note that LOESS regression lines drawn for the un-normalized MA plot lie slightly below and slightly above the horizontal flat line positioned at 0 on the y-axis. This indicates that systematic bias exists between runs. Furthermore, note that the LOESS regression lines are distinct, that is, they do not overlay one another. I conclude from this observation that the systematic biases are unequal. However, because



(a) Un-normalized Data



(b) Median Scale Normalized Data



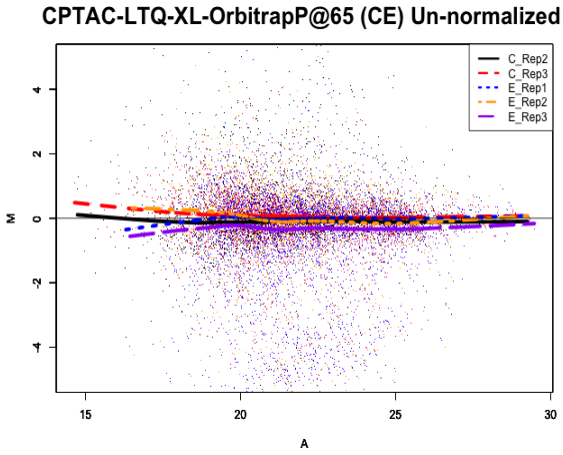
(c) PIN Normalized Data

Figure 7.6: Minus vs. Average Plots for Experiment 3: Serial Dilution

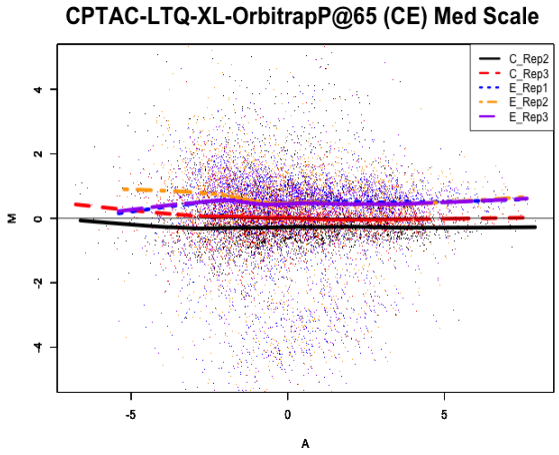
the amount of UPS1 standard proteins added to each yeast background differed for each sample analyzed, this is not unexpected.

Figure 7.7b shows the MA plot after the data were normalized using median scale. Note that the regression lines have diverged rather than converged around the horizontal flat line positioned at 0 on the y-axis. From this observation I conclude that median scale normalization injects systematic bias into the CPTAC Study 6 data set.

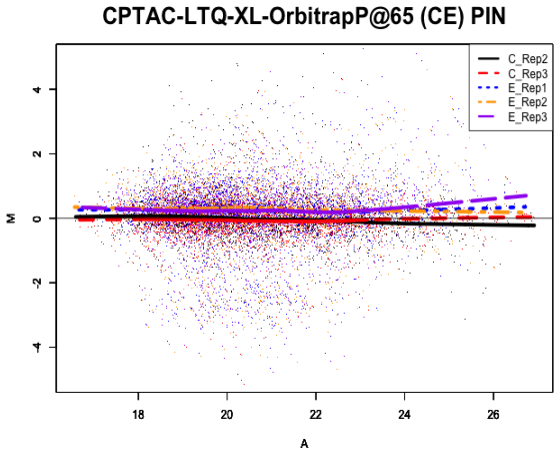
Figure 7.7c shows the MA plot after the data were normalized using PIN. Note that the LOESS regression lines on the left end of the plot now lie positioned on or very near the horizontal flat line positioned at 0 on the y-axis. Also note that the regression lines remain distinct, but more closely follow the same trends. I conclude from these observations, that PIN mitigates systematic bias, but still reflects proportionality differences due to the differing amounts of UPS1 standard proteins added to each yeast background differed for each sample analyzed.



(a) Un-normalized Data



(b) Median Scale Normalized Data



(c) PIN Normalized Data

Figure 7.7: Minus vs. Average Plots for CPTAC Study 6 Data Set

7.4.3 Complex Variability

Complex variability within HPLC-ESI-MS/MS runs is not well documented (see Chapter 3). However, it does exist, and as described by Karpievitch et al. global normalization methods cannot capture complex bias trends like those commonly seen in high-throughput proteomic experiments [175]. Therefore, I sought to show an example of complex variability and evaluate PIN's ability to mitigate complex variability vis-a-vis common normalization methods.

7.4.3.1 Overview

I use the CPTAC Study (6) data set to demonstrate the complex variability. The CPTAC Study 6 data set is described in Section 7.3.4 and its generation detailed in Appendix C.3.2. I chose this data set because it is a reference set, and, despite the effort to precisely follow standard operating procedures, it contains complex variability due to electrospray instability (see Figure 5.2).

7.4.3.2 Results (C4)

Figure 7.8 depicts three XC plots. In Figure 7.8a shows the LOESS regression lines for the three CPTAC Study 6 Experiment C replicate runs. The LOESS regression line for second technical replicate run, *C_Rep2*, shows a distinct trough in the left half of the plot.

Figure 7.8b, shows the LOESS regression line for the three CPTAC Study 6C replicate runs after median scale normalization. When examining this figure, I made two notable observations. First, note that the trough shown in the un-normalized data remains in the median scale normalized data. Second, note that median scale adversely effects portions of the LOESS regression line. The two peaks present in replicate 2's LOESS regression line in Figure 7.8b were very near the other two replicates' LOESS regression lines; the trough was the anomaly. After median scale normalization, these

two peaks diverge from the other replicates and become part of the anomaly. Therefore, peptide signals detected around these two peak's retention times have their intensities distorted, perhaps obscuring true biological variation.

Figure 7.8c shows the LOESS regression line for the three CPTAC Study 6 Experiment C replicate runs after normalization by PIN. Note that the replicate 2's trough has disappeared. Furthermore, note that the LOESS regression lines for each of the three replicates now nearly lie on top of one another.

From these observations, I draw two conclusions. First, when complex variability is present, global normalization methods do not perform well. In fact, they can inject bias where none existed before. Second, PIN performs well in mitigating complex variability.

7.4.4 Internal Standard (Spike-in)

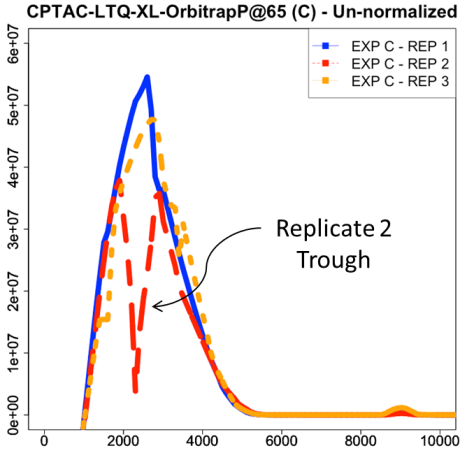
Researchers commonly use a housekeeping or spiked in protein to normalize HPLC-ESI-MS/MS data sets. Therefore, I sought to evaluate PIN vis-a-vis a spiked in protein.

7.4.4.1 Overview

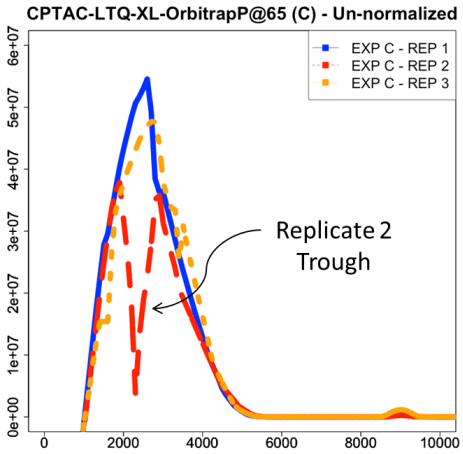
I use the Serial Dilution (1) data set to compare PIN results to spike-in normalization results. Recall that the Serial Dilution data set is described in Section 7.3.1. When preparing the Serial Dilution samples, we added a constant amount of bradykinin to each aliquot. Thus, I could use a detected bradykinin peptide (m/z = 452.73) intensity as a scaling factor for normalizing each data set.

7.4.4.2 Results (D1)

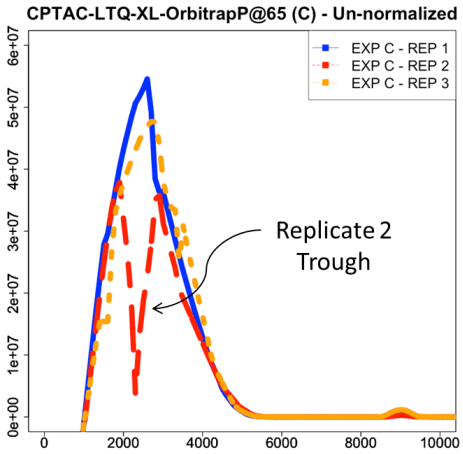
The sub-section provides two four-panel figures for evaluating PIN vis-a-vis spike in normalization. Figure 7.9 shows XCs for each of the six Serial Dilution runs and Figure 7.10 contains bar charts for a single representative peptide in each of the Serial Dilution runs.



(a) Un-normalized Data



(b) Median Scale Normalized Data



(c) PIN Normalized Data

Figure 7.8: CPTAC Study 6 Experiment C - Complex Variability Normalization Results

Un-normalized Data

Figures 7.9a and 7.10a plot data from the un-normalized Serial Dilution data set. In Figure 7.9a, I plot a peptide signal XC for each of the six Serial Dilution runs. Similar to the description of regression lines described in Section 7.4.1, I expected to see monotonically increasing XCs corresponding to the increasing amounts of salivary endogenous peptides in each run. The four grey XCs follow expectations. However, note that two of the six runs did follow expectations. The two annotated black lines, 2.0μg and 3.0μg, appear to contain systematic bias that resulted in peptide signal intensities lower than expected. Note a similar pattern in Figure 7.10a's bar chart. Again, the single peptide signal's intensities for two runs do not meet expectations. Furthermore, I plot a linear regression line, its equation, and the R^2 value. Note that the slope is very large and R^2 = 0.81. While this R^2 indicates a positive correlation, the correlation is not very strong (see Appendix C).

Internal Standard (Spike-in) Normalized Data

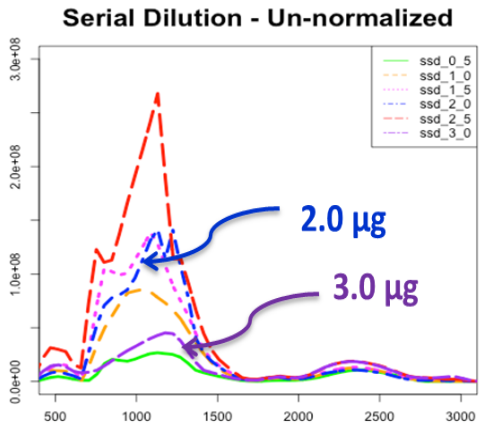
Figures 7.9b and 7.10b plot data from the Serial Dilution data set after scaling by the bradykinin peptide signal intensity. In Figure 7.9b, I plot a peptide signal XC for each of the six Serial Dilution runs. As described in Chapter 3, spike-in normalization can be used as calibration for accurate absolute protein quantification. Therefore, I would expect spike-in normalization to correct the data so that the XCs monotonically increased as amounts of salivary endogenous peptides increased in each run. Unfortunately, the two annotated black lines, 2.0μg and 3.0μg, appear to continue to have systematic bias similar to that depicted in the un-normalized data in Figure 7.9b. Unlike the XCs, I do not observe a similar patterned bar chart Figure 7.10a and Figure 7.10b. Here, the bars are monotonically increasing as expected. Furthermore, when I plot a linear regression line, its equation, and the R^2 value, I observe that the slope decreases and improves to R^2 = 0.98. This R^2 indicates an extremely high correlation between peptide signal intensity and loading amount.

PIN Normalized Data

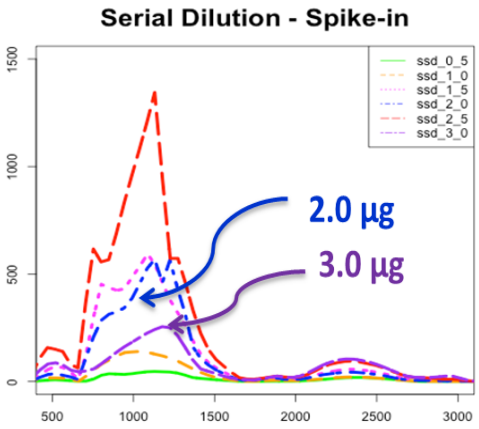
Figures 7.9c and 7.10c plot data from the Serial Dilution data set after normalizing with PIN. In Figure 7.9c, I plot a peptide signal XC for each of the six Serial Dilution runs. As described in Chapter 5, PIN is designed for detecting biological variation. Furthermore, the salivary peptides were pipetted from the same sample. Therefore, the goal should not be to achieve monotonically increasing XCs. Instead, the goal should be XCs overlaying one another. Figure 7.9c shows that PIN performs well to mitigate systematic bias. The black XCs, 2.0μg and 3.0μg, now lie very close to the other grey XCs. Note that a similar trend in Figure 7.10c. Here, the bars are not monotonically increasing. Furthermore, when I plot a linear regression line, its equation, and the R^2 value, I observe that while R^2 decreases to 0.81, the slope decreases to 0.01. Because the slope is nearly zero, I conclude from PIN normalization removes nearly all the systematic bias from loading amount differences has been removed.

PIN Normalized Data - Scaled by Loading Amount

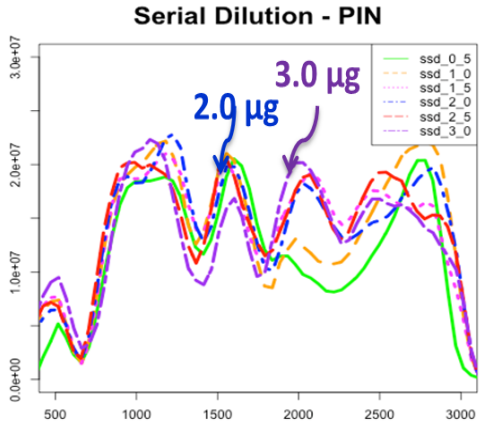
Figures 7.9d and 7.10d plot data from the Serial Dilution data set after normalizing with PIN and then scaling by the original sample loading amounts. Here, I observe that the XCs are monotonically increasing, corresponding to original loading amounts. Furthermore, the black XCs, 2.0μg and 3.0μg, are now in the correct place relative to the grey XCs. Note a similar trend in Figure 7.10c. Here, the bars are not monotonically increasing. Furthermore, when I plot a linear regression line, its equation, and the R^2 value, I observe that R^2 increases to 0.99. From these observations, I conclude that if I know the anticipated loading amounts, I can scale the PIN normalization results by those loading amounts and accurately compute absolute abundance.



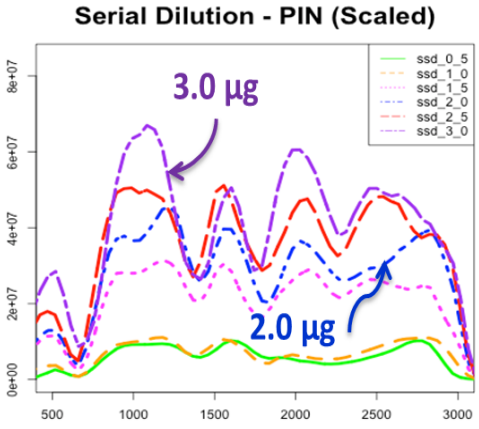
(a) Un-normalized Data



(b) Normalized by bradykinin peptide signal intensity



(c) PIN Normalized Data



(d) PIN Normalized Data Scaled by Loading Amt.

Figure 7.9: Serial Dilution Spike-in Example - XCs

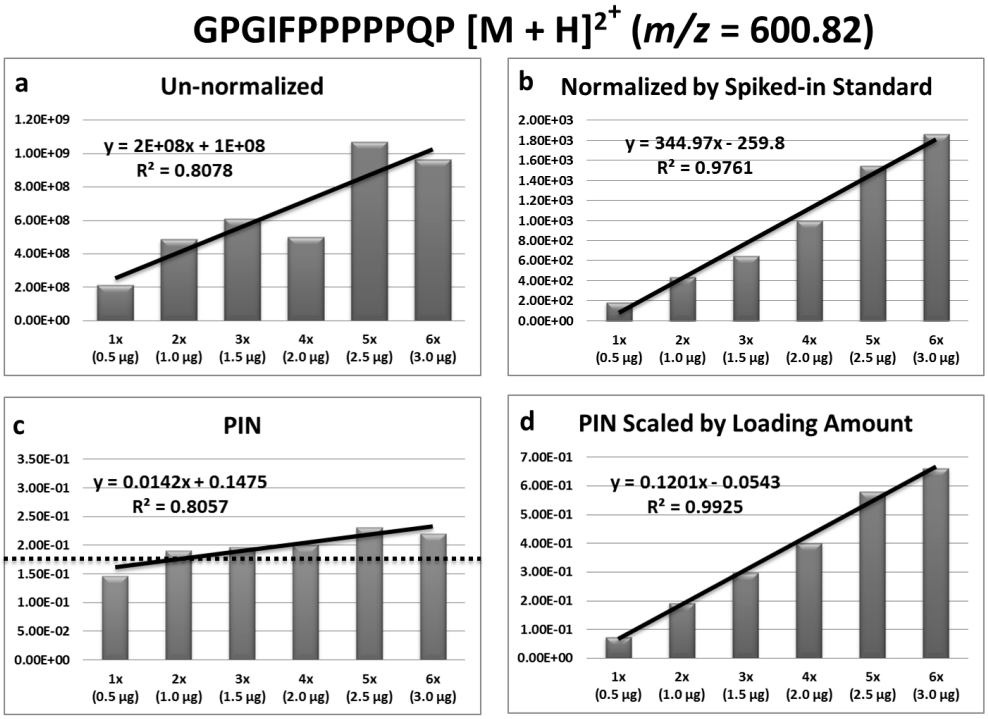


Figure 7.10: Representative peptide sequence GPGIFPPPPPQP (m/z = 600.82^{2+}) in the Serial Dilution data set a) Un-normalized data with regression line b) Normalized data with regression line c) PIN normalized data with regression line (solid) vs. ideal regression line (dashed) d) PIN normalized data scaled by original loading amounts

7.4.5 Repeatability

I conducted repeatability experiments to compare PIN to normalization methods described by Callister et al. and Kultima et al., the two most relevant studies to the work presented in this dissertation. The following two sections provide 1) an overview of the experiment and methods, and 2) the results of the experiment.

7.4.5.1 Overview

I designed this experiment to evaluate PIN's repeatability improvement vis a vis common normalization methods. As described in Chapter 3, and in accordance with Callister et al. and Kultima et al., I use CV and PEV as a measurement of variance (see Appendix C for detailed descriptions of these normalization methods).

Data Sets

I demonstrate PINs ability to improve repeatability using multiple data sets. To do so, I use four data sets. Instrument Variability, Sample Variability, Serial Dilution and CPTAC Study 6.

Normalization Methods

While Kultima et al. describe ten normalization methods, I present results here from normalizing data using PIN and the top five performing common normalization methods. The five common normalization methods are regression, LOESS, quantile, reference run, and median scale (see Appendix C for detailed descriptions of these normalization methods).

7.4.5.2 Results (E1 - E4)

I analyzed the data using existing normalization methods and PIN. Table 7.7 summarizes the reduction in CVs and PEVs for each data set after normalization. Then, for each

data set, I list CVs and PEVs in a bar chart, CVs and PEVs in a tabular format, and reduction in CVs and PEVs in a tabular format. In the bar charts, the darkest column on the left represents repeatability measurements for un-normalized data; the middle five bars represent repeatability measurements after normalizing the data with existing methods; the light bar on the right represents repeatability measurements after normalizing with PIN.

Instrument Variability

Figure 7.11 and Table 7.8 show the CVs and PEVs for the Instrument Variability data set before and after normalization. Here, I see that the five common normalization methods perform comparably. Common normalization methods achieve CVs $\simeq$ 17 (-19 %) and PEVs $\simeq$ 27 (-13 %). However, PIN generates normalized data with a CV $\simeq$ 0.11 (-46%) and a PEV $\simeq$ 0.08 (-73 %). I conclude from this observation that PIN outperforms common normalization methods in improving instrument variability.

Sample Variability

Figure 7.12 and Table 7.9 show the CVs and PEVs for the Sample Variability data set before and after normalization. Note that the five common normalization methods perform comparably. Common normalization methods achieve CVs $\simeq$ 26 (-8%) and PEVs $\simeq$ 46 (-11%). However, PIN generates normalized data with a CV $\simeq$ 0.17 (-41%) and a PEV $\simeq$ 0.15 (-71%). I conclude from this observation that PIN outperforms common normalization methods in improving sample variabilty.

Serial Dilution

Figure 7.13 and Table 7.10 show the CVs and PEVs for the Serial Dilution data set before and after normalization. Note that the five common normalization methods perform comparably. Common normalization methods achieve CVs $\simeq$ 0.49 to 0.55 (-20 to 29%) and PEVs $\simeq$ 1.25 to 1.40 (-32 to 39%). However, PIN generates normalized data with a

CV $\simeq$ 0.31 (-55%) and a PEV $\simeq$ 0.50 (-75%). I conclude from this observation that PIN outperforms common normalization methods in improving systematic bias from loading amount differences.

CPTAC Study 6

Figure 7.14 and Table 7.11 show the CVs and PEVs for the CPTAC Study 6 data set before and after normalization. Note that the five common normalization methods perform comparably. Common normalization methods achieve CVs $\simeq$ 0.49 to 0.55 (-1 to +24%) and PEVs $\simeq$ 1.25 to 1.40 (-4 to -11%). However, PIN generates normalized data with a CV $\simeq$ 0.15 (-19%) and a PEV $\simeq$ 0.28 (-61%).

CV Reduction

Experiment	Regression	Loess	Quantile	Reference	Median Scale	PIN
Instrument Variability	0.19	0.19	0.18	0.2	0.19	**0.46**
Sample Variability	0.08	0.08	0.08	0.08	0.08	**0.41**
Serial Dilution	0.21	0.29	0.21	0.20	0.24	**0.55**
CPTAC C vs E	-0.23	-0.03	-0.22	0.01	-0.24	**0.19**

PEV Reduction

Experiment	Regression	Loess	Quantile	Reference	Median Scale	PIN
Instrument Variability	0.13	0.13	0.14	0.13	0.13	**0.73**
Sample Variability	0.11	0.11	0.12	0.11	0.11	**0.71**
Serial Dilution	0.35	0.39	0.32	0.32	0.32	**0.75**
CPTAC C vs E	0.11	0.08	0.13	0.04	0.11	**0.61**

Table 7.7: Reduction in Pooled Estimate of Variance - PIN vs. Existing

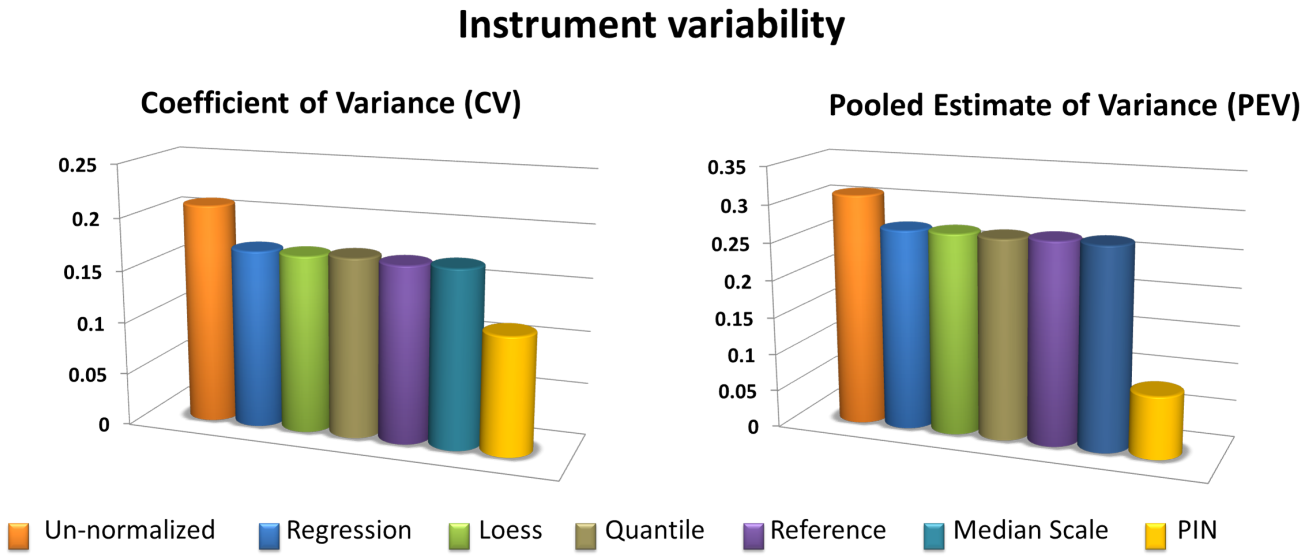


Figure 7.11: Repeatability Measurements PIN vs. Common Normalization Methods - Instrument Variability

	Un-norm.	**Regress.**	**Loess**	**Quantile**	**Ref.**	**Med. Scale**	**PIN**
CV	0.21	0.17	0.17	0.17	0.17	0.17	**0.11**
PEV	0.31	0.27	0.27	0.27	0.27	0.27	**0.08**

Table 7.8: Variance Measurements PIN vs. Common Normalization Methods - Instrument Variability

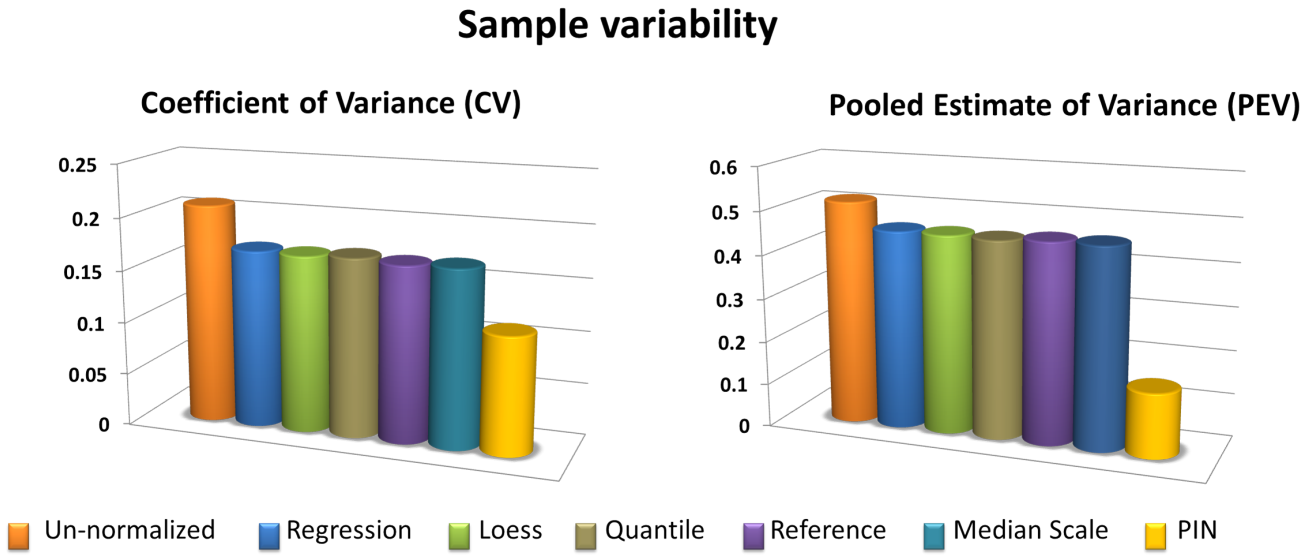


Figure 7.12: Variance Measurements PIN vs. Common Normalization Methods - Sample Variability

	Un-norm.	**Regress.**	**Loess**	**Quantile**	**Ref.**	**Med. Scale**	**PIN**
CV	0.28	0.26	0.26	0.26	0.26	0.26	0.17
PEV	0.52	0.46	0.46	0.45	0.46	0.46	**0.15**

Table 7.9: Variance Measurements PIN vs. Common Normalization Methods - Sample Variability

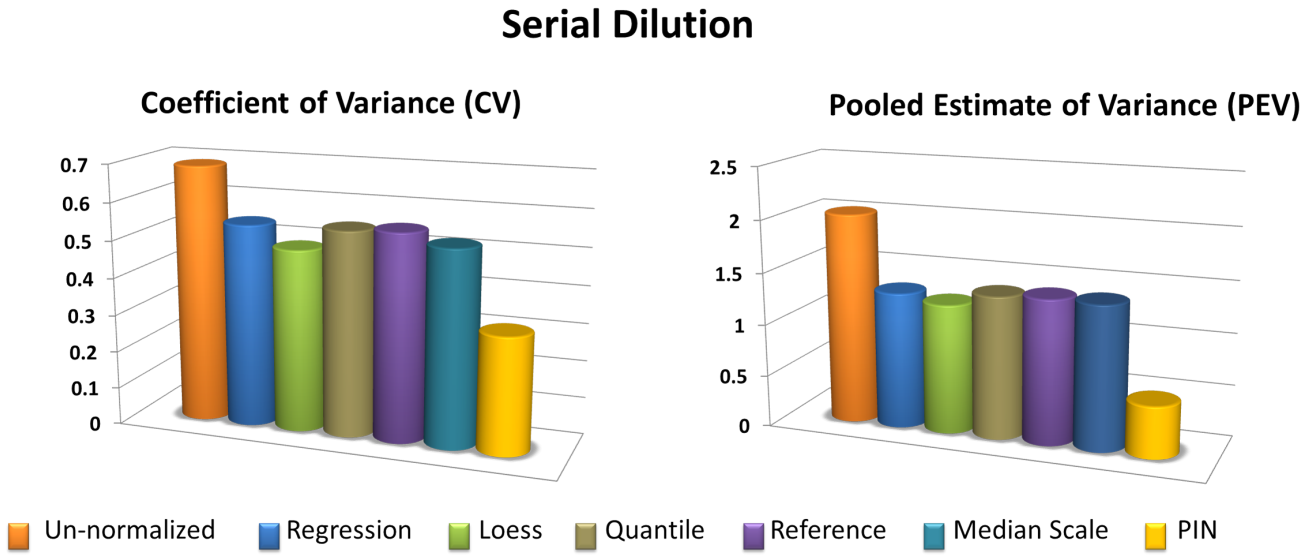


Figure 7.13: Variance Measurements PIN vs. Common Normalization Methods - Serial Dilution

	Un-norm.	**Regress.**	**Loess**	**Quantile**	**Ref.**	**Med. Scale**	**PIN**
CV	0.69	0.54	0.49	0.55	0.55	0.52	**0.31**
PEV	2.03	1.32	1.25	1.37	1.40	1.39	**0.50**

Table 7.10: Variance Measurements PIN vs. Common Normalization Methods - Serial Dilution

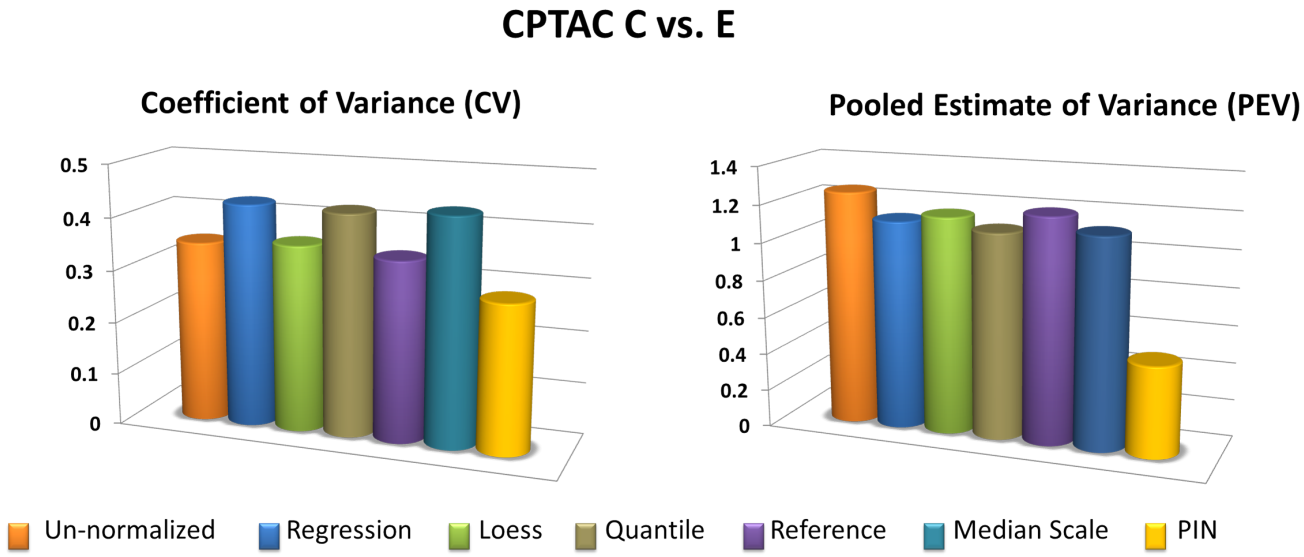


Figure 7.14: Variance Measurements PIN vs. Common Normalization Methods - CPTAC Study 6

	Un-norm.	**Regress.**	**Loess**	**Quantile**	**Ref.**	**Med. Scale**	**PIN**
CV	0.52	0.46	⌋ 0.46	0.46	0.46	0.46	**0.15**
PEV	0.35	0.43	0.36	0.42	0.34	0.43	**0.28**

Table 7.11: Variance Measurements PIN vs. Common Normalization Methods - CPTAC Study 6

7.4.6 Overfitting

With any normalization method, researchers must consider overfitting. As described in Chapter 3, overfitting scales the data to the point the true biological variation obscured. Therefore, I sought to determine whether PIN was overfitting data. I reasoned that if PIN could detect biological variation in a standard reference data set, then PIN was not overfitting data.

7.4.6.1 Overview

I use the CPTAC Study 6 (4) data set to determine if PIN is overfitting data. As in previous experiments, I use the C vs. E portions of the data set. Section 7.3.4 provides an overview of the CPTAC Study 6 data set and Appendix C.1.2 describes in detail the methods for conducting this experiment. Briefly, I first computed a t-test p-value for each m/z pair in the un-normalized and PIN normalized RIPPER .csv reports. I then upload the updated reports, as well as the corresponding Scaffold identification reports to Oracle 11g where I use SQL to assign protein and peptide sequences to the m/z intensity pairs. Finally, I run a series of queries to extract the data included in the following section.

7.4.6.2 Results (F4)

This sub-section describes the results from the overfitting experiment. Table 7.12 lists the number of proteins and peptides differences detected with statistical significance. The first column lists the number of UPS1 proteins(42) with at least 1 peptide difference detected with statistical significant.

7.4.7 Biomarker Discovery (Preliminary)

This section describes preliminary results from applying PIN to a salivary endogenous peptide biomarker discovery study for oral cancer progression. I first provide the context

	UPS Proteins	Yeast Proteins	UPS Peptides	Yeast Peptides	UPS %↑ Peptides	Yeast %↑ Peptides
raw	42	371	208	794	0.97	0.87
PIN	42	368	202	751	0.92	0.33

Table 7.12: CPTAC Study 6 C vs. E Statistical Differences

of oral cancer (why oral cancer is important to study). I then provide an overview of the experiment, preliminary results, and a discussion of the preliminary results.

7.4.7.1 Overview

I am part of an interdisciplinary research team whose long range goal is to discover a panel of proteins and/or peptides to serve as a basis for a clinical assay capable of detecting early stage oral cancer. In line with my ongoing research, I sought to test the following null hypothesis: endogenous peptide abundances do not change between OPML and OSCC populations.

We collected saliva from 18 OPML patients and 19 OSCC patients (see Appendix C.5). After applying PIN to the resulting data, Dr. Koopmeiners, our research team's collaborating biostatistician, employed a recently developed combinatorial-based statistical analysis algorithm to identify a panel of peptide signals that distinguish healthy and disease state (see Appendix C.5). I then investigated a single peptide signal which was present in each of the final predictive models to investigate its biological relevance. Finally, I computed its p-value (via t-test) for un-normalized and normalized intensities for this single peptide.

7.4.7.2 Results (G5)

The statistical analysis produced three peptide signal candidate biomarker panels. Of the 12190 PIN normalized peptide signals extracted using RIPPER, we found 79 differentially proportional at FDR < 0.05 and 137 differentially proportional at FDR < 0.01

(see Table 7.13).

Alpha	No. Significant (out of 12190)	FDR
0.000033	79	0.05
0.000115	137	0.10

Table 7.13: FDR Estimates for various significance levels - OPML vs. OSCC

We built three final predictive models.

- Final Model 1: Model built using all 12190 peptides as candidates
- Final Model 2: Model built using 137 peptides significant with FDR of 0.10 as candidates
- Final Model 3: Model built using 79 peptides significant with FDR of 0.05 as candidates

Tables 7.14 and 7.15 list the peptide signals in the final three models. Figure 7.15 displays the corresponding ROC plots for the models. The ROC plots show good sensitivity and specificity.

Charge	Peptide m/z	Final Model 1 (AUC = 0.959)	Final Model 2 (AUC = 0.988)	Final Model 3 (AUC = 0.988)
3	560.6104126	x	NA	NA
3	574.5993652	NA	NA	x
3	809.4275513	NA	x	x
3	830.7489014	NA	x	NA
3	842.7472534	NA	NA	x
4	531.0033569	NA	NA	x
4	680.8531494	NA	NA	x
4	743.4451904	x	NA	NA
4	792.9900513	x	NA	NA
4	812.890564	x	NA	NA
2	463.6635437	NA	x	NA
2	463.7323303	x	NA	NA
2	482.7054138	NA	x	x
2	486.775238	x	NA	NA
2	510.7426147	x	NA	NA
2	526.2462769	x	NA	NA
2	537.7817993	x	NA	NA
2	556.2946777	x	NA	NA
2	604.272522	NA	x	NA
2	620.7511597	NA	x	x
2	638.2677002	x	NA	NA
2	640.7568359	x	NA	NA
2	641.2999878	x	x	x
2	364.1983643	NA	x	x
2	666.4089966	x	NA	NA

Table 7.14: List of peptide signals included in final models (Part I)

Charge	Peptide m/z	Final Model 1 (AUC = 0.959)	Final Model 2 (AUC = 0.988)	Final Model 3 (AUC = 0.988)
2	690.8308716	NA	NA	x
2	698.4381714	NA	x	x
2	699.3563232	NA	x	NA
2	715.8849487	NA	x	x
2	722.2884521	NA	NA	x
2	750.8275146	x	NA	NA
2	416.2211609	x	NA	NA
2	858.9049683	x	NA	NA
2	869.3991699	NA	x	x
2	964.4642944	x	NA	NA
2	1223.588745	NA	NA	x
2	1272.789917	x	NA	NA
3	406.5023193	x	NA	NA
3	456.1983337	x	NA	NA
3	528.2607422	NA	NA	x

Table 7.15: List of peptide signals included in final models (Part II)

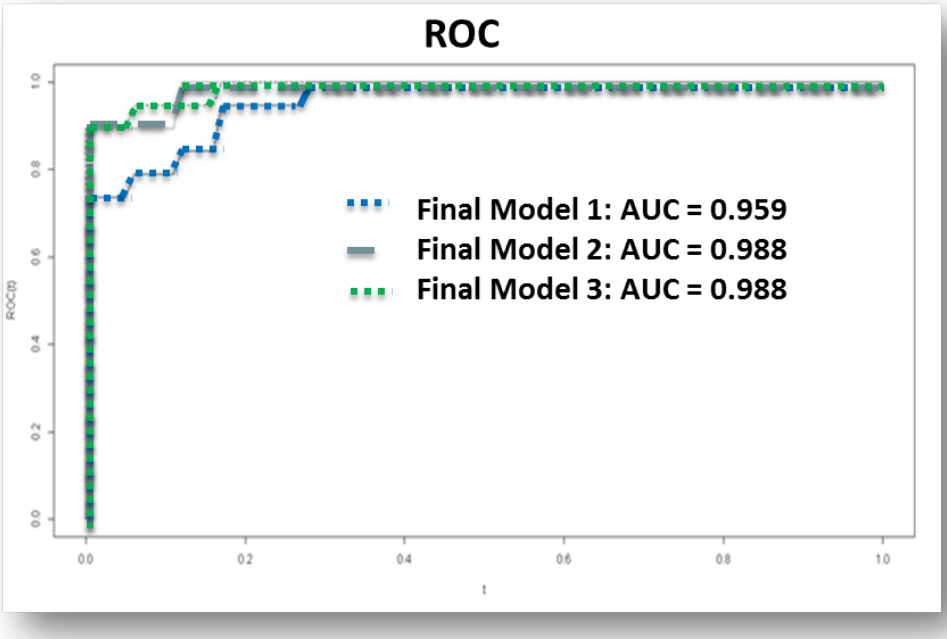


Figure 7.15: OSCC vs. OPML ROC

7.5 Discussion

This section discusses each experiment, providing observations as warranted. My evaluations demonstrate PIN's domination over common normalization methods. The following list summarizes each experiment in my evaluation.

- SN Signal vs. Peptide Signal: I demonstrated that signal stemming from peptides is a subset of all recorded signal above the specified SN threshold. I plotted XCs for recorded signal with intensity greater than the SN threshold and XCs for extracted peptide signals. Expectedly, I observed disparity between the two XCs. Unexpectedly, I also observed the disparity varied over time.

- Systematic bias: I showed via MA plots that PIN outperformed common normalization methods in removing systematic bias (by visual inspection).

- Complex Variability: I provided an example of complex variability in the standard reference data set, CPTAC Study 6. I plotted peptide signal XCs for un-normalized, median scale normalized, an PIN normalized data. By visual inspection, the results demonstrate that PIN outperforms median scale normalization. Furthermore, the results show that median scale normalization actually exaggerates the complex variability.

- Internal Standard (Spike-In): I used the serial dilution data set to show that PIN mitigates systematic bias due to loading amount differences. I divided the experiment into three sub experiments. For each sub-experiment, I compared resulting peptide signal XC plots to un-normalized peptide signal XC plots. Furthermore, I compared linear regression equation coefficients and correlation coefficients (R^2) for a single peptide from resulting normalized intensity scatter plots to un-normalized intensity scatter plots. .

  - Since internal standard is used to compute absolute abundance, I expected the normalized XCs to be monotonically increasing, correlating to their original sample loading amounts. Furthermore, I expected to move the R^2 value closer to one, indicating a higher level of correlation between sample loading amount and measured intensity. For the single peptide signal, after normalization by the internal standard, the R^2 value increased from 0.806 to 0.976 and peptide signal intensities monotonically increased. However, from visually inspecting the peptide signal XCs for the entire run, the overall peptide signal did not monotonically increase.

  - Since PIN is used to reveal biological variation, and samples contain almost no biological variation (they were pipetted from the same mixture with the only difference being the amounts of bradykinin spiked in), I expected the normalized XCs to be nearly indistinct, mitigating systematic bias stemming from loading amount differences. Furthermore, while I noted the R^2 value, I

didn't use it as a measure of success. Here, I expected the slope variable in the linear regression equation from the scatter plots to approach zero. For the single peptide signal, after normalization PIN, note that the slope decreased to 0.01. Furthermore, note that the peptide signals are very close to one another.

  - Since scaling PIN normalized results is akin to calibration, I expected the scaled PIN normalized XCs to be monotonically increasing, correlating to their original sample loading amounts. Furthermore, I expected to move the R^2 value closer to one, indicating a higher level of correlation between sample loading amount and measured intensity. For the single peptide signal, after normalization by the internal standard, the R^2 value increased from 0.806 to 0.993 and peptide signal intensities monotonically increased. Furthermore, by visually inspected the peptide signal XCs for the entire run, the overall peptide signal did monotonically increase.

- Repeatability: I showed PIN's dominance in improving repeatability using four data sets. For each data set, I compared PIN's CVs and PEVs to five common normalization methods' CVs and PEVs. I listed and plotted the CVs and PEVs. Furthermore, I listed the reduction in CVs and PEVs. Note that, in every measure, PIN outperformed common normalization methods. In addition, I observed that the common normalization methods performed similarly in reducing CVs and PEVs. Finally, I unexpectedly observed that all but one of common normalization methods actually increased CVs and PEVs in the CPTAC Study 6C data set. I posit that this is caused by the presence of complex variability in CPTAC Study 6C's coupled with little systematic bias. Recall, common normalization methods are designed to mitigate systematic bias. Therefore, when faced with complex variability, common normalization methods actually induced additional variance. I plan to explore this phenomena further in future work.

- Overfitting: I demonstrated PIN does not overfit data using the CPTAC Study 6C data set. I also considered comparing my results to those reported by Milac et al. in 2012 [147]. Unfortunately, they detected differences between UPS1 spiked in levels to the quality control while I detected differences between UPS1 spike in levels. Their method of rolling up intensity levels, from detected peptide signals to peptides and proteins is different from RIPPER/PIN.

- Biomarker Discovery (Preliminary): Finally, I applied PIN to a biomarker discovery data set (OPML vs. OSCC). After applying sophisticated statistical analysis, Joel Koopmeiners, our research team's biostatistician, produced three peptide signal candidate biomarker panels. I selected a peptide signal present in each of the three panels (641.30), computed the un-normalized and normalized p-value, and plotted the results.

7.6 Summary

In summary, the reported results demonstrate that PIN dominates current normalization methods in reducing systematic bias and complex variability. Furthermore, it does so while, retaining the ability to detect statistically significant biological variation.

Chapter 8

Conclusion

Nothing endures but change. - Heraclitus

8.1 Introduction

Researchers conduct discovery-based studies to find biological variation that not only provides insight into the molecular machinery of disease progression, but accurately inform clinicians about a patient's health status, both current and future. Researchers discover biological variation by conducting large scale comparative studies and detecting differences in the molecular makeup (biomarkers) of healthy and diseased cells. Ideally suited for biomarkers are proteins because their cellular composition (proteome) and their degraded parts, endogenous peptides (peptidome) changes in response to disease by creating new proteins, modifying existing proteins, and degrading proteins into endogenous peptides.

Increasingly, researchers employ high performance liquid chromatography, coupled with electrospray ionization couple with tandem mass spectrometry (HPLC-ESI-MS/MS) for proteomics and peptidomics. For relative quantification of peptides, intensity-based label free relative quantification (iLFRQ) is desirable because it is cost effective and does not limit the number of samples analyzed in a single experiment. Unfortunately, iLFRQ for proteins, and especially peptides, is difficult.

The remainder of this chapter is organized as follows. Section 8.2 summarizes three

challenges that researcher's face in current comparative proteomic workflows. Section 8.3 summarizes my 3 contributions presented in this dissertation, The proportionality paradigm for iLFRQ, Proximity-based Intensity Normalization (PIN), and RIPPER: An iLFRQ Software Framework. Section 8.4 reviews my evaluation. Section 8.5 proposes future work. Section 8.6 discusses how my contributions will impact the proteomics field.

Figure 8.1 serves as a reference in the remaining sections. Panel a shows an adaptation of the Figure 2.2 in Chapter 2. This figure shows a generic current comparative proteomics workflow. In the adaptation, the workflow is now spread out. Panel b shows how my contributions change the generic current comparative proteomics work. Here, steps 1-3 are compressed into a single box.

8.2 Challenges

Researchers employing current comparative proteomics workflows face several challenges. In this section, I highlight three of them. While these challenges remain, researchers must conduct extensive hypothesis-driven experiments to weed out excessive false positives. Worse, real biological variation is missed because researchers do not see false negatives. As a result, researchers can draw incorrect conclusions about biological variability and thus miss key insights.

The Relative Abundance Paradigm

The current relative abundance paradigm for iLFRQ is ill-suited to detect biological variation. This paradigm asks the question, "Are the constituent peptides differentially abundant?" To answer this question, researchers select peptides with intensity fold changes greater than some threshold between two HPLC-ESI-MS/MS runs. Unfortunately, systematic differences, for example, sample amount, distort MS^1 peptide signal measurements and is, therefore, especially problematic for iLFRQ.

In Figure 8.1a, I depict the relative abundance paradigm as a box around the entire

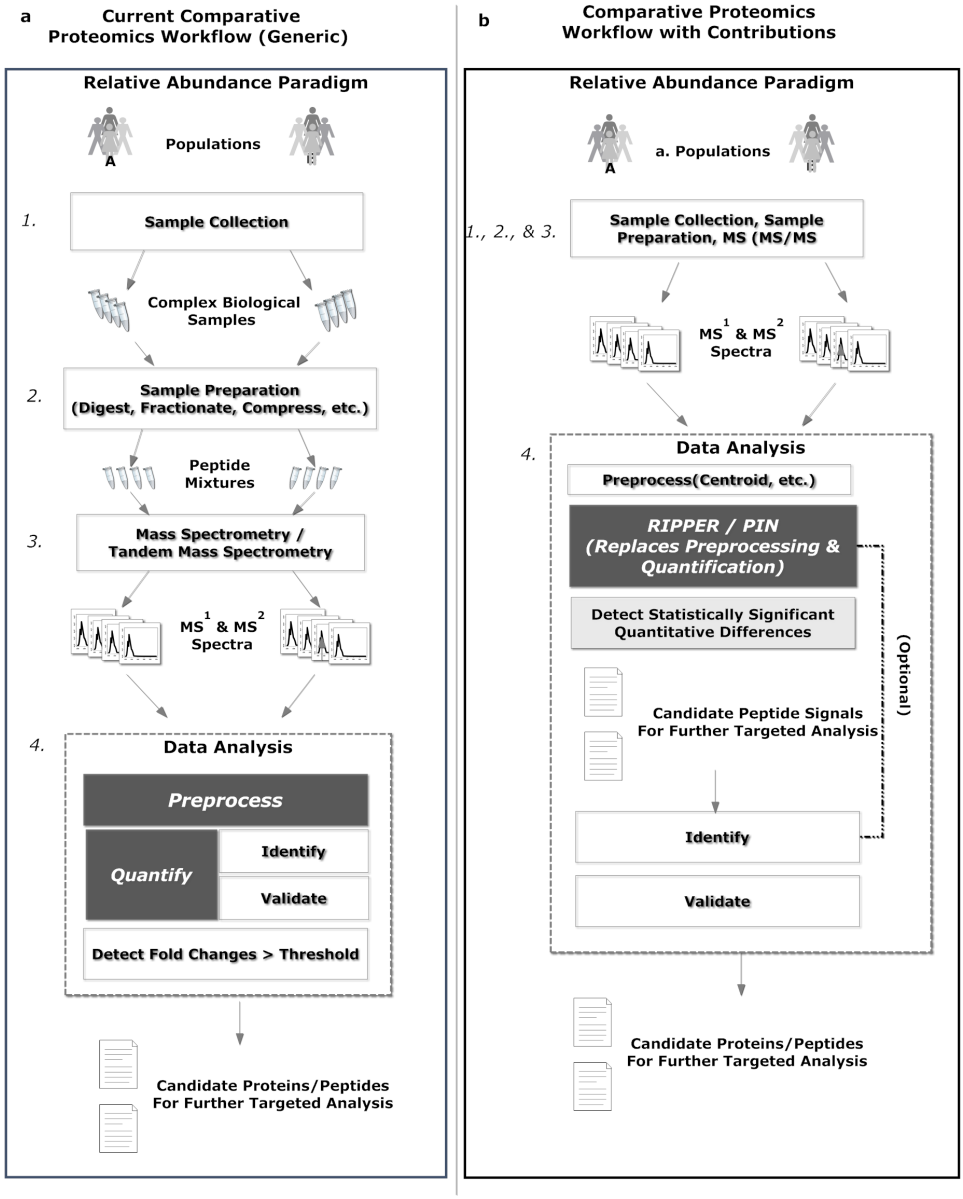


Figure 8.1: Current vs. Comparative Proteomics Workflow With Contributions - Panel a shows an adaptation of the Figure 2.2 in Chapter 2. This figure shows a generic current comparative proteomics workflow. In the adaptation, the workflow is now spread out. Panel b shows how my contributions change the generic current comparative proteomics work. Here, steps 1-3 are compressed into a single box.

comparative proteomics workflow. This is because this paradigm transcends analytical techniques.

Complex Variability

HPLC-ESI-MS/MS analyses produce poorly repeatable and reproducible results, primarily due to extraneous variability. While current normalization methods work well to mitigate global extraneous variability (systematic bias) in HPLC-ESI-MS/MS measurements, they fail to mitigate localized extraneous variability (complex variability in measurements) from transient stochastic events occurring during an HPLC-ESI-MS/MS run. In Figure 8.1a, complex variability can occur in any of the four steps. However, I have observed that the most impactful complex variability occurs within step 3, MS and MS/MS. Common normalization methods generally fall within the preprocessing (dark grey box).

iLFRQ Software Frameworks

Current software frameworks report protein level quantification rather than peptide level quantification. Furthermore, although open source software frameworks purport to be easily extendable, in practice, they are not. This limits researcher's ability to conduct quantitative peptidomics studies.

In Figure 8.1a, I depict software frameworks within the data analysis (dashed box). The software frameworks can include one, many, or all of the data analysis steps.

8.3 Contributions

To address over these challenges, I offer three contributions. My first contribution is the application of the proportionality paradigm for iLFRQ. My second contribution, Proximity-based Intensity Normalization (PIN), embodies the proportionality paradigm; it mitigates complex variability. My third contribution, RIPPER, is a new iLFRQ

software framework. I expect my contributions to change the way researchers analyze HPLC-ESI-MS/MS experimental data. The upshot will, I expect, be reproducibility and repeatability improved, and otherwise falsely reported or missed, statistically significant biological variation discovered.

8.3.1 The Proportionality Paradigm for iLFRQ

This section discusses my 1^{st} contribution, the application of the proportionality paradigm for iLFRQ. First, I compare and contrast the proportionality paradigm to the current relative abundance paradigm. Then, I list the proportionality paradigm's requirements/assumptions and disadvantages/limitations. Finally, I share some personal observations I made while formulating the proportionality paradigm.

In Figure 8.1b, I depict the proportionality paradigm as a box around the entire comparative proteomics workflow. This is because the proportionality paradigm transcends analytical techniques.

The Proportionality Paradigm vs. The Relative Abundance Paradigm

The proportionality paradigm differs from the current relative abundance paradigm. The proportionality paradigm, in contrast to the relative abundance paradigm, asks the question, "Are the constituent peptides differentially proportional?" Researchers answer this question by computing a peptide's proportional abundance and then detecting their statistically significant differences across multiple samples.

Under the current paradigm, researchers reveal biological variation by measuring the relative abundance of corresponding proteins or peptides in two or more samples. Here, relative abundance is the abundance of a protein or peptide in one sample divided by the corresponding abundance in another sample. The result is a ratio (fold change). While the relative abundance paradigm works well with quantitative analytical techniques such as microarray studies, the relative abundance paradigm can fail in the presence of extraneous variability (systematic bias and complex variability stemming from changes in

experimental conditions). Under the proportionality paradigm, researchers reveal biological variation by detecting statistically significant relative proportions as opposed to relative abundance ratios (fold changes) of corresponding proteins or peptides in two or more samples. Here, relative proportion is that of a protein or peptide in a single sample.

Requirements/Assumptions

In order for the proportionality paradigm to work properly, certain assumptions must be met. Here, I highlight four of them.

- The analyzed samples must be comparable. This means that they contain essentially the same proteins or peptides.
- Only a small portion of proteins or peptides in the sample are biological variable, quantitatively.
- There are enough proteins or peptides in the sample that are not biologically variable, quantitatively, such that biologically variable proteins or peptides can be considered outliers as compared to the overall distribution.
- Important biological variation of proteins and peptides in relation to proteins and peptides is biologically relevant.

Disadvantages / Limitations

Of course, the proportionality paradigm has its disadvantages and limitations. Here, I highlight three of them.

- The proportionality paradigm will not quantify protein absolute abundance. It is only applicable for revealing biological variation. Therefore, the proportionality paradigm is not applicable for studies measuring absolute abundances. However,

as shown in Chapter 7.4, if the original loading amounts are known with certainty, the absolute abundance can be computed by scaling the results by the orignal loading amounts.

- The proportionality paradigm cannot be used on samples that don't meet the assumptions. For example, samples that are pre-fractionated violate the assumptions listed in the previous sub-section.

- The proportionality paradigm tends to compress fold changes for proteins and peptides . Therefore, simple fold change threshold for detecting biological variation is no longer applicable. Instead, researchers must use statistical significance to detect outliers in the distribution of measured proportions.

  - Statistical significance tests require at least three technical and biological replicates.
  - Statistical significance tests are computationally more expensive than simple ratio (fold change) calculations.

Observations

Paradigm shifts are rarely fully embraced when introduced. Therefore, it's not surprising that my colleagues pose interesting and challenging questions concerning the proportionality paradigm. First, they challenge the notion that the proportionality paradigm is not just another normalization method. Second, they challenge that the proportionality paradigm rises to the level of a paradigm shift.

I defend the notion that proportionality paradigm is not a normalization method with three arguments.

- Normalization's intent differs from the proportionality paradigm's intent. The normalization's goal is to remove systematic bias. The proportionality paradigms goal is to reveal biological variation by comparing proportions.

- Normalization implies that the amount of sample analyzed in multiple mass spectrometry runs is the same. This is in contrast to the proportionality paradigm where the amount of sample measured is irrelevant.

- Normalization, in the context of HPLC-ESI-MS/MS workflows, can include peak intensities not related to peptide signals. Scaling factors such as total ion current can be used. In the proportionality paradigm, again in the context of HPLC-ESI-MS/MS workflows, peptide signal intensities must be normalized by other peptide signal intensities.

I also defend my position that the proportionality paradigm does rise to the level of a paradigm shift with two arguments.

- As discussed in Chapter 4, the proportionality paradigm requires researchers to shift their fundamental perspective from "one" to "many". Using the relative abundance paradigm, a researcher analyzes a peptide signal's abundance across samples, without regard to sample composition. That is, researchers analyze a peptide signal in isolation. Analyzing the sample as whole, that is, examining its composition using the proportionality paradigm, more closely mirrors systems biology. In vivo, peptides and proteins do not exist in isolation; they form multi-protein complexes and interact with each other, as well as other biomolecules, for example, enzymes and metabolites, thus driving cellular processes.

- I posit that the problem with the relative abundance paradigm is not incorrect answers, the problem is that we are asking the wrong question. I *do not* contend that we should change or replace the relative abundance paradigm. The relative abundance paradigm is valid and has its place in scientific research, where there is minimal extraneous variability. However, I *do* contend that using the relative abundance paradigm to discover biological variation in complex biological studies using HPLC-ESI-MS/MS, for example, biomarker discovery studies, is inappropriate because it asks the wrong question. Thus, we need a new paradigm that asks

the right question, "Are the constituent peptides differentially proportional?"

8.3.2 Proximity-based Intensity Normalization (PIN)

This section discusses my 2^{nd} contribution, Proximity-based Intensity Normalization (PIN). First, I compare and contrast PIN to common normalization methods. However, I do not present results from the evaluation chapter here. Instead, I will summarize results in Section 8.4. After comparing and contrasting PIN to other methods, I list PIN's requirements/assumptions and disadvantages/limitations. Finally, I share some personal observations made while developing PIN.

In Figure 8.1b, I moved PIN out of pre-processing and combined it with RIPPER (my 3^{rd} contribution, a new software framework described in the next sub-section). While PIN is a new method that could be implemented in a software framework other than RIPPER, I combined them in the figure because, now, PIN is implemented within RIPPER.

PIN vs. common normalization methods

PIN is an embodiment of the proportionality paradigm that normalizes a peptide's signal intensity measured via HPLC-ESI-MS/MS by constructing its temporal neighborhood and then computing its relative proportion within that neighborhood. PIN, to my knowledge, is the first normalization method designed specifically for HPLC-ESI-MS/MS workflows. Through my evaluations, I demonstrated that PIN mitigates both systematic bias and complex variability.

PIN is a departure from common normalization methods. In the past, researchers adapted common normalization methods used in HPLC-ESI-MS/MS workflows from genomics microarray studies. As discussed in Chapter 1, the Human Genome Project was phenomenally successful in providing a foundation for genetic biomarker discovery. In the post-genome era, researchers often turn to genomics' analytical tools as inspiration for developing analytical tools in post-genome fields. This is the history of common

normalization methods used today in HPLC-ESI-MS/MS workflows.

While genomics microarray analysis and HPLC-ESI-MS/MS analysis share many characteristics, they are distinctly different. The primary difference related to research for this dissertation is the presence of complex variability in HPLC-ESI-MS/MS chromatographic data from transient stochastic occurring within a HPLC-ESI-MS/MS run. The common normalization methods adapted from genomics microarray studies are designed to mitigate systematic bias; they fail to mitigate complex variability.

PIN employs the proportionality paradigm within a dynamically populated temporal peptide signal neighborhood centered at the retention time of the peptide signal being normalized. The algorithm populates the neighborhood with only proximal peptide signal peaks (the peptide signal's XIC peaks). This means that prominent peaks (those peaks above the SN threshold), but not contributing to a peptide signal, are discarded. This is in fitting with the proportionality paradigm. In the proportionality paradigm, we compute a peptide's proportional abundance using abundances of other peptides. In PIN, we normalize a peptide's signal's intensity by other peptide signal intensities. Including any other chromatographic data not stemming from peptide signal's would violate the proportionality paradigm.

Requirements/Assumptions

As an embodiment of the proportionality paradigm, PIN inherits the requirements and assumptions from the proportionality paradigm. In addition, the following requirements and assumptions must be met for PIN.

- High resolution mass spectrometry analysis with concomitant mass accuracy.
- The peptide elution order is similar for analyzed samples must be similar; PIN requires similar order to form similar neighborhoods.
- Retention times are similar.

Disadvantages / Limitations

Again, as an embodiment of the proportionality paradigm, PIN inherits its disadvantages and limitations. Here, I highlight three additional disadvantages and limitations not inherited from the proportionality paradigm.

- PIN has only been tested with HPLC-ESI-MS/MS workflows employing Thermo-Scientific's Orbitrap type mass spectrometers.
- PIN can't compare samples analyzed using different types of chromatographic systems.
- Window size for neighborhood population must be validated for each type of system.

Observations

Researchers adapted common normalization methods from genome microarray studies for HPLC-ESI-MS/MS workflows. However, these common normalization methods fail in the face of complex variability. As with the relative abundance paradigm, I do not contend that common normalization methods are wrong. They just aren't appropriate for HPLC-ESI-MS/MS workflows where complex variability is inevitable.

The research presented in this section should serve as a cautionary tale for researchers adopting methods from other disciplines. In today's age, simply processing data to achieve some result is insufficient. Researchers need to understand their data and what their results mean. I contend that researchers need to know several things about their data.

- What is the context of the study; what is the research goal?
- What are characteristics of the sample being analyzed?
- Is the workflow susceptibility to extraneous variability?

- How was the data manipulation; how is the data extracted, normalized, and presented?

I contend that researchers need to know several things about their results.

- What do the results mean?
- Are the results statistically significant?
- What is acceptable in the field?
- Are the results relevant?
- In biomedical informatics and computational biology, are they biologically relative?

8.3.3 RIPPER: an iLFRQ Framework

.

The section discusses my 3^{rd} contribution, RIPPER: an iLFRQ framework. First, I compare and contrast RIPPER and other iLFRQ frameworks. Then, I list RIPPER's requirements/assumptions and disadvantages/limitations. Finally, I share some personal observations made while working on RIPPER.

In Figure 8.1b, I combined RIPPER with PIN. While I could implement other normalization methods in RIPPER, now, PIN is the only normalization method implemented in RIPPER.

RIPPER vs. Other iLFRQ Frameworks

A new iLFRQ software framework named RIPPER reports normalized peptide signal intensities rather than protein intensities. Optionally, researchers can match peptide signals to third party software peptide and protein identifications. RIPPER is available from the (https://z.umn.edu/ripper).

RIPPER was born out of frustration with current iLFRQ software frameworks. First, while analyzing data from salivary endogenous peptide experiments, I found that current

iLFRQ software frameworks report normalized intensities at the protein level. Protein level intensities are of little use to a researcher studying endogenous peptides. Second, using SQL, as a proof of concept, I implemented a labor intensive version of PIN (data not shown). However, to provide researchers with easy access to PIN, I wanted to automate it by integrating it into and existing framework. To do so, I experimented with converting the SQL statements to source code and retrofitting it into several open source applications. After several attempts, I found integrating PIN into an existing framework would have required an infeasible amount of modification to existing code. Frustrated, I created a new software framework for iLFRQ, which I named RIPPER.

Requirements/Assumptions

Several requirements and assumptions must be met in order for RIPPER to run properly.

- Requires Java installed.
- The input file is in mzXML standard format with centroid peak data .
- The mzXML extension is exact, including case. This is a quirk in the JRAP software.
- Depending on the mzXML file size and the number of files processed in a single run, RIPPER requires 2+ GB to 6+ GB of random access memory (RAM).

Disadvantages / Limitations

Any software has disadvantages and limitations. Here, I list three for RIPPER.

- RIPER is not well suited for experiments using pre-fractionation, for example, MUDPit experiments.
- RIPPER does not currently support the newest standard format for MS data, mzML

- RIPPER requires MS/MS data.

Observations

Many current open source software frameworks for iLFRQ purport that they are extensible. In theory, they are. In practice, they are not. While they are open source, sometimes they are limited to a particular operating system, have extensive requirements, or simply do not compile. For example, arguably the most popular iLFRQ software framework's downloaded source code (MaxQuant) does not compile. (I have confirmed this by discussing it this with leaders in the field.) Without intimate knowledge of the source code, it would take longer to try to extend the code than to write a new framework from scratch. Fortunately, software frameworks such as msInspect are implemented well, based on sound software engineering practices. Therefore, adapting some of their algorithms for RIPPER was straightforward.

Unfortunately, most literature describing software frameworks lack detailed descriptions of normalization methods. Commonly, researchers will describe normalization as "we normalized the data by median scale." What does that mean? There are numerous ways of performing median scale normalization. Worse, different median scale normalization methods will produce different results.

8.4 Evaluation

I evaluated the application of the proportionality paradigm for iLFRQ, as well as, PIN and RIPPER using datasets from HPLC-ESI-MS/MS analyses of complex peptide mixtures. PIN dominates current normalization methods in reducing systematic bias and complex variability and finds statistically significant biological variation which otherwise is falsely reported or missed.

My evaluation demonstrates PIN's domination over common normalization methods. The remainder of this section summarizes each analysis and provides observations as

warranted.

SN Signal vs. Peptide Signal

I demonstrated that signal stemming from peptides is a subset of all recorded signal above the specified SN threshold.

Systematic bias

I showed via MA plots that PIN outperformed common normalization methods in removing systematic bias (by visual inspection).

Complex Variability

I provided an example of complex variability in the standard reference data set, CPTAC Study 6.

Internal Standard (Spike-In)

I used the serial dilution data set to show that PIN mitigates systematic bias due to loading amount differences. I divided the analysisinto three sub-analyses. For each sub-analyses, I compared resulting peptide signal XC plots to the un-normalized peptide signal XC plots. Furthermore, I compared linear regression equation coefficients and correlation coefficients (R^2) for a single peptide from resulting normalized intensity scatter plots to un-normalized intensity scatter plots.

- For a single peptide signal, after normalization by the internal standard, the R^2 value increased from 0.806 to 0.976, and peptide signal intensities monotonically increased. However, from visually inspecting the peptide signal XCs for the entire run, the overall peptide signal did not monotonically increase.
- For the single peptide signal, after normalization PIN, I see that the slope decreased to 0.01 (goal = 0.0).. Furthermore, I see the peptide signals are very close to one another.

- Scaling PIN normalized results by the original loading amount showed monotonically increasing peptide signal XCs and for the single peptide, R^2 increased to R^2 0.99.

Repeatability

I showed PIN's dominance in improving repeatability using four data sets. I observed that, in every measure, PIN outperformed common normalization methods. Tables 8.1 shows the CV and PEV performance improvements for PIN vs. the mean of the five common normalization methods. It does so for five experiments.

Unexpectedly, I observed that all but one of common normalization methods actually increased CVs and PEVs in the CPTAC Study 6C data set. I posit that this is caused by the presence of complex variability in CPTAC Study 6C's coupled with little systematic bias. Recall, common normalization methods were designed to mitigate systematic bias. Therefore, when faced with complex variability, common normalization methods actually induced additional variance. I plan to explore this phenomena further in my future work.

CV Reduction

Data Set	Common Mean	PIN	%Perf. Inc.
Instrument Variability	0.19	0.46	2.3x
Sample Variability	0.08	0.41	5x
Serial Dilution	0.23	0.55	2.4x
CPTAC C vs E	-0.14	0.19	NA

PEV Reduction

Data SEt	Common Mean	PIN	%Perf. Inc.
Instrument Variability	0.13	0.73	5.6x
Sample Variability	0.11	0.71	6.5x
Serial Dilution	0.37	0.75	2x
CPTAC C vs E	0.09	0.61	6.8x

Table 8.1: CV and PEV performance improvements for PIN vs. the mean of the five common normalization methods for five analyses. Since these common normalization methods performed similarly with the variance of theanalysis results were low (not to be confused with the variance of the normalized data), comparing PIN to the mean is a fair comparison. The exception is CV reduction for the CPTAC C vs E experiment. This analysis had results with high variance and 4 out of the 5 normalization methods actually induced additional systematic bias. Thus, it's average is negative.

Overfitting

I demonstrated PIN that does not overfit data using the CPTAC Study 6C data set. I identified 42 UPS1 proteins with statistically significant quantitative difference in both the un-normalized and normalized data. I also showed percentage of UPS1 and yeast peptides positively increased (based on the ratio). In theory, in the un-normalized data, 100% of the UPS1 peptides and 0% of the yeast peptides should increase and the number

of peptides decreasing should be 0%. Interestingly, 97% of the UPS1 peptides increased in the un-normalized data compared to only 92% for the PIN normalized data. However, I observed 87% of the yeast peptides increased in the un-normalized data compared to only 33% in the PIN normalized data. I have yet to investigate thoroughly, with statistical rigor, these interesting results.

As discussed in Chapter 7, I also considered comparing my results to those reported by Milac et al. in 2012 [147]. Unfortunately, they detected differences between UPS1 spiked in levels to the quality control while I detected differences between UPS1 spiked in levels. Their method of rolling up intensity levels, from detected peptide signals to peptides and proteins is different from mine.

Biomarker Discovery (Preliminary)

Finally, I applied PIN to a biomarker discovery data set (OPML vs. OSCC). Out of 12,190 peptide signals extracted and normalized by RIPPER/PIN, I found 137 peptide signals with statistically significant quantitative differences (FDR < 0.10, $\alpha = 0.000115$), and 79 peptide signals with statistically significant quantitative differences (FDR < 0.05, $\alpha = 0.000033$). After applying sophisticated statistical analysis, Dr. Koopmeiners, our research team's biostatistician, produced three peptide signal candidate biomarker panels. I then plotted the ROC for the three levels and reported AUCs of 0.95, 0.98, and 0.99. Finally, I demonstrated that using a simple t-test for a single peptide in the biomarker panel, un-normalized data had a p-value of 0.026 and normalized data had a p-value of 0.001.

8.5 Future Work

This dissertation provides the basis for interesting and exciting future work. Here I highlight future work in three areas.

First, my preliminary findings from the OPML vs. OSCC biomarker discovery study,

while exciting, require thorough investigation. As an interdisciplinary effort, I am currently participating in the development of analysis designs for work over the next 6-12 months.

Second, While I have conducted several analyses evaluating my contributions, I plan to evaluate RIPPER and PIN further. I highlight three possible experiments.

- Evaluate PIN against the results reported by Milac et al. Fortunately, this will not require additional HPLC-ESI-MS/MS analysis. I can download the additional data required. However, the amount of data may be too large to evaluate on my personal system. Therefore, I may need to transition my work to the MSI.
- Evaluate RIPPER vis a vis other label free relative quantification software frameworks. Because these software frameworks report at the protein level, I will need to extend RIPPER to report at the protein level as well as the peptide level.
- Formally test window size ramifications with varying HPLC-ESI-MS/MS run durations.

Third, successful software evolves. In order for RIPPER to become widely adopted by the proteomics community (and perhaps others, for example, metabolomics), RIPPER must, therefore, evolve. Here, I highlight several planned activites for evolving RIPPER.

- Integrate RIPPER/PIN into GalaxyP, the proteomic pipeline being developed at the Minnesota Super Computing Institute (MSI).
- Remove RIPPER Limitations.
  - Remove requirement for MS^2scans.
  - Accept mzML file input.
- Extend RIPPER - New functionality for wide spread adoption

- Add interface to R for statistically analyses, such as detecting statistically significant quantiative difference.
- Add visualization is appealing to many users. Thus, I believe that integrating visualization, such as three-dimensional peaks, XICs, heat-maps, and volcano plots, would attract a larger user base.

8.6 Impact

A decade ago, the proteomics community proclaimed that mass spectrometry holds great promise for biomarker discovery, but its potential is unmet. If the Human Genome Project can be completed in a decade, why has mass spectrometry's potential as a biomarker discovery vehicle not been met? In part, the potential is unmet because comparative proteomics work flows lack the repeatability and reproducibility achieved by genomics workflows. Furthermore, I contend that we, the collective proteomics community, have been using an ill-suited paradigm for revealing biological variation.

According to Thomas Kuhn in his 1962 book, *The Structure of Scientific Revolutions*, [19], a paradigm shift is "...a change in the basic assumptions, or paradigms, within the ruling theory of science..." and "... successive transition from one paradigm to another via revolution is the usual developmental pattern of mature science."

Currently, the ruling paradigm for revealing proteomic and peptidomic biological variation is the relative quantification. As stated in a recent Forbes magazine leader ship article, "Paradigm shifts are discontinuous. Working ever more diligently within the existing paradigm leads to frustration, not progress. Instead, scientists have to look at the problem in a fundamentally different way to solve the problem." Doing so, I came to the conclusion that the relative abundance paradigm is ill-suited or discovering biological variation. Consequently, I proposed an alternative paradigm, the proportionality paradigm. Furthermore, I developed a novel method, PIN, implementing the proportionality paradigm and I provide a means to use it, a new iLFRQ software framework

name RIPPER. Unfortunately, the same Forbes article states ..."an interesting facet of paradigm shifts in science is that the older paradigm is difficult to displace precisely because it has been shown to work in solving problems in the past." Despite these difficulties, I contend **the proteomics community must shift to the proportionality paradigm for detecting biological variation**. My reasoning follows.

First, Kuhn specified six criteria for warranting a paradigm shift. In the following list, I describe how the proportionality paradigm fulfills each of the six criteria. With each of criteria fulfilled, a paradigm shift is warranted.

- "Accurate - empirically adequate with experimentation and observation."

  This dissertation presents evidence that the proportionality paradigm is more accurate than the relative abundance paradigm in revealing biological variation.

- "Consistent - internally consistent, but also externally consistent with other theories."

  The proportionality paradigm is consistent with systems biology in that it examines variation in the context of a biological system.

- "Simple - the simplest explanation, principally similar to Occam's Razor."

  The proportionality paradigm is very simple; it is formulated with one simple equation.

- "Broad Scope - a theory's consequences should extend beyond that which it was initially designed to explain."

  The proportionality paradigm should be applicable to wide variety of 'omics disciplines employing HPLC-ESI-MS/MS as an analytical tool.

- "Fruitful - a theory should disclose new phenomena or new relationships among phenomena."

The proportionality paradigm reveals new biological variation, allowing researchers to gain insights into molecular machinery of cellular function and disease progression.

Second, with improved repeatability and reproducibity, researchers will gain confidence in results produced by these comparative proteomic workflows. Without improved repeatabilty and reproducibility in revealing biological variation, mass spectrometry based proteomics' potential will remain unmnet.

In conclusion, I expect my contributions, the application of the proportionality paradigm for iLFRQ, embodied in PIN, and implemented in RIPPER, to change the way researchers analyze HPLC-ESI-MS/MS experimental data. I expect with my contributions' widespread adoption, mass spectrometry based proteomics will be much closer to meeting its potential as a biomarker discovery vehicle. The upshot will, I expect, be reproducibility and repeatability improved, and otherwise falsely reported or missed, statistically significant biological variation discovered. Furthermore, my contributions have the potential to transcend 'omics disciplines. For example, other peptide centric 'omics disciplines employing HPLC-ESI-MS/MS comparative workflows, such as lipidomics, glycomics, and metabolomics, may benefit from detecting biological variation with the proportionality paradigm, PIN, and RIPPER. With multiple disciplines armed with higher confident results, researchers will expend fewer recourses exhaustively validating results through hypothesis-driven experiments. This will shorten the biomarker discovery cycle, thus reducing the bench to bedside cycle time (see Figure 1.1) Ultimately, patients will suffer less, and lives will be saved.

References

[1] SEER. Deaths: Preliminary data for 2011. Technical report, Center for Disease Control and Prevention, 2010 2012.

[2] AmericanCancerSociety. Cancer facts & figures 2009, March 2010 2009.

[3] M. P. Pavlou, E. P. Diamandis, and I. M. Blasutig. The long journey of cancer biomarkers from the bench to the clinic. *Clin Chem*, 59(1):147–57, 2013.

[4] P. R. Srinivas, M. Verma, Y. Zhao, and S. Srivastava. Proteomics for cancer biomarker discovery. *Clin Chem*, 48(8):1160–9, 2002.

[5] J. D. Watson. The human genome project - past, present, and future. *Science*, 248(4951):44–49, 1990.

[6] C. Lengauer, K. W. Kinzler, and B. Vogelstein. Genetic instabilities in human cancers. *Nature*, 396(6712):643–649, 1998.

[7] M. E. Macdonald, C. M. Ambrose, M. P. Duyao, R. H. Myers, C. Lin, L. Srinidhi, G. Barnes, S. A. Taylor, M. James, N. Groot, H. Macfarlane, B. Jenkins, M. A. Anderson, N. S. Wexler, J. F. Gusella, G. P. Bates, S. Baxendale, H. Hummerich, S. Kirby, M. North, S. Youngman, R. Mott, G. Zehetner, Z. Sedlacek, A. Poustka, A. M. Frischauf, H. Lehrach, A. J. Buckler, D. Church, L. Doucettestamm, M. C. Odonovan, L. Ribaramirez, M. Shah, V. P. Stanton, S. A. Strobel, K. M. Draths, J. L. Wales, P. Dervan, D. E. Housman, M. Altherr, R. Shiang, L. Thompson, T. Fielder, J. J. Wasmuth, D. Tagle, J. Valdes, L. Elmer, M. Allard, L. Castilla, M. Swaroop, K. Blanchard, F. S. Collins, R. Snell, T. Holloway, K. Gillespie, N. Datson, D. Shaw, and P. S. Harper. A novel gene containing a trinucleotide repeat that is expanded and unstable on huntingtons-disease chromosomes. *Cell*, 72(6):971–983, 1993.

[8] J. R. Yates. The revolution and evolution of shotgun proteomics for large-scale proteome analysis. *Journal of the American Chemical Society*, 135(5):1629–1640, 2013. 087WF Times Cited:0 Cited References Count:148.

[9] Charlotte W. Pratt and Kathleen Cornely. *Essential biochemistry*. J. Wiley, Hoboken, NJ, 2004.

[10] M. Bantscheff, M. Boesche, D. Eberhard, T. Matthieson, G. Sweetman, and B. Kuster. Robust and sensitive itraq quantification on an ltq orbitrap mass spectrometer. *Molecular & Cellular Proteomics*, 7(9):1702–13, 2008.

[11] B. Domon and R. Aebersold. Review - mass spectrometry and protein analysis. *Science*, 312:212–217, 2006.

[12] N. L. Anderson and N. G. Anderson. The human plasma proteome: History, character, and diagnostic prospects (vol 1, pg 845, 2002). *Molecular & Cellular Proteomics*, 2(1):50–50, 2003.

[13] M. Bantscheff. Mass spectrometry-based chemoproteomic approaches. *Methods Mol Biol*, 803:3–13, 2012.

[14] K. Mann and M. Mann. The chicken egg yolk plasma and granule proteomes. *Proteomics*, 8(1):178–91, 2008.

[15] S. E. Ong and M. Mann. Mass spectrometry-based proteomics turns quantitative. *Nat Chem Biol*, 1(5):252–62, 2005.

[16] J. Cox and M. Mann. Maxquant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, 26(12):1367–72, 2008.

[17] D. B. Allison, X. Q. Cui, C. P. Page, and M. Sabripour. Microarray data analysis: from disarray to consolidation and consensus (vol 7, pg 55, 2006). *Nature Reviews Genetics*, 7(5):406–406, 2006.

[18] A. I. Nesvizhskii, O. Vitek, and R. Aebersold. Analysis and validation of proteomic data generated by tandem mass spectrometry. *Nat Methods*, 4(10):787–97, 2007.

[19] Thomas S. Kuhn. *The structure of scientific revolutions*. University of Chicago Press, Chicago, 1962.

[20] M. R. Wilkins, J. C. Sanchez, A. A. Gooley, R. D. Appel, I. HumpherySmith, D. F. Hochstrasser, and K. L. Williams. Progress with proteome projects: Why all proteins expressed by a genome should be identified and how to do it. *Biotechnology and Genetic Engineering Reviews*, 13:19–50, 1996.

[21] IUPAC. Iupac gold book, 2004.

[22] T. Reynolds. For proteomics research, a new race has begun. *J Natl Cancer Inst.*, 94:552–554, 2002.

[23] B. F. Cravatt, G. M. Simon, and 3rd Yates, J. R. The biological impact of mass-spectrometry-based proteomics. *Nature*, 450(7172):991–1000, 2007.

[24] A. Talapatra, R. Rouse, and G. Hardiman. Protein microarrays: challenges and promises. *Pharmacogenomics*, 3(4):527–36, 2002.

[25] D. A. Cairns. Statistical issues in quality control of proteomic analyses: good experimental design and planning. *Proteomics*, 11(6):1037–48, 2011.

[26] A. L. Oberg and O. Vitek. Statistical design of quantitative mass spectrometry-based proteomic experiments. *Journal of Proteome Research*, 8(5):2144–56, 2009.

[27] E. P. de Jong, S. K. van Riper, J. S. Koopmeiners, J. V. Carlis, and T. J. Griffin. Sample collection and handling considerations for peptidomic studies in whole saliva; implications for biomarker discovery. *Clin Chim Acta*, 412(23-24):2284–8, 2011.

[28] A. G. Paulovich, D. Billheimer, A. J. Ham, L. Vega-Montoto, P. A. Rudnick, D. L. Tabb, P. Wang, R. K. Blackman, D. M. Bunk, H. L. Cardasis, K. R. Clauser, C. R. Kinsinger, B. Schilling, T. J. Tegeler, A. M. Variyath, M. Wang, J. R. Whiteaker, L. J. Zimmerman, D. Fenyo, S. A. Carr, S. J. Fisher, B. W. Gibson, M. Mesri, T. A. Neubert, F. E. Regnier, H. Rodriguez, C. Spiegelman, S. E. Stein, P. Tempst, and D. C. Liebler. Interlaboratory study characterizing a yeast performance standard for benchmarking lc-ms platform performance. *Molecular & Cellular Proteomics*, 9(2):242–54, 2010.

[29] Michael P. Washburn. Sample preparation and in-solution protease digestion of proteins for chromatography-based proteomic analysis. *Current Protocols in Protein Science*, 53:23.6.1âĂŞ23.6.11, 2001.

[30] J. V. Olsen, L. M. de Godoy, G. Li, B. Macek, P. Mortensen, R. Pesch, A. Makarov, O. Lange, S. Horning, and M. Mann. Parts per million mass accuracy on an orbitrap mass spectrometer via lock mass injection into a c-trap. *Molecular & Cellular Proteomics*, 4(12):2010–21, 2005.

[31] D. A. Wolters, M. P. Washburn, and 3rd Yates, J. R. An automated multidimensional protein identification technology for shotgun proteomics. *Analytical Chemistry*, 73(23):5683–90, 2001.

[32] J. Calvin Giddings. Maximum number of components resolvable by gel filtration and other elution chromatographic methods. *Analytical Chemistry*, 39:1027–1028, 1967.

[33] J. M. Davis and J. C. Giddings. Statistical-theory of component overlap in multicomponent chromatograms. *Analytical Chemistry*, 55:418–424, 1983.

[34] B. Futcher, G. I. Latter, P. Monardo, C. S. McLaughlin, and J. I. Garrels. A sampling of the yeast proteome. *Mol Cell Biol*, 19(11):7357–68, 1999.

[35] S. P. Gygi, G. L. Corthals, Y. Zhang, Y. Rochon, and R. Aebersold. Evaluation of two-dimensional gel electrophoresis-based proteome analysis technology. *Proc Natl Acad Sci U S A*, 97(17):9390–5, 2000.

[36] J. R. Yates. Mass spectral analysis in proteomics. *Annu. Rev. Biophys. Biomol. Struct.*, 33:297–316, 2004.

[37] M. L. Fournier, J. M. Gilmore, S. A. Martin-Brown, and M. P. Washburn. Multidimensional separations-based shotgun proteomics. *Chemical Reviews*, 107:3654–3686, 2007.

[38] D. F. Hunt, R. A. Henderson, J. Shabanowitz, K. Sakaguchi, H. Michel, N. Sevilir, A. L. Cox, E. Appella, and V. H. Engelhard. Characterization of peptides bound to the class-i mhc molecule hla-a2.1 by mass-spectrometry. *Science*, 255:1261–1263, 1992.

[39] H. F. Walton. Ion-exchange and liquid column chromatography. *Analytical Chemistry*, 48:R52–R66, 1976.

[40] A. Ducret, I. Van Oostveen, J. K. Eng, J. R. Yates, and R. Aebersold. High throughput protein characterization by automated reverse-phase chromatography electrospray tandem mass spectrometry. *Protein Science*, 7:706–719, 1998.

[41] A. J. Alpert. Hydrophilic-interaction chromatography for the separation of peptides, nucleic-acids and other polar compounds. *J Chromatogr A*, 499:177–196, 1990.

[42] P. Cuatrecasas. Protein-purification by affinity-chromatography - derivatizations of agarose and polyacrylamide beads. *Journal of Biological Chemistry*, pages 16–16, 1980.

[43] L. R. Snyder, J. W. Dolan, and J. R. Gant. Gradient elution in high-performance liquid-chromatography: 1. theoretical basis for reversed-phase systems. *Journal of Chromatography A*, 165:3–30, 1979.

[44] J. W. Dolan, J. R. Gant, and L. R. Snyder. Gradient elution in high-performance liquid-chromatography. *Journal of Chromatography A*, 165:31–58, 1979.

[45] G. B. Cox, L. R. Snyder, and J. W. Dolan. Preparative high-performance liquid-chromatography under gradient conditions. *J. Chromatogr. A*, 484:409–423, 1989.

[46] R. Aebersold and M. Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207, 2003.

[47] Junmin Peng, Joshua E. Elias, Carson C. Thoreen, Larry J. Licklider, and Steven P. Gygi. Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (lc/lcâĹŠms/ms) for large-scale protein analysis:âĂĽthe yeast proteome. *Journal of Proteome Research*, 2:43–50, 2002.

[48] J. Chen, C. S. Lee, Y. Shen, R. D. Smith, and E. H. Baehrecke. Integration of capillary isoelectric focusing with capillary reversed-phase liquid chromatography for two-dimensional proteomics separation. *Electrophoresis*, 23(18):3143–8, 2002.

[49] J. Z. Chen, B. M. Balgley, D. L. DeVoe, and C. S. Lee. Capillary isoelectric focusing-based multidimensional concentration/separation platform for proteome analysis. *Analytical Chemistry*, 75(13):3145–3152, 2003.

[50] T. Guo, W. J. Wang, P. A. Rudnick, T. Song, J. Li, Z. P. Zhuang, R. J. Weil, D. L. DeVoe, C. S. Lee, and B. M. Balgley. Proteome analysis of microdissected formalin-fixed and paraffin-embedded tissue specimens. *Journal of Histochemistry & Cytochemistry*, 55(7):763–772, 2007.

[51] H. W. Xie, N. L. Rhodus, R. J. Griffin, J. V. Carlis, and T. J. Griffin. A catalogue of human saliva proteins identified by free flow electrophoresis-based peptide separation and tandem mass spectrometry. *Molecular and Cellular Proteomics*, 4:1826–1830, 2005.

[52] H. W. Xie, S. Bandhakavi, and T. J. Griffin. Evaluating preparative isoelectric focusing of complex peptide mixtures for tandem mass spectrometry-based proteomics: A case study in profiling chromatin-enriched subcellular fractions in saccharomyces cerevisiae. *Analytical Chemistry*, 77:3198–3207, 2005.

[53] R. J. C. Slebos, J. W. C. Brock, N. F. Winters, S. R. Stuart, M. A. Martinez, M. Li, M. C. Chambers, L. J. Zimmerman, A. J. Ham, D. L. Tabb, and D. C. Liebler. Evaluation of strong cation exchange versus isoelectric focusing of peptides for multidimensional liquid chromatography-tandem mass spectrometry. *Journal of Proteome Research*, 7(12):5286–5294, 2008.

[54] S. Bandhakavi, M. D. Stone, G. Onsongo, S. K. Van Riper, and T. J. Griffin. A dynamic range compression and three-dimensional peptide fractionation analysis platform expands proteome coverage and the diagnostic potential of whole saliva. *Journal of Proteome Research*, 8(12):5590–600, 2009.

[55] S. Bandhakavi, T. W. Markowski, H. Xie, and T. J. Griffin. Three-dimensional peptide fractionation for highly sensitive nanoscale lc-based shotgun proteomic analysis of complex protein mixtures. *Methods Mol Biol*, 790:47–56, 2011.

[56] F. Hillenkamp, M. Karas, D. Holtkamp, and P. Klusener. Energy deposition in ultraviolet-laser desorption mass-spectrometry of biomolecules. *International Journal of Mass Spectrometry and Ion Processes*, 69(3):265–276, 1986.

[57] F. Hillenkamp and M. Karas. Mass spectrometry of peptides and proteins by matrix-assisted ultraviolet laser desorption/ionization. *Methods Enzymol*, 193:280–95, 1990.

[58] F. Hillenkamp, M. Karas, R. C. Beavis, and B. T. Chait. Matrix-assisted laser desorption/ionization mass spectrometry of biopolymers. *Analytical Chemistry*, 63(24):1193A–1203A, 1991.

[59] K. Tanaka, H. Waki, Y. Ido, S. Akita, Y. Yoshida, T. Yoshida, and T. Matsuo. Protein and polymer analyses up to m/z 100 000 by laser ionization time-of-flight mass spectrometry. *Rapid Comm Mass Spectrometry*, 2(8):151–3, 1988.

[60] K. Tanaka. The origin of macromolecule ionization by laser irradiation (nobel lecture). *Angew Chem Int Ed Engl*, 42(33):3860–70, 2003.

[61] J. B. Fenn, M. Mann, C. K. Meng, S. F. Wong, and C. M. Whitehouse. Electrospray ionization for mass spectrometry of large biomolecules. *Science*, 246(4926):64–71, 1989.

[62] J. R. Yates, C. I. Ruse, and A. Nakorchevsky. Proteomics by mass spectrometry: approaches, advances, and applications. *Annual Review of Biomedical Engineering*, 11:49–79, 2009.

[63] L. Ceraulo, G. Giorgi, V. T. Liveri, D. Bongiorno, S. Indelicato, F. Di Gaudio, and S. Indelicato. Mass spectrometry of surfactant aggregates. *Eur J Mass Spectrom (Chichester, Eng)*, 17(6):525–41, 2011.

[64] R. D. Smith, J. A. Loo, C. G. Edmonds, C. J. Barinaga, and H. R. Udseth. New developments in biochemical mass spectrometry: electrospray ionization. *Analytical Chemistry*, 62(9):882–99, 1990.

[65] R. J. Cotter, A. S. Woods, and T. J. Cornish. Biological applications of time-of-flight mass-spectrometry. *Biochem. Soc. Trans.*, 22:539–542, 1994.

[66] A. Makarov. Electrostatic axially harmonic orbital trapping: a high-performance technique of mass analysis. *Analytical Chemistry*, 72(6):1156–62, 2000.

[67] Qizhi Hu, Robert J Noll, Hongyan Li, Alexander Makarov, Mark Hardman, and R Graham Cooks. The orbitrap: a new mass spectrometer. *Journal of Mass Spectrometry*, 40:430–443, 2005.

[68] Alexander Makarov, Eduard Denisov, Oliver Lange, and Stevan Horning. Dynamic range of mass accuracy in ltq orbitrap hybrid mass spectrometer. *Journal of the American Society for Mass Spectrometry*, 17(7):977–982, July 2006.

[69] F. Dubois, R. Knochenmuss, R. Zenobi, A. Brunelle, C. Deprun, and Y. Le Beyec. A comparison between ion-to-photon and microchannel plate detectors. *Rapid Communications in Mass Spectrometry*, 13:786–791, 1999.

[70] I. J. Amster. Fourier transform mass spectrometry. *J Mass Spectrom*, 31:1325–1337, 1996.

[71] B. T. Chait, R. Wang, R. C. Beavis, and S. B. Kent. Protein ladder sequencing. *Science*, 262(5130):89–92, 1993.

[72] F. W. McLafferty. Interpretation of mass-spectra - basic data and computer aids. *J. Am. Chem.*, 25:1058–1059, 1979.

[73] D. F. Hunt, 3rd Yates, J. R., J. Shabanowitz, S. Winston, and C. R. Hauer. Protein sequencing by tandem mass spectrometry. *Proc Natl Acad Sci U S A*, 83(17):6233–7, 1986.

[74] M. Sandin, J. Teleman, J. Malmstrom, and F. Levander. Data processing methods and quality control strategies for label-free lc-ms protein quantification. *Biochim Biophys Acta*, 2013.

[75] M. Bellew, M. Coram, M. Fitzgibbon, M. Igra, T. Randolph, P. Wang, D. May, J. Eng, R. Fang, C. Lin, J. Chen, D. Goodlett, J. Whiteaker, A. Paulovich, and M. McIntosh. A suite of algorithms for the comprehensive analysis of complex protein mixtures using high-resolution lc-ms. *Bioinformatics*, 22(15):1902–9, 2006.

[76] T. Pluskal, S. Castillo, A. Villar-Briones, and M. Oresic. Mzmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics*, 11:395, 2010.

[77] E. Lange, R. Tautenhahn, S. Neumann, and C. Gropl. Critical assessment of alignment procedures for lc-ms proteomics and metabolomics measurements. *BMC Bioinformatics*, 9:375, 2008.

[78] M. W. Senko, S. C. Beu, and F. W. Mclafferty. Determination of monoisotopic masses and ion populations for large biomolecules from resolved isotopic distributions. *J Am Soc Mass Spectrom*, 6(4):229–233, 1995.

[79] S. Gay, P. A. Binz, D. F. Hochstrasser, and R. D. Appel. Modeling peptide mass fingerprinting data using the atomic composition of peptides. *Electrophoresis*, 20(18):3527–3534, 1999.

[80] H. Sakoe and S. Chiba. Dynamic-programming algorithm optimization for spoken word recognition. *Ieee Transactions on Acoustics Speech and Signal Processing*, 26(1):43–49, 1978.

[81] C. P. Wang and T. L. Isenhour. Time-warping algorithm applied to chromatographic peak matching gas-chromatography fourier-transform infrared mass-spectrometry. *Analytical Chemistry*, 59(4):649–654, 1987.

[82] N. P. Nielsen, J. Smedsgaard, and J. C. Frisvad. Full second-order chromatographic/spectrometric data matrices for automated sample identification and component analysis by non-data-reducing image analysis. *Analytical Chemistry*, 71(3):727–35, 1999.

[83] D. Bylund, R. Danielsson, G. Malmquist, and K. E. Markides. Chromatographic alignment by warping and dynamic programming as a pre-processing tool for parafac modelling of liquid chromatography-mass spectrometry data. *Journal of Chromatography A*, 961(2):237–244, 2002.

[84] P. H. Eilers. Parametric time warping. *Analytical Chemistry*, 76(2):404–11, 2004.

[85] M. Bantscheff, M. Schirle, G. Sweetman, J. Rick, and B. Kuster. Quantitative mass spectrometry in proteomics: a critical review. *Anal Bioanal Chem*, 389(4):1017–31, 2007.

[86] Ingvar Eidhammer, Harald Barsnes, Geir Egil Eide, and Lennart Martens. *Computational and statistical methods for protein quantification by mass spectrometry.* 2013.

[87] P. V. Bondarenko, D. Chelius, and T. A. Shaler. Identification and relative quantitation of protein mixtures by enzymatic digestion followed by capillary reversed-phase liquid chromatography-tandem mass spectrometry. *Analytical Chemistry*, 74(18):4741–9, 2002.

[88] T. J. Griffin and R. Aebersold. Advances in proteome analysis by mass spectrometry. *J Biol Chem*, 276(49):45497–500, 2001.

[89] Ingvar Eidhammer, Flikka Kristian, Lennart Martens, and Svein-Ole Mikalsen. *Computational methods for mass spectrometry proteomics*. John Wiley & Sons, Chichester, England ; Hoboken, NJ, 2007.

[90] J. K. Eng, A. L. Mccormack, and J. R. Yates. An approach to correlate tandem mass-spectral data of peptides with amino-acid-sequences in a protein database. *J Am Soc Mass Spectrom*, 5(11):976–989, 1994.

[91] L. A. Powell and G. M. Hieftje. Computer identification of infrared-spectra by correlation-based file searching. *Anal. Chim. Acta*, 100:313–327, 1978.

[92] 3rd Yates, J. R., S. F. Morgan, C. L. Gatlin, P. R. Griffin, and J. K. Eng. Method to compare collision-induced dissociation spectra of peptides: potential for library searching and subtractive analysis. *Analytical Chemistry*, 70(17):3557–65, 1998.

[93] H. Lam, E. W. Deutsch, J. S. Eddes, J. K. Eng, N. King, S. E. Stein, and R. Aebersold. Development and validation of a spectral library searching method for peptide identification from ms/ms. *Proteomics*, 7(5):655–67, 2007.

[94] C. Bartels. Fast algorithm for peptide sequencing by mass-spectroscopy. *Biomedical and Environmental Mass Spectrometry*, 19(6):363–368, 1990.

[95] A. Frank and P. Pevzner. Pepnovo: de novo peptide sequencing via probabilistic network modeling. *Analytical Chemistry*, 77(4):964–73, 2005.

[96] V. Dancik, T. A. Addona, K. R. Clauser, J. E. Vath, and P. A. Pevzner. De novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 6(3-4):327–42, 1999.

[97] T. Chen, M. Y. Kao, M. Tepel, J. Rush, and G. M. Church. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology*, 8(3):325–37, 2001.

[98] B. Fischer, V. Roth, F. Roos, J. Grossmann, S. Baginsky, P. Widmayer, W. Gruissem, and J. M. Buhmann. Novohmm: a hidden markov model for de novo peptide sequencing. *Analytical Chemistry*, 77(22):7265–73, 2005.

[99] A. I. Nesvizhskii and R. Aebersold. Interpretation of shotgun proteomic data: the protein inference problem. *Molecular & Cellular Proteomics*, 4(10):1419–40, 2005.

[100] Alselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[101] J. M. Bland and D. G. Altman. Multiple significance tests - the bonferroni method. *British Medical Journal*, 310(6973):170–170, 1995.

[102] D. J. States, G. S. Omenn, T. W. Blackwell, D. Fermin, J. Eng, D. W. Speicher, and S. M. Hanash. Challenges in deriving high-confidence protein identifications from data gathered by a hupo plasma proteome collaborative study. *Nature Biotechnology*, 24(3):333–8, 2006.

[103] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate - a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B-Methodological*, 57(1):289–300, 1995.

[104] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by ms/ms and database search. *Analytical Chemistry*, 74(20):5383–92, 2002.

[105] A. I. Nesvizhskii, A. Keller, E. Kolker, and R. Aebersold. A statistical model for identifying proteins by tandem mass spectrometry. *Analytical Chemistry*, 75(17):4646–58, 2003.

[106] Eric W Deutsch, Luis Mendoza, David Shteynberg, Terry Farrah, Henry Lam, Natalie Tasman, Zhi Sun, Erik Nilsson, Brian Pratt, and Bryan Prazen. A guided tour of the trans-proteomic pipeline. *Proteomics*, 10:1150–1159, 2010.

[107] B. C. Searle and M. Turner. Improving computer interpretation of linear ion trap proteomics data using scaffold. *Molecular & Cellular Proteomics*, 5(10):S297–S297, 2006.

[108] B. C. Searle. Scaffold: a bioinformatic tool for validating ms/ms-based proteomic studies. *Proteomics*, 10(6):1265–9, 2010.

[109] M. P. Molloy, E. E. Brzezinski, J. Hang, M. T. McDowell, and R. A. VanBogelen. Overcoming technical variation and biological variation in quantitative proteomics. *Proteomics*, 3(10):1912–9, 2003.

[110] J. M. Bland and D. G. Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet*, 1(8476):307–10, 1986.

[111] D. L. Tabb, L. Vega-Montoto, P. A. Rudnick, A. M. Variyath, A. J. Ham, D. M. Bunk, L. E. Kilpatrick, D. D. Billheimer, R. K. Blackman, H. L. Cardasis, S. A. Carr, K. R. Clauser, J. D. Jaffe, K. A. Kowalski, T. A. Neubert, F. E. Regnier, B. Schilling, T. J. Tegeler, M. Wang, P. Wang, J. R. Whiteaker, L. J. Zimmerman, S. J. Fisher, B. W. Gibson, C. R. Kinsinger, M. Mesri, H. Rodriguez, S. E. Stein, P. Tempst, A. G. Paulovich, D. C. Liebler, and C. Spiegelman. Repeatability and reproducibility in proteomic identifications by liquid chromatography-tandem mass spectrometry. *Journal of Proteome Research*, 9(2):761–76, 2010.

[112] Alan D. McNaught, Andrew Wilkinson, International Union of Pure, and Applied Chemistry. Compendium of chemical terminology : Iupac recommendations, 1997.

[113] Victor Gold, International Union of Pure, and Applied Chemistry. Compendium of chemical terminology : Iupac recommendations, 1987.

[114] N. Delmotte, M. Lasaosa, A. Tholey, E. Heinzle, A. van Dorsselaer, and C. G. Huber. Repeatability of peptide identifications in shotgun proteome analysis employing off-line two-dimensional chromatographic separations and ion-trap ms. *J Sep Sci*, 32(8):1156–64, 2009.

[115] P. M. van Midwoud, L. Rieux, R. Bischoff, E. Verpoorte, and H. A. Niederlander. Improvement of recovery and repeatability in liquid chromatography-mass spectrometry analysis of peptides. *Journal of Proteome Research*, 6(2):781–91, 2007.

[116] M. Wang, J. You, K. G. Bemis, T. J. Tegeler, and D. P. Brown. Label-free mass spectrometry-based protein quantification technologies in proteomic analysis. *Brief Funct Genomic Proteomic*, 7(5):329–39, 2008.

[117] S. J. Callister, R. C. Barry, J. N. Adkins, E. T. Johnson, W. J. Qian, B. J. Webb-Robertson, R. D. Smith, and M. S. Lipton. Normalization approaches for removing systematic biases associated with mass spectrometry and label-free proteomics. *Journal of Proteome Research*, 5(2):277–86, 2006.

[118] K. Kultima, A. Nilsson, B. Scholz, U. L. Rossbach, M. Falth, and P. E. Andren. Development and evaluation of normalization methods for label-free relative quantification of endogenous peptides. *Molecular & Cellular Proteomics*, 8(10):2285–95, 2009.

[119] K. A. Neilson, N. A. Ali, S. Muralidharan, M. Mirzaei, M. Mariani, G. Assadourian, A. Lee, S. C. van Sluyter, and P. A. Haynes. Less label, more free: approaches in label-free quantitative mass spectrometry. *Proteomics*, 11(4):535–53, 2011.

[120] M. Anderle, S. Roy, H. Lin, C. Becker, and K. Joho. Quantifying reproducibility for differential proteomics: noise analysis for protein liquid chromatography-mass spectrometry of human serum. *Bioinformatics*, 20(18):3575–3582, 2004.

[121] L. N. Mueller, O. Rinner, A. Schmidt, S. Letarte, B. Bodenmiller, M. Y. Brusniak, O. Vitek, R. Aebersold, and M. Muller. Superhirn - a novel tool for high resolution lc-ms-based peptide/protein profiling. *Proteomics*, 7(19):3470–80, 2007.

[122] W. M. Old, K. Meyer-Arendt, L. Aveline-Wolf, K. G. Pierce, A. Mendoza, J. R. Sevinsky, K. A. Resing, and N. G. Ahn. Comparison of label-free methods for quantifying human proteins by shotgun proteomics. *Molecular & Cellular Proteomics*, 4(10):1487–502, 2005.

[123] R. Ramanathan, R. Zhong, N. Blumenkrantz, S. K. Chowdhury, and K. B. Alton. Response normalized liquid chromatography nanospray ionization mass spectrometry. *J Am Soc Mass Spectrom*, 18(10):1891–9, 2007.

[124] S. Jung, U. Effelsberg, and U. Tallarek. Microchip electrospray: cone-jet stability analysis for water-acetonitrile and water-methanol mobile phases. *J Chromatogr A*, 1218(12):1611–9, 2011.

[125] David Lovell, Warren Muller, Jen Taylor, Alec Zwart, Chris Helliwell, Vera Pawlowsky-Glahn, and Antonella Buccianti. Proportions, percentages, ppm: do the molecular biosciences treat compositional data right? *Compositional Data Analysis: Theory and Applications*, page 193, 2011.

[126] J. W. Thompson, M. T. Forrester, M. A. Moseley, and M. W. Foster. Solid-phase capture for the detection and relative quantification of s-nitrosoproteins by mass spectrometry. *Methods*, 2012. Thompson, J Will Forrester, Michael T Moseley, M Arthur Foster, Matthew W R21 HL106121/HL/NHLBI NIH HHS/ San Diego, Calif. Methods. 2012 Oct 11. pii: S1046-2023(12)00256-3. doi: 10.1016/j.ymeth.2012.10.001.

[127] S. E. Ong, B. Blagoev, I. Kratchmarova, D. B. Kristensen, H. Steen, A. Pandey, and M. Mann. Stable isotope labeling by amino acids in cell culture, silac, as a simple and accurate approach to expression proteomics. *Molecular & Cellular Proteomics*, 1(5):376–86, 2002.

[128] S. P. Gygi, B. Rist, S. A. Gerber, F. Turecek, M. H. Gelb, and R. Aebersold. Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nature Biotechnology*, 17(10):994–9, 1999.

[129] O. A. Mirgorodskaya, Y. P. Kozmin, M. I. Titov, R. Korner, C. P. Sonksen, and P. Roepstorff. Quantitation of peptides and proteins by matrix-assisted laser desorption/ionization mass spectrometry using o-18-labeled internal standards. *Rapid Communications in Mass Spectrometry*, 14:1226–1232, 2000.

[130] S. Wiese, K. A. Reidegeld, H. E. Meyer, and B. Warscheid. Protein labeling by itraq: A new tool for quantitative mass spectrometry in proteome research. *Proteomics*, 7(3):340–350, 2007. 137CW Times Cited:136 Cited References Count:32.

[131] A. Thompson, J. Schafer, K. Kuhn, S. Kienle, J. Schwarz, G. Schmidt, T. Neumann, R. Johnstone, A. K. Mohammed, and C. Hamon. Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by ms/ms. *Analytical Chemistry*, 75(8):1895–904, 2003.

[132] S. K. Van Riper, E. P. de Jong, J. V. Carlis, and T. J. Griffin. Mass spectrometry-based proteomics: basic principles and emerging technologies and directions. *Advances in Experimental Medicine and Biology*, 990:1–35, 2013.

[133] C. C. Wu, M. J. MacCoss, K. E. Howell, D. E. Matthews, and 3rd Yates, J. R. Metabolic labeling of mammalian organisms with stable isotopes for quantitative proteomic analysis. *Analytical Chemistry*, 76(17):4951–9, 2004.

[134] M. M. Savitski, G. Sweetman, M. Askenazi, J. A. Marto, M. Lang, N. Zinn, and M. Bantscheff. Delayed fragmentation and optimized isolation width settings for

improvement of protein identification and accuracy of isobaric mass tag quantification on orbitrap-type mass spectrometers. *Analytical Chemistry*, 83(23):8959–67, 2011.

[135] N. M. Griffin, J. Yu, F. Long, P. Oh, S. Shore, Y. Li, J. A. Koziol, and J. E. Schnitzer. Label-free, normalized quantification of complex mass spectrometry data for proteomic analysis. *Nature Biotechnology*, 28(1):83–9, 2010.

[136] D. Chelius and P. V. Bondarenko. Quantitative profiling of proteins in complex mixtures using liquid chromatography and mass spectrometry. *Journal of Proteome Research*, 1(4):317–23, 2002.

[137] J. C. Silva, R. Denny, C. A. Dorschel, M. Gorenstein, I. J. Kass, G. Z. Li, T. McKenna, M. J. Nold, K. Richardson, P. Young, and S. Geromanos. Quantitative proteomic analysis by accurate mass retention time pairs. *Analytical Chemistry*, 77(7):2187–200, 2005.

[138] R. E. Higgs, M. D. Knierman, V. Gelfanova, J. P. Butler, and J. E. Hale. Comprehensive label-free method for the relative quantification of proteins from biological samples. *Journal of Proteome Research*, 4(4):1442–1450, 2005. 954VV Times Cited:114 Cited References Count:30.

[139] J. W. H. Wong and G. Cagney. An overview of label-free quantitation methods in proteomics by mass spectrometry. *Proteome Bioinformatics*, 604:273–283, 2010. Bmt37 Times Cited:16 Cited References Count:41 Methods in Molecular Biology.

[140] Yudell L. Luke and Milton Abramowitz. *Mathematical functions and their approximations*. Academic Press, New York, 1975. 75022358 Yudell L. Luke. 23 cm. An updated version of part of Handbook of mathematical functions with formulas, graphs, and mathematical tables, edited by M. Abramowitz and I. A. Stegun. Bibliography: p. 517-544. Includes indexes.

[141] H. Liu, R. G. Sadygov, and 3rd Yates, J. R. A model for random sampling and estimation of relative protein abundance in shotgun proteomics. *Analytical Chemistry*, 76(14):4193–201, 2004.

[142] A. Gilchrist, C. E. Au, J. Hiding, A. W. Bell, J. Fernandez-Rodriguez, S. Lesimple, H. Nagaya, L. Roy, S. J. C. Gosline, M. Hallett, J. Paiement, R. E. Kearney, T. Nilsson, and J. J. M. Bergeron. Quantitative proteomics analysis of the secretory pathway. *Cell*, 127:1265–1281, 2006.

[143] B. Zybailov, A. L. Mosley, M. E. Sardiu, M. K. Coleman, L. Florens, and M. P. Washburn. Statistical analysis of membrane proteome expression changes in saccharomyces cerevisiae. *Journal of Proteome Research*, 5(9):2339–47, 2006.

[144] J. Rappsilber, U. Ryder, A. I. Lamond, and M. Mann. Large-scale proteomic analysis of the human spliceosome. *Genome Res*, 12(8):1231–45, 2002.

[145] Y. Ishihama, Y. Oda, T. Tabata, T. Sato, T. Nagasu, J. Rappsilber, and M. Mann. Exponentially modified protein abundance index (empai) for estimation of absolute protein amount in proteomics by the number of sequenced peptides per protein. *Molecular & Cellular Proteomics*, 4(9):1265–72, 2005.

[146] F. Xie, T. Liu, W. J. Qian, V. A. Petyuk, and R. D. Smith. Liquid chromatography-mass spectrometry-based quantitative proteomics. *Journal of Biological Chemistry*, 286(29):25443–25449, 2011.

[147] T. I. Milac, T. W. Randolph, and P. Wang. Analyzing lc-ms/ms data by spectral count and ion abundance: two case studies. *Statistics and Its Interface*, 5(1):75–87, 2012. 902RO Times Cited:0 Cited References Count:38.

[148] T. Clough, M. Key, I. Ott, S. Ragg, G. Schadow, and O. Vitek. Protein quantification in label-free lc-ms experiments. *Journal of Proteome Research*, 8(11):5275–5284, 2009.

[149] K. Podwojski, A. Fritsch, D. C. Chamrad, W. Paul, B. Sitek, K. Stuhler, P. Mutzel, C. Stephan, H. E. Meyer, W. Urfer, K. Ickstadt, and J. Rahnenfuhrer. Retention time alignment algorithms for lc/ms data must consider non-linear shifts. *Bioinformatics*, 25(6):758–64, 2009.

[150] D. H. Lundgren, S. I. Hwang, L. Wu, and D. K. Han. Role of spectral counting in quantitative proteomics. *Expert Rev Proteomics*, 7(1):39–53, 2010.

[151] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–93, 2003.

[152] W. Huber, A. von Heydebreck, H. Sultmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18 Suppl 1:S96–104, 2002.

[153] Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4), 2002.

[154] P. Roy, C. Truntzer, D. Maucort-Boulch, T. Jouve, and N. Molinari. Protein mass spectra data analysis for clinical biomarker discovery: a global review. *Brief Bioinform*, 12(2):176–86, 2011.

[155] A. H. P. America and J. H. G. Cordewener. Comparative lc-ms: A landscape of peaks and valleys. *Proteomics*, 8(4):731–749, 2008.

[156] C. Colantuoni, G. Henry, S. Zeger, and J. Pevsner. Local mean normalization of microarray element signal intensities across an array surface: Quality control and correction of spatially systematic artifacts. *Biotechniques*, 32(6):1316–1320, 2002.

[157] J. Quackenbush, S. C. Geller, J. P. Gregg, P. Hagerman, and D. M. Rocke. Transformation and normalization of oligonucleotide microarray data. *Nat Genet*, 19(14):1817–1823, 2003.

[158] T. Park, S. G. Yi, S. H. Kang, S. Lee, Y. S. Lee, and R. Simon. Evaluation of normalization methods for microarray data. *BMC Bioinformatics*, 4:33, 2003.

[159] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

[160] S. Dudoit, Y. H. Yang, M. J. Callow, and T. P. Speed. Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments. *Statistica Sinica*, 12(1):111–139, 2002.

[161] Carolina Johansson, Jenny Samskog, Lars SundstrâĹŽâĹĆm, Henrik Wadensten, Lennart BjâĹŽâĹĆrkesten, and John Flensburg. Differential expression analysis of escherichia coli proteins using a novel software for relative quantitation of lc-ms/ms data. *PROTEOMICS*, 6(16):4475–4485, 2006.

[162] SushmitaMimi Roy and ChristopherH. Becker. Quantification of proteins and metabolites by mass spectrometry without isotopic labeling. In Salvatore Sechi, editor, *Quantitative Proteomics by Mass Spectrometry*, volume 359 of *Methods in Molecular Biology*, pages 87–105. Humana Press, 2007.

[163] W. X. Wang, H. H. Zhou, H. Lin, S. Roy, T. A. Shaler, L. R. Hill, S. Norton, P. Kumar, M. Anderle, and C. H. Becker. Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards. *Analytical Chemistry*, 75(18):4818–4826, 2003.

[164] G. K. Smyth. *Limma: linear models for microarray data*, pages 397–420. Springer, New York, 2005.

[165] M. Sturm, A. Bertsch, C. Gropl, A. Hildebrandt, R. Hussong, E. Lange, N. Pfeifer, O. Schulz-Trieglaff, A. Zerck, K. Reinert, and O. Kohlbacher. Openms-an open-source software framework for mass spectrometry. *BMC Bioinformatics*, 9, 2008.

[166] J. D. Jaffe, D. R. Mani, K. C. Leptos, G. M. Church, M. A. Gillette, and S. A. Carr. Pepper, a platform for experimental proteomic pattern recognition. *Molecular & Cellular Proteomics*, 5(10):1927–41, 2006.

[167] J. Q. Zhang, E. Gonzalez, T. Hestilow, W. Haskins, and Y. F. Huang. Review of peak detection algorithms in liquid-chromatography-mass spectrometry. *Current Genomics*, 10(6):388–401, 2009.

[168] M. Katajamaa and M. Oresic. Processing methods for differential analysis of lc/ms profile data. *BMC Bioinformatics*, 6:179, 2005.

[169] M. Katajamaa, J. Miettinen, and M. Oresic. Mzmine: toolbox for processing and visualization of mass spectrometry based molecular profile data. *Bioinformatics*, 22(5):634–636, 2006.

[170] F. F. Gonzalez-Galarza, C. Lawless, S. J. Hubbard, J. Fan, C. Bessant, H. Hermjakob, and A. R. Jones. A critical appraisal of techniques, software packages, and standards for quantitative proteomic analysis. *Omics-a Journal of Integrative Biology*, 16(9):431–442, 2012.

[171] John Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 139–177, 1982.

[172] Vera Pawlowsky-Glahn and Antonella Buccianti. *Compositional data analysis: theory and applications*. Wiley, 2011.

[173] John Vincent Carlis and Joseph D. Maguire. *Mastering data modeling : a user-driven approach*. Addison-Wesley, Boston, 2001.

[174] A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A. M. Michon, C. M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M. A. Heurtier, R. R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, and G. Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.

[175] Y. V. Karpievitch, T. Taverner, J. N. Adkins, S. J. Callister, G. A. Anderson, R. D. Smith, and A. R. Dabney. Normalization of peak intensities in bottom-up ms-based proteomics using singular value decomposition. *Bioinformatics*, 25(19):2573–80, 2009.

[176] P. A. Rudnick, K. R. Clauser, L. E. Kilpatrick, D. V. Tchekhovskoi, P. Neta, N. Blonder, D. D. Billheimer, R. K. Blackman, D. M. Bunk, H. L. Cardasis, A. J. Ham, J. D. Jaffe, C. R. Kinsinger, M. Mesri, T. A. Neubert, B. Schilling, D. L. Tabb, T. J. Tegeler, L. Vega-Montoto, A. M. Variyath, M. Wang, P. Wang, J. R. Whiteaker, L. J. Zimmerman, S. A. Carr, S. J. Fisher, B. W. Gibson, A. G. Paulovich, F. E. Regnier, H. Rodriguez, C. Spiegelman, P. Tempst, D. C. Liebler, and S. E. Stein. Performance metrics for liquid chromatography-tandem mass spectrometry systems in proteomics analyses. *Molecular & Cellular Proteomics*, 9(2):225–41, 2010.

[177] W. X. Schulze and B. Usadel. Quantitation in mass-spectrometry-based proteomics. *Annual Review of Plant Biology, Vol 61*, 61:491–516, 2010.

[178] Ian Sommerville. *Software engineering*. International computer science series. Pearson/Addison-Wesley, Boston, 7th edition, 2004.

[179] Thomas H. Cormen. *Introduction to algorithms*. MIT Press, Cambridge, Mass., 2009.

[180] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22:79–86, 1951.

[181] S. Orchard, P. A. Binz, C. Borchers, M. K. Gilson, A. R. Jones, G. Nicola, J. A. Vizcaino, E. W. Deutsch, and H. Hermjakob. Ten years of standardizing proteomic data: A report on the hupo-psi spring workshop. *Proteomics*, 12(18):2767–2772, 2012.

[182] J. Bishop and N. Horspool. Cross-platform development: Software that lasts. *Computer*, 39(10):26–+, 2006.

[183] Stuart Coles. *An introduction to statistical modeling of extreme values*. Springer, London ; New York, 2001.

[184] L. Zhang, Q. Y. Wei, L. Mao, W. B. Liu, G. B. Mills, and K. Coombes. Serial dilution curve: a new method for analysis of reverse phase protein array data. *Bioinformatics*, 25:650–654, 2009.

[185] Alfred O Nier and Earl A Gulbransen. Variations in the relative abundance of the carbon isotopes. *Journal of the American Chemical Society*, 61:697–698, 1939.

[186] K. N. Kapoor, D. T. Barry, R. C. Rees, I. A. Dodi, S. E. B. McArdle, C. S. Creaser, and P. L. R. Bonner. Estimation of peptide concentration by a modified bicinchoninic acid assay. *Analytical Biochemistry*, 393:138–140, 2009.

[187] J. Rappsilber, Y. Ishihama, and M. Mann. Stop and go extraction tips for matrix-assisted laser desorption/ionization, nanoelectrospray, and lc/ms sample pretreatment in proteomics. *Analytical Chemistry*, 75:663–670, 2003.

Appendix A

Acronyms

Care has been taken in this thesis to minimize the use of acronyms, but this cannot always be achieved. The following table contains a list of acronyms and their meaning.

Table A.1: Acronyms

Acronym	Meaning
2D-PAGE	two (2) Dimensional PolyAcrylamide Gel Electrophoresis
AUC	Area Under the Curve
CDC	Center for Disease Control
CID	Collision Induced Dissociation
CPTAC	Clinical Proteomic Tumor Analysis Consortium
CPU	Central Processing Unit
CV	Coefficient of Variation
DC	Direct Current
DNA	DeoxyRibonucleic Acid
DRC	Dynamic Range Compression
DTW	Dynamic Time Warping
EM	Expectation-Maximization
emPAI	exponentially modified Protein Abundance Index

Continued on next page

Table A.1 – continued from previous page

Acronym	Meaning
ESI	ElectroSpray Ionization
FASTA	FAST-All
ETD	Electron Transfer Dissociation
FDR	False Discovery Rate
FFE	Free Flow Electrophoresis
FFT	Fast Fourier Gransform
FT-ICR	Fourier Transform Ion Cyclotron Resonance
GUI	Graphical User Interface
HCD	Higher Collision Dissociation
HGP	Human Genome Project
HPLC	High Performance Liquid Chromatography
ICAT	Isotope-Coded Affinity Tag
IEX	Ion EXchange
iLFRQ	intensity-based Label Free Relative quantification
iTRAQ	isobaric Tags for Relative and Absolulte Quantification
IEF	IsoElectric Focusing
IPG	Immobilized pH Gradient
ISB	Institute for Systems Biology
ISO	International Standards Organization
IUPAC	International Union of Pure and Applied Chemistry
JRAP	Java Random Access Parser
JVM	Java Virtual Machine
KL	Kullback-Leibler similarity measure
LOESS	LOcally Estimated Scatterplot Smoothing

Continued on next page

Table A.1 – continued from previous page

Acronym	Meaning
LOWESS	LOcally WEighted regrESSion
MA	Minus vs Average
MALDI	Matrix Assisted Laser Desorption/Ionization
MS	Mass Spectrometry
MS/MS	Tandem Mass Spectrometry
mTRAQ	Mass Differential Tags for Relative and Absolute Quantification
MudPIT	Multi-dimensional Protein Identification Technology
m/z	mass to charge (z) ratio
NASF	Normalized Spectral Abundance Factor
NIH	National Institutes of Health
OPML	Oral Premalignant Lesion
OSCC	Oral Squameous Cell Carcinoma
OTC	Office for Technology Commercialization
PAI	Protein Abundance Index
PTDW	Parametric Dynamic Time Warping
PIN	Proximity-based Intensisty Normalization
PLS	Partial Least Squares
PEV	Pooled Estimate of Variance
PTM	Post Translation Modification
QTOF	Quadrupole Time Of Flight
RF	Radio Frequency
ROC	Receiver Operator Curve
RP	Reverse Phase

Continued on next page

Table A.1 – continued from previous page

Acronym	Meaning
RSD	Relative Standard Deviation
SCX	Strong Cation eXchange
S.D.	Standard Deviation
SDS-PAGE	Sodium Dodecyl Sulfate Polyacrylamide Gel Electrophoresis
SIn	Normalized Spectral Index
SILAC	Stable Isotope Labeling by Amino acids in Culture
S/N	Signal to Noise ratio
SQL	Structured Query Language
StAX	Streaming API for XML
TMT	Tandem Mass Tag
TOF	Time Of Flight
TPP	Trans-Proteomic Pipeline
UPS1	Universal Protein Standard 1
UROP	Undergraduate Research OPportunity
VSN	Variance Stabilization Normalization
XC	eXtracted Chromatogram
XIC	eXtracted Ion Chromatogram

Appendix B

Additional Background

B.1 Peptide Isotopes

Atoms of the same elements can have different numbers of neutrons. The different possible versions of the element are called isotopes [185]. When these elements are incorporated into amino acids, isotopes exist in ratios corresponding to their abundances in nature [86]. Amino acids, and their post translational modifications predominantly comprise six elements, hydrogen, carbon, nitrogen, oxygen, phosphorous, and sulfur. Table B.1 lists the natural abundances of these six elements in nature [86].

Elements		Abundances (%)	Masses
Hydrogen	H^{1}	99.99	1.0078
	H^{2}	0.01	2.0141
Carbon	C^{12}	98.91	12.0000
	C^{13}	1.09	13.0034
Nitrogen	N^{14}	99.63	14.0031
	N^{13}	0.37	15.0001
Oxygen	O^{16}	99.76	15.9949
	O^{17}	0.04	16.9991
	O^{18}	0.20	17.9992
Phosphorous	P^{31}	100.00	30.9738
Sulphur	S^{32}	95.02	31.9721
	S^{33}	0.76	32.9715
	S^{34}	4.22	33.9676

Table B.1: Naturally Occurring Isotopic Abundances

Appendix C

Materials and Methods

This appendix describes in detail the materials and methods used in evaluating RIPPER/PIN. Describing the experimental procedures is divided into five parts.

- Section C.1 Formulas - Details mathematical equations used in statistical analyses.
- Section C.2 Analytical Methods - Details common normalization methods, as well as, methods for generating XCs and Minus vs. Average plots.
- Section C.3 Data Sets - Describes how each of the five data sets were generated.
- Section C.4 Evaluation Methods - Describes in detail how each part of the evaluation was performed.
- Section C.5 Statistical Methods for Biomarker Panel Discovery - Provides detail for the statistical methods used for identifying significant peptide signals within the OPML vs. OSCC data set.

C.1 Formulas

C.1.1 General

Unless otherwise specified, all statistical analyses were performed using the R statistical package, version 2.14-0 2011-10-31, R.app 1.41. Two sample t-tests were conducted using

the function t.test specifying an alternate hypothesis = less and the default confidence level = 0.95.

C.1.2 Statistical Formulas

The following tables list statistical formulas used in this dissertation. Table C.1 list general formulas and Table C.2 lists the formulas used in Minus vs Average plots.

No.	Variable	Equation	Description
C.1.1	Inter-run Mean	$\mu_i = \frac{\sum_{r=1}^{n} x_{i,r}}{n}$	where $x_{i,r}$ is the intensity of peptide signal x_i in run r and n is the number of runs r
C.1.2	Intra-run Mean	$\mu = \frac{\sum_{i=1}^{p} x_{i,r}}{p}$	where $x_{i,r}$ is the intensity of peptide signal x_i in run r and p is the number peptides in run r
C.1.3	Inter-run Variance	$\sigma_i^2 = \frac{\sum_{r=1}^{n} x_{i,r}^2}{n} - \left(\frac{\sum_{r=1}^{n} x_{i,r}}{n}\right)^2$	where $x_{i,r}$ is the intensity of peptide signal x_i in run r and n is the number of runs r
C.1.4	Standard Deviation	$\sigma = \sqrt{\sigma^2}$	where σ^2 is computed in equation C.1.3
C.1.5	Pooled Estimate of Variance	$PEV = \frac{(n_1 - 1)\,\sigma_{m_1}^2 - (n_2 - 1)\,\sigma_{m_2}^2}{(n_1 - 1) + \ (n_2 - 1)}$	where σ^2 is computed in equation C.1.3, n is the number of runs
C.1.6	Coefficient of Variation	$cv = \frac{\sigma}{\mu}$	where σ is computed in equation C.1.3 and μ is computed in equation C.1.2
C.1.7	Correlation Coefficient	$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$	where X and Y are the two variables compared and n is the number of peptide signals

Table C.1: Statistical Equations

No.	Variable	Equation	Description
C.2.1	Minus	$m_i = log_2 \left(x_{i,r} / x_{i,r+1} \right)$	where $x_{i,r}$ is the intensity of peptide signal x_i in run r and $x_{i,r}$ is the intensity of peptide signal x_i in run $r+1$
C.2.2	Average	$a_i = log_2 \left(x_{i,r=1} * x_{i,r+1} \right) / 2$	where $x_{i,r=1}$ is the intensity of peptide signal x_i in run r and $x_{i,r}$ is the intensity of peptide signal x_i in run $r+1$
C.2.3	Convoluted Intensity	$m_i^{'} = m_i - m_i^*$	where m_i^* is a) the predicted peptide intensity from the regression normalization method or b) $/mu$, which is the mean, in the central tendency normalization method
C.2.4	Norm. Intensity (1)	$x_{i,r}^{'} = 2^{(m_i^{'} + 2a_r)/2}$	where $x_{i,r}$ is the deconvoluted intensity of peptide signal x in run r, a is the average computed in equation 2, and $m_i^{'}$ is computed in equation C.2.3
C.2.5	Norm. Intensity (2)	$x_{i,r+1}^{'} = 2^{-(m_i^{'} - 2a_r + 1)/2}$	where $x_{i,r+1}$ is the deconvoluted intensity of peptide signal x in run $r+1$, a is the average computed in equation C.2.2, and $m_i^{'}$ is computed in equation C.2.3

Table C.2: Minus vs Average Plot Normalization

C.2 Analytical Methods

C.2.1 Extracted Chromatogram Plots

Extracted chromatograms (XCs) are similar to the more familiar extracted ion chromatograms (XICs), but contain a subset of the total ion chromatogram. A *Signal > SN* XC contains signal above the SN threshold and A *Peptide Signal* XC contains signal stemming from peptides. RIPPER extracted the necessary data (see Chapter 6) and stored them in .csv files. These .csv file were used as input into an R script which parsed out the data by run identifier. The R script first plotted a scatterplot with the summed intensities along the y-axis and scan (retention time) along x-axis. Then, the R lowess function (locally-weighted polynomial smoother) drew the regression lines. I set the lowess smoothing span parameter set to 0.05. All other lowess parameters were defaulted. The following listing contains a generic R script for generating a plot containing two XCs.

```
#-------------------------------------------------------------------------------
# Function for XC
#  Arguments:  r matrix of retention times
#              s matrix of signal greater than the signal to noise threshold
#              p matrix of peptide signal intensities
#          title will be dsiplayed on the figure
#        pngfile location and name or the resulting .png file
#           xmin minimum x-axis value
#           xmax maximum x-axis value
#           ymin minimum y-axis value
#           ymax maximum y-axis value
#-------------------------------------------------------------------------------
plotXC <- function(r, s, p, pngfile, title, xmin, xmax, ymin, ymax) {
  # Location and name of unnormalized data
  png(pngfile)
  # y-axis label
  ylab = "Intensity"
  ## size of the dots...
  c = 1.3
```

```
  ## Main title
  title(main = expression(paste(title)),cex.main=1.5)
  ## Draw the lowess lines
  lines(lowess(r,s,f=fFactor), xlim=c(xmin,xmax), ylim=c(ymin,ymax),
        col='black', lwd=lineWidth, lty = 2)
  lines(lowess(r,p,f=fFactor), xlim=c(xmin,xmax), ylim=c(ymin,ymax),
        col='black', lwd=lineWidth, lty = 1)
  ## Plot the legend
  legend("topright",legend = c("Above SN", "Peptide Signal"),
        cex = 1, lty=c(2, 1))
  ## Turn device off
  dev.off()
}
```

C.2.2 Minus vs. Average Plots

In microarray studies (see Chapter 3), traditional plotting methods first log transforms R and G intensities and then constructs a scatterplot of $\log_2$R vs $\log_2$ G intensities [160]. Log transformation serves several purposes:

- it makes variation and ratios of logged intensities is less dependent on absolute magnitude;
- it evens out highly skewed distributions;
- it gives a more realistic sense of variation.

The following is the R Function written to plot MA plots.

```
#-------------------------------------------------------------------------------
# Function for MA Plots
#  Arguments:  x  matrix containing the intensities of each peptide signal
#                  the first column is the identifier (charge m/z value pair)
#                  the remaining columns contain the intensity for that charge -
#                  m/z value pair
#           title will be displayed on the figure
#         pngfile location and name or the resulting .png file
#-------------------------------------------------------------------------------
```

```
plotMA ← function(x, title, pngFile) {
  # Set up the colors and line types to be plotted.
  #   Note: The maximum number of runs in experiments plotted is 6. If more
  #           runs are are going to be plotted, these two arrays will need to
  #           be expanded.
  #
  colr ← c("black", "grey20", "grey30", "grey40","grey50")
  ltype ←c(1,2,3,4,5,6)
  # Convert the input into a matrix data type.
  y ← as.matrix(x)
  # Compute m and a values the first pair of runs. Here I arbitrarily select
  #   the first run as the reference run.
  m ← log2(y[,1]/y[,2])
  a ← log2(y[,1]*y[,2])/2
  # Compute m and a values for the remain pairs for runs. Again, the reference
  #   run is always the first run.
  for (i in 3:ncol(y)) {
        m ← cbind(m,(log2(y[,1]/y[,i])))
        a ← cbind(a,log2(y[,1]*y[,i])/2)
  }
  # Set up the minimum and maximum x and y axis values for the plot. Note that
  #   here the y values are hard coded. This is to ensure that the overlaid plots
  #   are on the same scale. The y values may need to be changed, depending on
  #   experimental values.
  xmin ← min(a)
  xmax ← max(a)
  ymin = -5
  ymax = 5
  # Open png device file
  png(pngFile,640,480)
  # Plot the M vs A scatter plot points
  for (i in 1:ncol(a)) {
        plot(a[,i],m[,i],  pch=".", xlim=c(xmin,xmax), ylim=c(ymin,ymax),
                col="black", cex = 1, xlab="A", ylab="M")
        par(new=TRUE)
  }
  # Plot the M vs A lowess lines. Note that the f value is set at .4, which is
  #  the value set by Dudoit, et al., 2002
  for (i in 1:ncol(a)) {
        lines(lowess(a[,i],m[,i],f=.4), xlim=c(xmin,xmax), ylim=c(ymin,ymax),
```

```
            col=colr[i], lwd=5, lty=ltype[i])
        par(new=TRUE)
    }
    # Plot the horzontal line at y = 0 and main title
    abline(h=0)
    title(title, cex.main = 2)
    # Turn off device for png file
    dev.off()
}
```

C.2.3 Normalization Methods

C.2.3.1 Linear regression

Linear regression normalization is performed by applying least squares regression on MA scatter plots using a pairwise iterative algorithm [117][118]. Because each experiment contains more than two runs, the iteration process was performed at least twice because the difference between the mean of all intensity ratios from the previous iteration was < 0.005.

```
#-------------------------------------------------------------------------------
# Function to normalize by cyclic regression of MA transfomed data
#  Arguments: x  matrix containing the intensities of each peptide signal
#       iterations number of iterations - default if none supplied
#-------------------------------------------------------------------------------
normalizeCyclicRegrMA <- function (x,  iterations = 3) {
    # Convert to matrix
    y <- as.matrix(x)
    # Iterate specified number of times
    for (k in 1:iterations) {
        # Process each pair of runs in a round robin fashion
        for (i in 1:(ncol(y) - 1)) {
            for (j in (i + 1):ncol(y)) {
                # Compute average and minus
                a <- (log2(y[,j]*y[,i]))/2
                m <- log2(y[,j]/y[,i])
                # Perform linear regression
                fit <- lm(m~a)
```

```
                    # Adjust based on fit of linear regression
                    mPrime <- m - fitted(fit)
                    # Transform back
                    y[,j] <- 2^(a + mPrime/2)
                    y[,i] <- 2^(a - mPrime/2)
                }
        }
}
```

C.2.3.2 Cyclic LOESS

Cyclic LOESS normalization is performed in R using the normalizeCyclicLoess function found in the limma package [164]. The data are first $\log_2$ transformed (as required by normalizeCyclicLoess), submitted to normalizeCyclicLoess using default parameters, and untransformed to return them to normal scale.

C.2.3.3 Quantile Normalization

Quantile normalization is performed in R using the normalizeQuantile function found in the limma package [164] using default parameters. In this case, data is not first $\log_2$ transformed.

C.2.3.4 Reference Run

Reference run is performed by selecting a run as a reference and all runsâĂŹ peptide signal intensities normalized to the selected run. The median of matched peptide signal intensity ratios is used as a normalization factor [118]. I arbitrarily selected the first replicate as the reference run.

```
#-------------------------------------------------------------------------
# Function to normalize by reference run
#  Arguments: x  matrix containing the intensities of each peptide signal
#-------------------------------------------------------------------------
normalizeRefRun <- function (x)
{
```

```
    # Convert to matrix
    y ← as.matrix(x)
    # Process each run against run 1 (the reference run)
     for (i in 2:ncol(y)) {
                y[,i] ← y[,i]/median(y[,i]/y[,1])
    }
}
```

C.2.3.5 Median Scale

Median scale (central tendency) is performed by scaling the peptide signal intensity values within each run by the median of all peptide signal intensities in that run.

```
#-------------------------------------------------------------------------------
# Function to normalize by central tendency
#  Arguments:  x  matrix containing the intensities of each peptide signal
#-------------------------------------------------------------------------------
normalizeCentralTendency ← function (x)
{
    # Convert to matrix
    y ← as.matrix(x)
    # Scale each value by run median
     for (i in 1:ncol(y)) {
      y[,i] ← y[,i]/median(y[,i])
    }
}
```

C.3 Data Sets

This section describes in detail the experimental procedures for generating the two types of data sets. While the salivary endogenous peptide data sets (2,3, & 5) are technically different in composition than the combined bradykinin and salivary endogenous peptide data sets (4), their generation protocols share many steps. Therefore, their protocol descriptions will be combined in sub-section C.3.1. Additionally, sub-section C.3.2 describes the generation protocol for the combined UPS1 and yeast (CPTAC Study 6)

data set (4).

C.3.1 Salivary Endogenous Peptides

The salivary endogenous peptides data sets, including the combined bradykinin and salivary endogenous peptide data set, were generated in the Griffin lab at the University of Minnesota. The protocol for generating these data sets contains the following common steps: These steps include:

- sample collection,
- sample preparation,
- HPLC-ESI-MS/MS,
- and peptide and protein identification.

. Within sample collection and sample preparation steps, additional details are given for each specific data set.

C.3.1.1 Sample Collection

This sub-section first describes the general protocol for collecting saliva from healthy and diseased volunteers. It then provides details for each data set where the sample collection protocol differs from the general saliva collection protocol.

General Protocol

All samples were collected according to a protocol approved by the University of Minnesota Institutional Review Board. Subjects included in the study were all non-smokers free of confounding conditions: periodontal disease, auto-immune disease, a prior history of diseases of the oral mucosa, or current use of potentially confounding medications. Donors refrained from eating or drinking for at least 1 hour prior to donation. After a

water rinse, donors allowed saliva to collect in their mouths before gently expectorating into a sterile 15 or 50 ml conical tube.

Data Set Specific Protocols

- Serial Dilution - fresh saliva was collected from a single donor
- Instrument Variability - fresh saliva was collected from a single donor
- Sample Variability - fresh saliva was collected from a single donor
- OPML vs. OSCC - fresh saliva was collected 17 patients with pathologically confirmed OPMLs and 18 patients with pathologically confirmed OSCC

C.3.1.2 Sample Preparation

This sub-section first describes the general protocol for isolating endogenous peptides from saliva. It then provides details for each data set where the endogenous peptide isolation protocol differs from the general endogenous peptide isolation protocol.

General Protocol

Clarified saliva was prepared from fresh whole saliva samples by centrifuging at 3000×g at 4^{o}C for 15 minutes, followed by 16,100×g at 4^{o}C for 1 minute. The supernatant was mixed in a 10:1 ratio with denaturing buffer consisting of 4% SDS, 100 mM dithiothreitol and 100mmol/l Tris, pH 7.4. The samples were boiled for 5 minutes, cooled to room temperature, then added to a centrifugal filter (Amicon Ultra, 0.5 ml, 10 kDa, Millipore). Two hundred microliters of buffered urea (8 mol/l urea with 100 mmol/l tris pH 8.5) was added to the sample, and this mixture was centrifuged at 14,000×g at room temperature for 40 minutes. An additional 200 μl of buffered urea was added and the sample was centrifuged at 14,000×g at room temperature for 40 minutes. The filters were discarded and the collected peptides were alkylated, by addition of iodoacetamide in buffered urea

to 50mmol/l, in the dark for 20 minutes. MCX cleanup was performed by diluting the samples to 3 ml with 2% formic acid and H2O to pH $\leq$3. The MCX columns (Oasis 3 cc, 60 mg, Waters Corp.) were equilibrated with 2 ml of 1:1 methanol:water followed by addition of the entire sample, washing with 3 ml of 0.1% formic acid, 2 ml of methanol, and elution with 1 ml of 95% methanol, 5% ammonium hydroxide. The eluted peptides were dried in a speed-vac, redissolved in water, and quantified by a modified BCA assay (Thermo Scientific) [186], using trypsin-digested saliva as a standard. Three micrograms of peptides were further purified and concentrated using the STAGE-tip protocol [187].

Data Set Specific Protocols

- Serial Dilution - Saliva from the single donor was processed once to isolate the endogenous peptides resulting. We placed increasing amounts (0.5μg, 1.0μg, 1.5μg, 2.0μg, 2.5μg, and 3μg) of same isolated peptide mixture into six individual vials. Then, to each vile, we add 500 fmol of bradykinin.

- Instrument Variability - Saliva from the single donor was processed once to isolate the endogenous peptides. We placed sufficient sample of isolated peptide mixture into a single autosampler vial.

- Sample Variability - Saliva from the single donor was first separated into three aliquots and then each aliquot was processed once in parallel to isolate the endogenous peptides. We then placed each processed sample into its own individual autosampler vial.

- OPML vs. OSCC - Saliva from each of the 35 donors were processed in parallel to isolate the endogenous peptides. We then placed each sample into its own autosampler vial (one for each donor).

C.3.1.3 HPLC-ESI-MS/MS

While HPLC-ESI-MS/MS processing follows protocols described in previous publications[27, 54], I report their descriptions here for completeness. Peptide samples were analyzed by online HPLC-ESI-MS on an LTQ Orbitrap-XL mass spectrometer (Thermo Scientific) equipped with an Eksigent (Eksigent Technologies) 1DLC nanoflow system and a MicroAS autosampler. The column used was an in-house pulled capillary tip of 100 μm inner diameter, packed to 13 cm with Magic C18AQ 5-μm, 200 ÃĚ pore particles (Michrom Bioresources, Inc.). Samples were redissolved in an aqueous solution containing 2% ACN with 0.1% formic acid, and separated by a 2âĂŞ40% ACN gradient in 0.1% formic acid over 60 min at 250 nl/minute. The column was mounted in a nanospray source directly in line with an LTQ Orbitrap- XL mass spectrometer (ThermoFisher Scientific). Spray voltage was at 1.75 kV, and the heated capillary was maintained at 160^{o}C. The orbital trap was set to acquire survey mass spectra (m/z 360-1800) with a resolution of 60,000 at m/z 400 with a target value set to 1E6 ions or 500 ms. The 5 most intense ions from the full scan were selected for fragmentation by collision-induced dissociation (normalized collision energy, 35%) in the LTQ ion trap with automatic gain control settings of 5000 ions or 100 ms concurrent to full-scan acquisition in the orbital trap. For enhanced mass accuracy, the lock mass option was enabled for real-time calibration using the following polysiloxane peaks of m/z 371.1012, 445.1200, and 519.1388. Precursor ion charge state screening was enabled, and all unassigned charge states as well as singly charged species were rejected. We used dynamic exclusion set to a maximum of 500 entries with a maximum retention period of 90 seconds and mass window of -0.6 to 1.2 amu. Data were acquired using Xcalibur software.

C.3.1.4 Protein and Peptide Identification

Tandem mass spectra were searched using Sequest v27.0 against an NCBI human database v200806 (70,711 entries) with concatenated reversed sequences. The search parameters

included variable oxidation of methionine, fixed acetamidylation and no enzyme specificity. The search results were filtered to include only peptides identified at 95% confidence with a charge 2+, 3+ and 4+, and with a 9 ppm precursor mass accuracy. This produced a peptide FDR of $< 1\%$ for each data set 1, 2, and 3.

C.3.2 UPS1 and Yeast (CPTAC Study 6)

C.3.2.1 Data Acquisition

National Cancer InstituteâĂŹs Clinical Proteomic Tumor Analysis Consortium (CPTAC) networkâĂŹs Study 6 [28] data set LTQ-XL-OrbitrapP@65 from: http://cptac.tranche.proteomecommons.org/study6.html, hash tag: c9wCnilmFqcPodCpUcaaprfu8865 M5jrvHW6YrWwksRKj95BvjNcfltPn9zD6D7QLyoP8E+hieqygIscRcaBrmFSFo vAAAAAAAAAjdw==. Because data acquisition for this study was performed in profile mode, we converted the .raw files to mzXML files using msConvert version 3.0.3364 specifying centroid = true.

C.3.2.2 Protein and Peptide Identification

Tandem mass spectra were extracted, charge state deconvoluted and deisotoped by msConvert. All MS/MS samples were analyzed using Sequest version 27, rev 12 (Thermo Scientific). Sequest was set up to search a database constructed in the following manner. First, I downloaded the yeast uniprot FASTA database on February 3, 2012 from ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/proteomes/YEAST.fasta.gz and the cRAP contaminant FASTA database on February 28, 2012 from ftp://ftp.thegpm.org/fasta/cRAP. Because the UPS entries in the cRAP database were outdated, I replaced the cRAP UPS entries with entries from an updated UPS FASTA database downloaded on February 22, 2012 from http://www.sigmaaldrich.com/life-science/proteomics/mass-spectrometry/ups1-and-ups2-proteomic.html. I then concatenated YEAST and cRAP FASTA databases creating a combined forward database. The

entire database was then reversed using a perl script (Matrix Science) and concatenated to the forward database. The resulting database contained 13,532 proteins. For the C vs. E experiment, Sequest was searched with a fragment ion mass tolerance of 1.00 Da Oxidation of methionine was specified in Sequest as a variable modification. Scaffold, version Scaffold_3.6.1, (Proteome Software, Inc.) was used to validate MS/MS based peptide and protein identifications. Peptide identifications were accepted if they met minimum criteria (7 ppm, 1 ntt, and 6 min. length) if they could be established at greater than 90.0% probability as specified by the Peptide Prophet algorithm [104] Protein identifications were accepted if they could be established at greater than 80.0% probability and contained at least 1 identified peptide. Protein probabilities were assigned by the Protein Prophet algorithm [105]. Proteins that contained similar peptides and could not be differentiated based on MS/MS analysis alone were grouped to satisfy the principles of parsimony. The resulting listâĂŹs computed false discovery rate (FDR) [103] within Scaffold was 1.4% at the protein level and 0.1% at the peptide level.

C.4 Evaluation Methods

C.4.1 SN vs Peptide Signal

Using reports generated by RIPPER, I used R to plot the extracted chromatograms (XCs). In the R script, I first read in *experimentID*_extractedScnIntensities.csv (I will refer to this as in.csv for brevity). For the XC of Serial Dilution > SN Threshold, I used function plotXC (see Section C.2.1) with r = the column RT in the file in.csv, s = sumSNIntensity in the file in.csv, and p = the column sumPeakIntensity in the file in.csv.

C.4.2 Systematic Bias

For each data set, I used R to read in the *experimentID*_peptideSignalREPORT.csv report (which contained the un-normalized intensities) and the *experimentID*_normalized

PeptideSignalREPORT.csv (which contained the PIN normalized intensities) files generated by RIPPER. I applied the normalization methods (see Section C.2.3) to the un-normalized intensities. Then, for the un-normalized and normalized data, I generated Minus vs Average Reports using function PlotMA in Section C.2.2.

C.4.3 Complex Variabilty

Using reports generated by RIPPER, I used R to plot the XCs. In the R script, I first read in *experimentID* _extractedScnIntensities.csv (I will refer to this as in.csv for brevity). For the plotting the XCs, I used a variation of function plotXC without the s parameter (see Section C.2.1) with r = the column RT in the file in.csv, and p = the column sumPeakIntensity in the file in.csv.

C.4.4 Internal Standard (Spike-in)

For Figure 7.9, I used RIPPER generated reports as input in to R to plot the XCs. In the R script, I first read in *experimentID* _extractedScnIntensities.csv (I will refer to this as in.csv for brevity). For plotting the un-normalized XC, I used a variation of function plotXC without the s parameter (see Section C.2.1) with r = the column RT in the file in.csv, and p = the column sumPeakIntensity in the file in.csv. For plotting the medican scale normalized XC, I used a variation of function plotXC without the s parameter (see Section C.2.1) with r = the column RT in the file in.csv, and p = the column sumPeakIntensity in the file in.csv.However, here, p was first normalized by median scale (see Section C.2.3). For plotting the PIN normalized XC, I used a variation of function plotXC without the s parameter (see Section C.2.1) with r = the column RT in the file in.csv, and p = the column sumNormalizedIntensity in the file in.csv.

For Figure 7.10, I used Microsoft Excel® chart function and requested the display of linear regression lines and equations.

C.4.5 Repeatability

For each data set, I used R to read in the *experimentID*_peptideSignalREPORT.csv report (which contained the un-normalized intensities) and the *experimentID*_normalized PeptideSignalREPORT.csv (which contained the PIN normalized intensities) files generated by RIPPER. I applied the normalization methods (see Section C.2.3) to the un-normalized intensities. I then used R to compute the CVs and the PEVs according to the equations in Section C.1.2.

C.4.6 Overfitting

For the CPTAC Study 6 data set, I first used Scaffold to identify peptides and proteins with parame ters listed in Table C.3. I then match peptide signals from the *experimentID*_normalizedPeptideSignalREPORT.csv (which contained the PIN normalized intensities) files generated by RIPPER to Scaffold identified peptides and proteins. Next, I applied a two-sided t-test (in Microsoft Excel®) and selected those identified peptide signals with a p-value < 0.01. If the reference identification contained UPS, I counted it as a hit for a UPS1 protein and peptide and then removed duplicates. Otherwise, I counted it as a hit for a yeast protein and peptide.

Parameter	Value
Experiment	SEQUEST_RESULTS_CE
Conversion	MSCONVERT
Version	V
Charge States Calculated	TRUE
Deisotoped	TRUE
Textual Annotation	
Database Set	1 Database
Database Name	the Yyeast_uniprot_20120203_plus_crap_cont_20120229_minus_ups_plus_ups1_ups2_standard_fasta_20120222_uniprotformat database
Version	1
Taxonomy	All Entries
Number of Proteins	13532
Does database contain common contaminants?	Yes
Search Engine Set	1 Search Engine
Search Engine	Sequest
Version	27, rev. 12
Samples	All Samples
Fragment Tolerance	1.00 Da (Monoisotopic)

Parent Tolerance	0.072 Da (Monoisotopic) (klc_031308p_cptac_study6_6E004),
	0.073 Da (Monoisotopic) (klc_031308p_cptac_study6_6C008_080316072741),
	0.076 Da (Monoisotopic) (klc_031308p_cptac_study6_6E004_080318120052),
	0.087 Da (Monoisotopic) (klc_031308p_cptac_study6_6C008),
	0.095 Da (Monoisotopic) (klc_031308p_cptac_study6_6E004_080316165744),
	0.11 Da (Monoisotopic) (klc_031308p_cptac_study6_6C008_080318023052)
Fixed Modifications	
Variable Modifications	+16 on M (Oxidation)
Database	the Yyeast_uniprot_20120203_plus_crap_cont_20120229
	_minus_ups_plus_ups1_ups2_standard_fasta_20120222
	_uniprotformat database (1.0, 13532 entries)
Digestion Enzyme	Trypsin
Max Missed Cleavages	2
Scaffold Version	Scaffold_3.6.1
Peptide Thresholds	90.0% minimum
Protein Thresholds	80.0% minimum and 1 peptides minimum

Table C.3: Scaffold Parameters

C.5 Statistical Methods for Biomarker Panel Discovery

Briefly, the statistical method works as follows. Case versus control labels are randomly permuted for each peptide signal and t-statistic calculated for each permuted peptide signal. This is repeated a large number of times, for example, 1000, to determine a null distribution of the test statistic, and provides an FDR calculation. Peptide signals are considered significant if they have a test statistic larger in absolute value than the critical value identified to control the FDR. A predictive model is constructed for candidates and AUC for the receiver operator curve (ROC) calculated and the panel of best performing candidates outputted to the user. Ultimately, using the list of significant peptide signals, a predictive model is created for differentiating between case and control using the Lasso method. For the peptide signal panel an area under the ROC curve is estimated using a leave-one-out cross-validation procedure.