2014-05-01

Elucidation and Improvement of Algorithms for Mass Spectrometry Isotope Trace Detection

Robert Anthony Smith
Brigham Young University - Provo

Elucidation and Improvement of Algorithms for Mass Spectrometry Isotope Trace Detection

Rob Smith

A dissertation submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Dan Ventura, Chair
John T. Prince
Mark Clement
Sean Warnick
Bill Barrett

Department of Computer Science

Brigham Young University

May 2014

ABSTRACT

Elucidation and Improvement of Algorithms for Mass Spectrometry Isotope Trace Detection

Rob Smith
Department of Computer Science, BYU
Doctor of Philosophy

Mass spectrometry facilitates cutting edge advancements in many fields. Although instrumentation has advanced dramatically in the last 100 years, data processing algorithms have not kept pace. Without sensitive and accurate signal segmentation algorithms, the utility of mass spectrometry is limited. In this dissertation, we provide an overview and analysis of mass spectrometry data processing. A tutorial to ease the learning curve for those outside the field is provided. We draw attention to the lack of critical evaluation in the field and describe the resulting effects, including a glut of algorithm contributions of questionable novel contribution. To facilitate increased critical evaluation, we show the importance of a modular paradigm for mass spectrometry data processing through highlighting the impact of data processing algorithm choice upon experimental results. Our novel controlled vocabulary is presented with the aim of facilitating literature reviews for comparisons. We propose a novel nomenclature and mathematical characterization of mass spectrometry data. We present several novel algorithms for mass spectrometry data segmentation that outperform existing standard approaches. We end with an overview of future research which will continue to advance the state of the art in mass spectrometry data processing.

Keywords: Isotope Trace Detection, Feature Detection, Chromatogram Detection, LC-MS Simulation, Mass Spectrometry

ACKNOWLEDGMENTS

Many thanks to Dan Ventura and John Prince for their invaluable assistance throughout my tenure at BYU. Thanks to my wife Nicky for her unconditional support and patience. Thanks to NSF for their financial support in form of the Graduate Research Fellowship, [DGE-0750759].

Table of Contents

List of Figures viii

List of Tables xi

1 Introduction 1

2 MS Tutorial for Computer Scientists 5

2.1 Background . . . 6

2.2 Results and discussion . . . 6

2.2.1 Relationship of genomics, proteomics, lipidomics, and metabolomics . 6

2.2.2 MS-omics pipeline . . . 9

2.2.3 Data types . . . 16

2.2.4 Data sets . . . 16

2.2.5 Problems of interest . . . 18

2.3 Conclusions . . . 28

2.4 Competing interests and declarations . . . 28

2.5 Author's contributions . . . 28

3 Novel Algorithms and the Benefits of Comparative Validation 29

3.1 Funding . . . 34

4 Controlling for Confounding Variables in MS-omics Protocol: Why Modularity Matters 35

4.1 Key Points . . . 39

5 Clarity in Concepts **40**

5.1 Introduction 41

5.2 Proposed Terms 43

5.2.1 Base Terms 43

5.2.2 Qualifiers 45

5.3 Current Terms and Usage 49

5.3.1 Why Current Official Terms are Incomplete 49

5.3.2 Ambiguous Colloquial Terms 52

5.4 Conclusion 55

5.4.1 Acknowledgements 55

6 LC-MS Mathematical Characterization **56**

6.1 Introduction 57

6.2 Isotope Trace Extraction 58

6.2.1 Extant Objective Functions 61

6.2.2 Proposed Objective Functions 63

6.2.3 Alleviating Current Limitations in Isotopic Trace Extraction 63

6.3 Isotopic Envelope Extraction 64

6.3.1 Extant Objective Functions 65

6.3.2 Proposed Objective Functions 66

6.3.3 Alleviating Current Limitations in Isotopic Envelope Extraction . . . 67

6.4 Correspondence 68

6.4.1 Extant Objective Functions 68

6.4.2 Proposed Objective Functions 69

6.4.3 Alleviating Current Limitations in Correspondence 71

6.5 Conclusion 72

6.6 Acknowledgements 72

7 Mspire-Simulator **73**

7.1 Introduction 74

7.2 Methods 76

7.3 Results 77

7.3.1 Retention Time and Intensity Prediction 77

7.3.2 Feature Shape 79

7.3.3 Drops and Noise 82

7.3.4 Merging Overlapping Peaks 82

7.3.5 MS/MS 83

7.3.6 Modifications 83

7.3.7 Output 83

7.3.8 Parameter Fitting Automation 84

7.3.9 Using the Simulator to Assess Quantitation Performance 84

7.4 Discussion and Conclusion 85

7.5 Acknowledgement 89

8 Java Mass Spectrometry Simulator **90**

8.1 Introduction 91

8.2 Methods 92

8.3 Results 94

9 Massifquant **96**

9.1 Introduction 97

9.2 Methods 99

9.2.1 Description of the Massifquant algorithm 99

9.2.2 Annotation 101

9.2.3 Performance Evaluation 104

9.3 Results 107

9.3.1 Overall Evaluation 107

9.3.2 Evaluation by IT Type 111

9.4 Discussion & Conclusions 111

9.4.1 The evaluation process 113

9.4.2 Algorithm performance 115

9.4.3 Ease of use 116

9.4.4 The use of m/z information in IT detection 117

9.5 Acknowledgement 118

10 Statistical Agglomeration: Peak Summarization for Direct Infusion Lipidomics 119

10.1 Introduction 120

10.2 System and Methods 122

10.2.1 Data 122

10.2.2 Metrics 123

10.3 Algorithms 128

10.3.1 Fixed Ridge Width Method 128

10.3.2 Modified Bin Agglomeration 129

10.3.3 Statistical Agglomeration 129

10.3.4 Xcalibur 133

10.4 Results 133

10.5 Discussion 136

11 Conclusion 142

11.1 Future Work 144

References 146

List of Figures

1.1 An isotopic envelope, the accumulated signal generated by a class of molecule at a given charge state in the sample analyzed via mass spectrometer. 2

2.1 The MS-omics pipeline 8
2.2 Common nomenclature 10
2.3 Deisotoping 13
2.4 Profile and centroid signals 14
2.5 MS correspondence 20
2.6 Baseline subtraction 21
2.7 Feature detection 22
2.8 Dynamic range 25
2.9 Difference detection 27
2.10 Deconvolution 28

3.1 A comparison of the number of papers presenting MS alignment algorithms . 30

4.1 Biomarker differences due to data processing 36

5.1 A graphical representation of the proposed nomenclature 44
5.2 Profile vs. centroid 45
5.3 An integrated isotopic trace 47
5.4 A deisotoped isotopic envelope 48
5.5 A deisotoped molecular envelope 49

6.1 A graphical overview of LC-MS experiments 59

6.2 Explanation of υ 60

6.3 A modified Theil-Sen estimator to characterize one aspect of LC-MS data . . 61

6.4 LC-MS correspondence 69

7.1 The simulation process 78

7.2 Elution peak shape 80

7.3 Isotope trace intensity and m/z variance 81

7.4 Curve fitting results 84

7.5 A simulated BSA run 86

7.6 A qualitative comparison of real and simulated data 87

7.7 A simulated run 88

8.1 JAMSS has a straightforward GUI interface to facilitate parameter selection for MS simulation. 93

9.1 Performance of various isotope trace detection algorithms 108

9.2 A comparison of quantitation errors of various isotope trace detection algorithms 109

9.3 The hand-annotated data set 110

9.4 Performance of various isotope trace detection algorithms by signal intensity 112

9.5 Massifquant isotope trace feature detection on two data sets 116

10.1 Direct infusion lipidomics 121

10.2 The scan combination process 122

10.3 Fixed width segmentation 129

10.4 Modified bin agglomeration segmentation 130

10.5 MBA bin splitting 130

10.6 Statistical agglomeration segmentation 131

10.7 SA bin agglomeration 132

10.8 Normalized true peak distance - m/z 134

10.9 Normalized true peak distance - intensity 135

10.10Number of ridges predicted by method 136

10.11Purity results 136

10.12Normalized mutual information results 137

10.13Sum squared error - intensity 137

10.14Sum squared error - m/z 138

10.15Peak Summarization of Sample 3: 784-785 138

10.16Peak Summarization of Sample 3: 1040-1041 139

10.17Results on Noyce data set 139

List of Tables

3.1 A list of papers presenting novel -omics alignment algorithms 31

4.1 Biomarker differences due to data processing 37

7.1 Comparison of real and simulated data 85

9.1 centWave parameter values 107

10.1 SSE Error assignment 128

Chapter 1

Introduction

Mass spectrometry (MS) refers to the analysis of the molecular composition of samples through the mass measurement of ions and their fragments. It is a key technology for innovations in diverse fields including drug design, medical diagnostics, and forensics. MS is a complex domain with many challenging and open problems (see Chapter 2, published in *BMC Bioinformatics*).

Mass spectrometers produce a complex 3-d output typically consisting hundreds of thousands of signal patterns called an isotopic envelopes (see Figure 1.1)—each caused by the accumulation of a class of molecule at a given charge state. The central task in mass spectrometry is the quantification and identification of an analyzed sample. Identification refers to the elucidation of the list of molecules that produced the signal in the MS output for a given sample. Quantification refers to the measurement of the amount of each of these molecules in a sample.

The industry-preferred method for MS identification is MS/MS fragmentation. In this technique, certain portions of sample in the mass spectrometer are broken into smaller molecules, creating an additional detected signal pattern. The idea is that this secondary signal, called an MS/MS spectrum, provides a fingerprint that can be matched to a theoretical database to suggest possibilities for the identity of the source of the original signal (called MS1) generated by the portion of the sample that was selected for fragmentation. Despite the ubiquity of MS/MS, it can only be used on a small percentage of the molecules in a

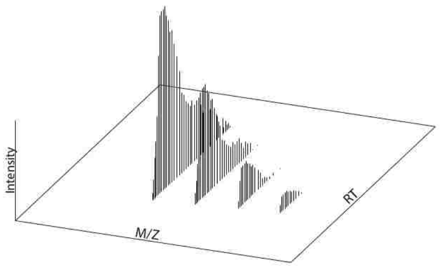


Figure 1.1: An isotopic envelope, the accumulated signal generated by a class of molecule at a given charge state in the sample analyzed via mass spectrometer.

complex sample (approx. 15%), and the database assignment is subject to high false positives (approx. 50%) [64].

An alternative to MS/MS-based identification and quantification is to match the unfragmented mass spectrometer signal output directly to a database of MS1 signals generated via *in silico* calculation of signals that would be produced by real molecules. This approach, though similar in spirit to the MS/MS approach, is free from the capture limitations of MS/MS. What's more, MS1 data could provide more identifications of higher likelihood than MS/MS alone.

Robust, MS/MS-independent isotopic envelope extraction methods are not widely used in the community. The first step towards this end is to create an accurate algorithm for extracting the component signals of an isotopic envelope: isotope traces. Though several algorithms for isotope trace extraction exist, they are unable to capture the majority of MS output signals. Moreover, this characteristic remains widely unknown due to a general lack of algorithm evaluations (see Chapter 3, published in *Bioinformatics*). The lack of evaluations, in our experience, stems from: 1) a misunderstanding of data processing's impact on the final experimental MS result, 2) shallow literature searches and comprehension of algorithm description due to ambiguous terms describing MS data concepts, and 3) the lack of labeled data.

In Chapter 4 (published in *Briefings in Bioinformatics*), we use a case study to unequivocally show that data processing algorithms are a variable in the overall MS experiment with significant impact. We make a strong case for treating each module in the data processing pipeline as an independent problem which deserves its own literature search and evaluation of a novel solution in the context of existing algorithms.

In Chapter 5 (submitted to *Proteomics*), we describe the problem of a lack of a unique, unambiguous nomenclature for MS data concepts and propose a standard nomenclature. Without a standard nomenclature, it is virtually impossible to concisely and precisely describe novel algorithms for MS data processing, a condition that is partially responsible for the bloat of publications in the field that do not necessarily improve upon the current state of the art.

Ground truth data in this field is virtually non-existant. Hand labeling is a challenge as a single data file can require a year of full-time work to label, and still may be inconsistent. This lack of ground truth data precludes quantitative evaluation of MS data processing algorithms on all but the smallest scale. What's more, with ground truth it is difficult to create advanced models for data behavior that can inform algorithm creation. A partial solution to both problems lies in creating a mathematical characterization of the data. Although such a characterization is not as informative as a full generative model due to the lack of evidence-based parameters, it is a necessary first step in creating advanced algorithms. A mathematical formalization informs new algorithms that capture known data characteristics and makes possible a simulator construction which can provide labeled data. Quantitative feedback creates new information to refine both the algorithms and the simulator, yielding an iteratively improving process.

In Chapter 6 (submitted to *IEEE/ACM Transactions on Computational Biology and Bioinformatics*), we provide a novel mathematical characterization of the behavior of chromatographic MS data. This characterization informs the Mspire-simulator, the first ever simulator to include a realistic characterization of isotope trace variance (see Chapter 7, published in *Journal of Proteomics Research*). In Chapter 8 (submitted to *Bioinformatics*), we introduce JAMSS, an MS simulator that incorporates the novel aspects of Mspire-simulator while providing a GUI, multi-threaded logic, dataset cloning capabilities, and a modular framework more fitting for future model refinements. In Chapter 9 (published in *Bioinformatics*), we use the mathematical characterization to inform a novel isotope trace extraction algorithm that greatly outperforms existing isotope trace extraction algorithms. However, not all MS studies are chromatographic. In Chapter 10 (published in *Bioinformatics*), we propose a novel solution for isotope trace extraction in non-chromatographic data using a statistical approach suggested by the mathematical data characterization.

Chapter 2

Proteomics, lipidomics, metabolomics: A Mass Spectrometry Tutorial From a Computer Scientist's Point of View[1]

Abstract

For decades, mass spectrometry data has been analyzed to investigate a wide array of research interests, including disease diagnostics, biological and chemical theory, genomics, and drug development. Progress towards solving any of these disparate problems depends upon overcoming the common challenge of interpreting the large data sets generated. Despite interim successes, many data interpretation problems in mass spectrometry are still challenging. Further, though these challenges are inherently interdisciplinary in nature, the significant domain-specific knowledge gap between disciplines makes interdisciplinary contributions difficult. This paper provides an introduction to the burgeoning field of computational mass spectrometry. We illustrate key concepts, vocabulary, and open problems in MS-omics, as well as provide invaluable resources such as open data sets and key search terms and references. This paper will facilitate contributions from mathematicians, computer scientists, and statisticians to MS-omics that will fundamentally improve results over existing approaches and inform novel algorithmic solutions to open problems.

[1] Smith, R., Mathis, Andrew D., Ventura, D., and Prince, J.T.: **Proteomics, lipidomics, metabolomics: A mass spectrometry tutorial from a computer scientist's point of view**, *BMC Bioinformatics*, 2014 (in press)

2.1 Background

Robust data processing tools for MS data are lagging behind the substantial advances occurring in instrumentation and protocol [14]. One reason for this is that few outside experts—mathematicians, computer scientists, and statisticians—have climbed the learning curve (usually requiring several years of dedicated study) to understand the terminology, chemical theory, workflows, and challenges of MS-omics (proteomics, lipidomics, and metabolomics). This sort of interdisciplinary learning curve is not unusual in bioinformatics; however, the influx of external experts to genomics has not been seen to date in MS-omics. One reason for this is the lack of a succinct and cogent introductory resource that can bring outside experts to a basic but functional level of MS-omics familiarity.

In this primer, we will elucidate the mechanisms of MS-omics, the problems it is used to solve, key concepts and terms found in the literature, and open problems and their salient literature. The purpose of this tutorial is to expedite the new researcher's acquisition of a functional knowledge of MS-omics sufficient for contribution to the field.

2.2 Results and discussion

2.2.1 Relationship of genomics, proteomics, lipidomics, and metabolomics

The exponential growth of genomics studies during the last ten years has not been matched by corresponding research in MS-omics [118]. Genomics researchers have several peer-reviewed conferences in which to publish their results. To the best of our knowledge, there has not been a single peer-reviewed conference to date on lipidomics or metabolomics, let alone any specifically addressing algorithmic approaches to problems specific to either area, although there are periodic special genomics conferences dedicated to proteomics. Several existing venues labeled as bioinformatics will not accept papers on MS-omics, as their stated area of interest is limited to a distinct subfield of bioinformatics such as genomics. This phenomenon of focus on genomics is also reflected in institutional research programs. In a recent review of

78 post-secondary degree-granting bioinformatics programs, 22 programs noted a research emphasis in genomics, while 18 noted a research emphasis in proteomics. Not a single institution listed a research program in lipidomics or metabolomics [46].

The biological reach and impact of research in MS-omics is so extensive that it can be argued that MS-omics should now be the highest priority of systems biology [41]. From a pragmatic perspective, the large set of fresh problems and substantial potential for impact in MS-omics ought to be very attractive to those in more crowded disciplines.

Proteomics

Proteomics is the study of biological processes via the analysis of protein expression or state in cells or tissue. Proteins are ubiquitous building blocks of life, and they are composed of peptides, which are chains of amino acids built by translating mRNA. There are 20 amino acids, uniquely abbreviated with a single letter. Peptides thus can be described as a string of the letters corresponding to the amino acids. Though protein sequences are determined by DNA sequences, post translational protein modifications (such as acetates, phosphates, lipids etc.) are not as easily predicted. These modifications quickly diversify and regulate/complicate protein function and cellular protein composition and are characteristic in most cellular processes and diseases. Therefore, the aim of MS-proteomics is to provide data that DNA sequences cannot—namely, individual protein concentrations and identification of post-translational modifications.

Lipidomics

Lipidomics is the systems-level analysis of lipids (fat molecules) and their interactions [34]. It is a science still in its infancy but one that promises to revolutionize biochemistry [41]. Lipids are grouped into eight categories that share common physical and chemical properties [33, 41], and there are currently some 38,000 documented lipids.

Lipids that occur rarely or in small quantities are often the most effectual lipids in biological processes, meaning they are particularly important in disease diagnostics and in understanding pathology [34]. Lipidomics can elucidate the pathology and treatment of many diseases such as cancer, diabetes, obesity, cardiovascular disease, arthritis, asthma, inflammatory bowel disease, Alzheimer's and others due to the associated disruption of lipid metabolic enzymes and pathways [34, 53, 66, 118]. A better understanding of lipidomics could significantly advance diagnostic medicine as well as provide novel treatment options.

Metabolomics

Metabolomics is the study of metabolomes—small molecular end products of cellular regulatory pathways [35] that can provide a snapshot of cell physiology. Metabolites are much smaller than proteins and smaller than most lipids. Their small size precludes the direct overlap of some techniques used in proteomics or lipidomics, but they may be generally analyzed in similar ways. Lipids may be classified as a subset of metabolites; however, mass spectrometrists typically consider lipids distinct from metabolites because analytically they must be treated separately (i.e., require different solvents).

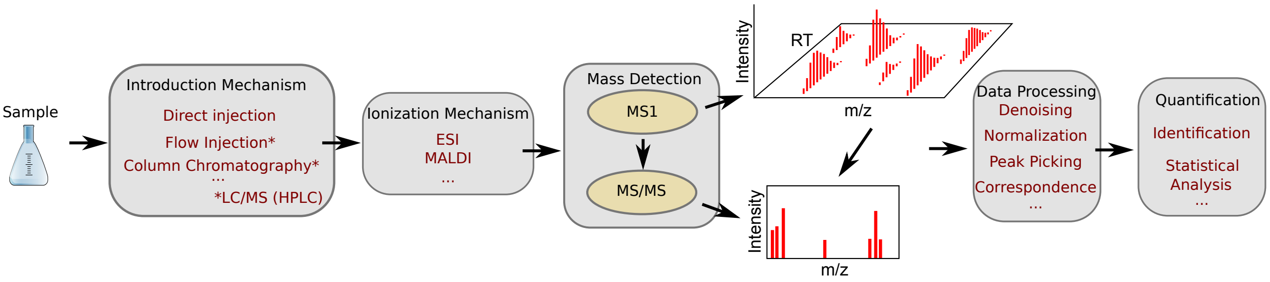


Figure 2.1: The MS-omics pipeline. A sample is introduced to an ionization mechanism with or without a preliminary separation technique, where particles receive a charge enabling the detector to estimate the mass-to-charge ratio (m/z) and intensity of each analyte. If the system has tandem mass spec capabilities, some precursor ions (MS1) are selected for fragmentation (MS/MS). Data processing techniques prepare the data to be quantified via statistical methods and identified via matches to theoretical databases.

2.2.2 MS-omics pipeline

The workflow from sample preparation to result quantification, can be split into two consecutive pipelines: the wet-lab pipeline and the data processing pipeline. The data processing pipeline consists of many possible processing steps that take the data resulting from the wet-lab pipeline (the mass spectrometer output) to the end result: identification and quantification (see Figure 2.1). The quality of each step in the pipeline affects the sensitivity and reliability of the outcome [84]. There are many optional steps, some of them very popular. We will describe the essential and some optional steps.

All MS experimental data share a set of descriptive keywords that are essential for referencing components of the output map (see Figure 2.2). A unique and unambiguous visual lexicon for MS-omics data processing data structures is given in [95]. A comprehensive reference of key MS terms is provided in [69].

Sample preparation

The details of sample preparation are beyond the scope of this paper. However, at a general level, sample preparation strategies prior to mass spectral analysis are based on isolating analytes of interest and removing all other contaminating molecules. For instance, filters can be used to separate high molecular weight proteins from low molecular weight lipids and metabolites, or contaminates. Other sample preparation techniques exploit analyte hydrophobicity, charge, and analyte-specific affinity. The degree of specificity in sample preparation is determined by the end goal of the experiment [26]. For example, if an experiment requires the analysis of only phosphorylated proteins, the sample preparation should isolate only phosphorylated proteins. Of course, this is very challenging but using an appropriate sample preparation strategy specific to an experimental need significantly simplifies mass detection and data analysis and in some cases is required to identify analytes of interest. Proteomics, lipidomics, and metabolomics each have unique considerations in sample preparation.

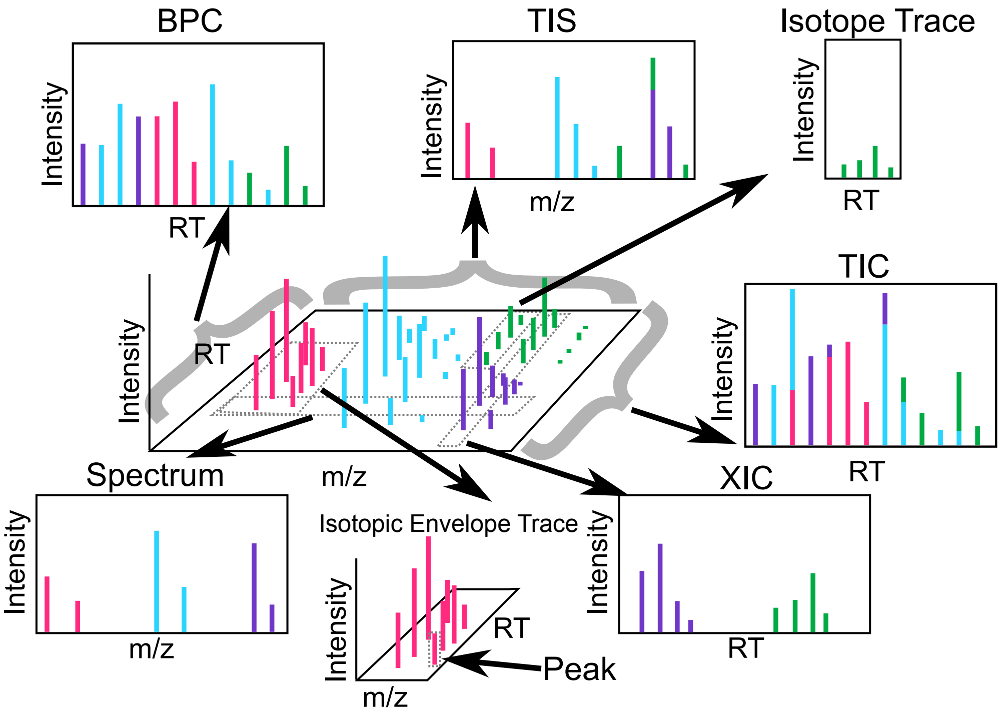


Figure 2.2: Common nomenclature. Each portion or summary of an MS run is referred to by a different name. A *spectrum* contains all points with a single RT value. The sum of signals across all spectra is called the *total ion spectrum (TIS)*. A slice of data containing a contiguous m/z range extending across all RT is called an *extracted ion chromatogram (XIC)*. While the *total ion chromatogram (TIC)* is the sum of all signals across all m/z, the *base peak chromatogram (BPC)* is the set containing the most intense signal for each RT across all m/z. An *isotope trace* is the signal produced by a single ion of a single analyte (i.e., a peptide or a lipid) at a particular charge state. An *isotopic envelope trace* is the group of isotopic traces produced by a single analyte at a particular charge state. Note that certain terms like peak, feature, and chromatogram, are overloaded in the literature and as such are exceedingly unclear [95].

Introduction methods

Direct injection refers to infusing the sample directly into the mass detector. This is usually done with some sort of machine to make the flow constant.

While it is sometimes advantageous to allow all analytes to flow through detection at once, most MS experiments of complex samples will use chromatography due to its ability to spread out analytes over time, making it less likely that the ionization capacity will be overcome by large quantities of analyte or background ions, a phenomenon called ion suppression.

Chromatography disperses the introduction of analytes into the mass detector through time based on some chemico-physico property (hydrophobicity, for instance). All chromatography systems have two phases: the stationary phase and the mobile phase. The stationary phase causes analyte separation and the mobile phase carries the analytes through the chromatographic column to the mass spectrometer. Methods include:

- LC-MS - mass spectrometry coupled to liquid chromatography. Liquid chromatography uses a liquid mobile phase and a column packed with chemically derivated beads as a stationary phase. The mobile phase is composed of a two-liquid gradient. Changes in the gradient (the percent composition of each liquid) cause analytes to be slowly released from the column and enter the mass spectrometer. Different stationary phases can separate analytes based on hydrophobicity, charge, size, or affinity. However, the most common stationary phases for LC-MS on biomolecules are reversed phase (hydrophobic) and strong cation (charge) [23].
- GC-MS - mass spectrometry coupled to gas chromatography. In gas chromatography systems the mobile phase is an inert gas (such as helium) and the stationary phase is a column designed to separate molecules based on polarity. The gradient is temperature increase; molecules with a high affinity for the column elute at higher temperatures.
- CE-MS - mass spectrometry coupled to capillary electrophoresis. Electrophoresis differs from chromatography, relying on electric fields, rather than mobile and stationary phases, to separate molecules [65]. Capillary electrophoresis uses an electric field applied to long narrow capillaries to separate molecules based on size, charge, and flow resistance through the capillary.

Multidimensional chromatography (sometimes referred to as tandem chromatography) refers to two chromatographic systems applied to the same system. In the case of LC-GC-MS, for example, analytes are introduced into the gas chromatography system as they elute from the LC system, with each system causing analytes with specific properties to elute with precedence. A more common multidimensional system in MS-omics is MUDPIT. MUDPIT uses two orthogonal separation strategies like strong cation ion exchange (charge based) and reversed phase (hydrophobicity based) chromatography to achieve greater resolution.

Ionization methods

Analytes must be ionized (i.e., in a charged state) in order to be detected by the mass spectrometer. Electrospray ionization (ESI) was developed in 1994 and is the most popular in MS-omics due largely to its ability to ionize unstable molecules without breaking chemical bonds and to the diverse range of analytes that can be ionized by the method [20, 44]. Other methods include atmospheric pressure chemical ionization (APCI) [43], matrix-assisted laser/desorption ionization (MALDI) [43], and electron-ionization (EI) [43]. Ionization methods for ms-omics are generally referred to as soft ionization methods and include ESI and MALDI. EI is a harsh ionization method and will destroy most biomolecules except for very stable lipids and metabolites.

Mass detection

As charged particles are passed through the mass spectrometer, the mass-to-charge ratio (m/z) of detected particles is registered. A single scan on the resulting output represents a snapshot of the precursor ions passing through the mass spectrometer at that particular retention time (RT). The ions in this stage are called precursor ions because in tandem mass spectrometry (MS/MS), ions in small m/z windows are captured for fragmentation and MS detection a second time, yielding a second set of ions called product ions that can be used to identify precursor ions by matching their MS/MS patterns to a database of possibilities. It is important to understand that the ratio of solution selected for MS/MS fragmentation is low, normally capturing only 10-20% of the precursor (MS1) data. Because most MS/MS systems autoselect what segments to capture based on intensity, much of that portion overlaps between replicates. Of that 10-20%, less than 60% are identified via database lookup, and even that is subject to false positive identifications [64].

An analyte can contain certain naturally occurring rare isotopes, such as carbon-13. These isotopes tend to occur in individual analytes in known quantities, causing a characteristic pattern called an isotopic envelope (see Figure 2.2). The envelope is characterized by the number of and relative intensity between its isotopes. The monoisotopic peak, or peak that appears at the theoretical mass discounting any attached heavy isotopes, usually appears alongside the slightly heavier masses of any portion of the peptide or lipid in the sample that contains heavy isotopes.

When an analyte exists in a run in more than one charge state (a very common occurrence due to variability in ionization), its isotopic envelope will reappear in a compressed and shifted form due to increased charge, as illustrated in Figure 2.3. The equation for the shift is specific to the source of the charge. For instance, a charge can be induced by the addition of a proton, in which case the shift is defined by $(\mu + k)$/charge m/z with a gap between ions in the isotopic envelope of $1/k$, where k is the charge of the analyte (3+, 2+, 1+, and 1+, respectively in Figure 2.3) and μ is the m/z of the single-charged analyte (this is the analyte with only a +1 charge—399 in Figure 2.3).

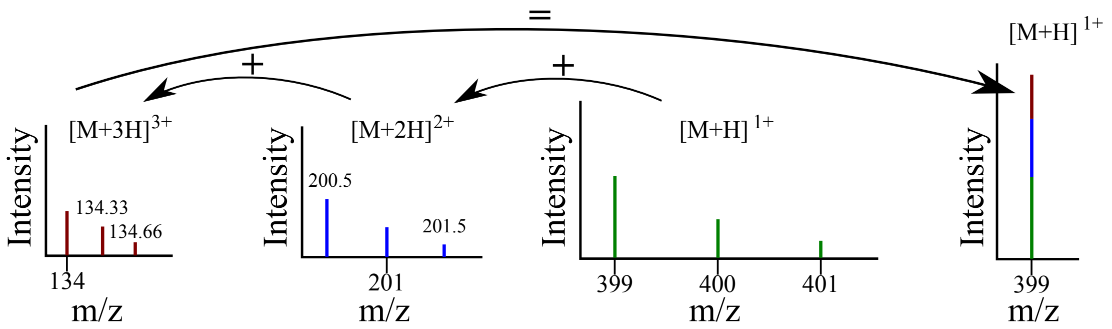


Figure 2.3: Deisotoping. A contrived example of deistoping. The same molecule is displayed here in three reduced isotopic envelopes (denoted by color) created from single- ($[M + H]^{+1}$), double- ($[M + H]^{+2}$) and triple-charged ($[M + H]^{+3}$) instances of the molecule. The monoisotope (the lowest m/z ion) from each isotopic envelope is combined to form the deistoped monoisotopic peak.

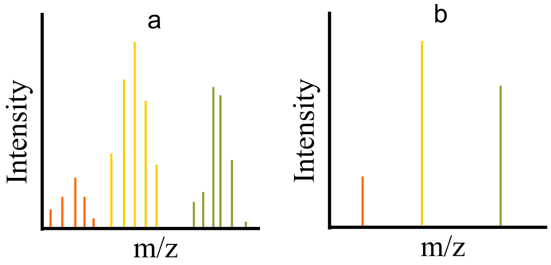


Figure 2.4: A profile (a) and centroid (b) version of the same spectrum. The profile raw data detected by a mass spectrometer consists of distributed signal across m/z values at each point where an ion is detected. Centroid data is raw data that has been processed by an algorithm to retain only the local maximum in each range in which an ion is detected. Because each ion detected creates an m/z distribution of signal, the distribution itself (in profile mode) or the maximum to which it is reduced (its centroid) is sometimes called a peak. This ion intensity distribution along m/z is not to be confused with the distribution of ion intensity along time in chromatographic studies (see Figure 2.2).

Mass spectrometers output raw data—a large collection of data points each consisting of a tuple of m/z, intensity, and time (RT) either in profile or centroid form. Profile data contains all data points registered by the mass spectrometer (see Figure 2.4a), while centroid data has been reduced to data points that represent the local maxima in a single spectrum, a distribution of data over an m/z range for a given RT (see Figure 2.4b). Centroid data is much more concise than profile data, but the reduction incurs information loss.

Experiments can run in full scan mode—where the full range of m/z values is read—or the mass spectrometer can scan only certain m/z values (called single reaction monitoring in the case of one m/z value or multiple reaction mode in the case of several) [43].

Mass spectrometers have varying characteristics depending on the mechanisms used for mass detection, each with a different resolution. Resolution at a certain m/z is given by the ratio of that m/z to the smallest m/z gap between two distinguishable ions. Higher resolution instruments yield narrower profile peaks (see Figure 2.4a), allowing the signals from two distinct ions to be distinguished despite their similarity in m/z.

Data processing

Data processing consists of each of the possible steps in the MS-omics pipeline (Figure 2.1) involving digital manipulation of the mass spectrometer data or products from that data. These methods are constantly being improved upon and are discussed in detail in Section 2.2.5. Here, we provide a high-level overview of the role of data processing in the MS-omics pipeline.

The first step in data processing is handling the raw data produced by the mass spectrometer. Algorithms for noise reduction, feature detection, and correspondence exist that operate on the raw data. However, many require preliminary conversion out of the proprietary data format of the instrument and into an open data type (see below for a discussion of existing data types). It is important to note that, due to the size of the data sets, random access data processing—where only a portion of the data file is loaded into memory at a time—is a must, although some current tools load the full file and are therefore prone to crashing and subject to file size limits as memory is exhausted.

Prior to analyte identification, the data must be denoised, peak-picked, feature-detected, deisotoped, and deconvoluted. These are significant and open problems and are discussed in more detail below.

Analyte identification follows data processing. Here, one of several available databases are used to compare the experimental feature observations (i.e. isotopic envelopes, isotopic traces, etc.) to theoretical patterns. These include Sequest [32] for proteins, LIPIDMAPS [86] for lipids, and METLIN [91] for metabolites. Due to incomplete/growing databases and noisy data, closest-match assignment is prone to false positives and mismatches. Statistical analysis is almost always incorporated in this or prior steps in order to ascertain the significance of the identification.

The ultimate goal of data processing is to yield the quantity of each analyte. The identification and quantity of analytes, as well as the underlying raw data, must be stored in data structures that allow for efficient access and manipulation of the data.

2.2.3 Data types

Raw data is a general label that actually describes a set of data formats specific to the vendor of the instrument. Many data converters from raw to open data formats exist. One popular converter is pwiz (`http://proteowizard.sourceforge.net/`). The Network Common Data Form (NetCDF), a generic open science data format, is an early data format that is still in use in some applications. mzXML is an open XML based data format with wide support. mzML was developed to replace mzXML and has more information from the raw data encoded and uses extensible ontologies to encode meta-data. mzQuantML is an open data format specifically intended for the storage of quantities associated with identified feature data. mzIdentML and pepXML are standards designed to facilitate database identity searches. Annotated Putative Peptide Markup Language (APML) is an XML standard designed to provide a single data file encoding of the original data set and its modifications via data processing tools [13].

2.2.4 Data sets

Lack of labeled data

The prevailing problem in developing and evaluating computational approaches to MS-omics problems is the lack of labeled data [74]. Labeled data is difficult to obtain both because of the size of data sets—which can easily consist of millions of data points per file and hundreds of GBs of files for a replicate experiment series—and the undependability of hand-labeling—which is both time consuming and subjective. Several approaches for mitigating this problem exist: qualitative metrics, spiked mixtures, and *in silico* simulated data.

Qualitative metrics Evaluation metrics that do not use ground truth avoid the need for labeled data. For example, replicate alignment quality can be assessed via the Pearson correlation coefficient, feature overlap rate, or coefficient of variation. This approach is sub-optimal, as a good score on a qualitative metric does not necessarily translate into a good quantitative score using labeled data, but it is easy to compute and is comparable across problem instances.

Spiked mixtures Commercially available purified and quantified measures of a specific analyte are combined to produce a data set with known composition and quantity. These samples are not exactly ground truth, however. Due to ionization inefficiencies, environmental contaminants, and the variability of mass spectrometry, no instrument will report the same quantity and composition predicted by a spiked mixture. What's more, a mixture of a few analytes, which often do not co-occur in nature, is hardly representative of real-world scenarios, in which complex samples can easily contain hundreds of thousands of distinct analytes. To create more realistic conditions, spiked mixtures can be added to samples where the spiked analytes are not expected to occur. However, a method's accuracy on a few analytes is not necessarily indicative of performance across all analytes, particularly given the variability and limitations of MS/MS, which is commonly used to single out the m/z of the expected analytes but cannot be expected to capture the gross majority ($\approx 80 - 90\%$) of the remaining sample.

***In silico* simulated data** Simulated data is used in the field to refer to real-world data sets that have been purtubed with m/z shifts or intensity value modifications in order to create psuedo-new data without having to rerun costly experiments. True simulated data, called *in silico* to identify that it as purely sourced from simulation algorithms on a computer, is a relatively new advent in MS-omics. Creating realistic *in silico* data requires the analysis of many ground truth datasets, which creates a chicken and egg problem, as the difficulty of obtaining ground truth datasets is the very reason an *in silico* simulator would be beneficial.

Sources of open data

To facilitate strictly algorithmic advances in MS-omics, to avoid the need for a costly wet lab for creating mass spectrometry data, and to aid in evaluative comparisons against existing methods, more and more practitioners are making their data freely available online. Although any serious foray into MS-omics should certainly include a collaborator with mass spectrometry assets and formal training, we present a list of some of these open data sets in order to aid those who are interested in investigating MS-omics for the first time as well as more seasoned investigators who would simply like to make a case for the generality of their methods.

Lange *et al.* have provided two proteomic and two metabolomic data sets [55] which they have used to assess the quality of several alignment algorithms at `http://msbi.ipb-halle.de/msbi/caap`. The data is already segmented into reduced isotopic envelopes (isotopic envelopes whose isotopic traces are integrated into a single point).

Listgarten *et al.* provide centroided replicate data with spiked-in peptides [58]. There are two data sets: a set of 11 replicate LC-MS runs from ruptured *E. Coli* cells and a set of 14 LC-MS runs of human serum samples.

Jeffries provides a data set consisting of raw replicates of SELDI data [49] at `http://data.ninds.nih.gov/Jeffries/alignment/index.html`.

The SuperHirn data set [67] can be found at `http://proteomics.ethz.ch/muellelu/web/Latin_Square_Data.php`. It consists of 18 LC-MS runs from tryptic digests of 6 nonhuman proteins spiked with different concentrations into a complex human peptide sample and includes the raw as well as processed data. The data was obtained on an FT-LTQ.

2.2.5 Problems of interest

Among the data processing portion of the MS-omics pipeline, some problems are widely studied, and some are emerging. All provide future research potential.

In silico simulation

The lack of ground truth data for evaluation of data processing algorithms precludes effective validation and comparison. *In silico* data simulation is a relatively new approach to providing on demand ground truth simulated data. By modeling a list of analytes and a description of experimental conditions, simulators can provide estimates of mass spectrometer output combined with labels of the analytes and quantities used *in silico* to generate the data (see [9, 70, 87, 94]).

Correcting mass shift

Analyte detection on the m/z axis in mass spectrometers is subject to two types of error: systematic mass error—a functional deviation from true mass—and random mass error [28]. Typically, systematic mass error is mitigated by routine machine recalibration—a process wherein analytes of known mass are processed in the mass spectrometer to create a model that is used to interpolate m/z shift for any given m/z value. However, the efficacy of this calibration reduces over time as the mass constantly continues to shift. Additionally, some machines benefit from an injection of spiked standards during a normal experiment for internal calibration, which helps overcome the temporal effects of space charge effects, electric fields, peak intensity, and temperature [28]. Internal standards are undesirable due to the additional cost of standards and the suppression implications of spiked standards. Computational mass calibration techniques have been proposed in order to provide the mass accuracy of internal calibration but with better consistency and lower cost [28]. This is an active but not crowded area of research with practical implications.

Correspondence

Correspondence, the registration of recurring signals from the same analyte over replicate samples, is a crucial problem in any of the many MS experiments where multiple runs of similar samples are compared to each other (see Figure 2.5). For a comprehensive review of current algorithms, see [97]. Persisting problems are an abundance of user parameters, models that do not include known behavior, prohibitively long runtimes, and a lack of performance comparison between methods [98].

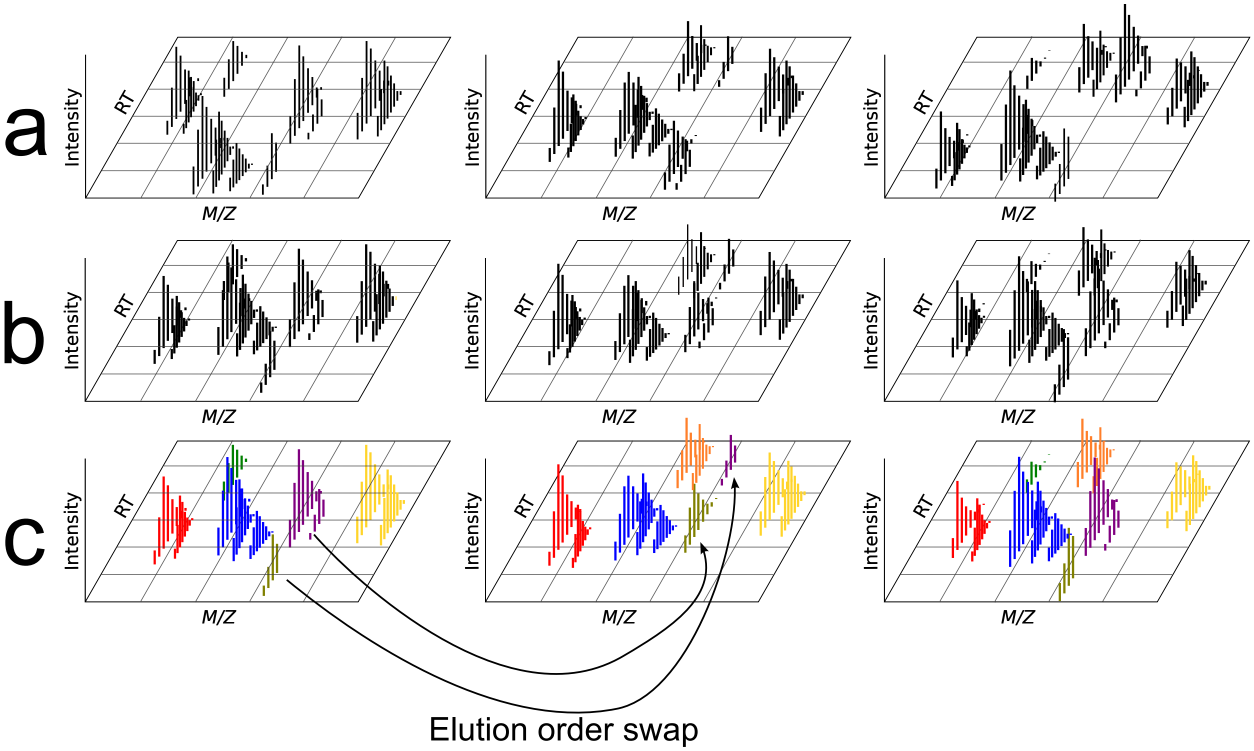


Figure 2.5: MS correspondence. Correspondence is the problem of registering features across multiple samples (matches across the samples are depicted in the same color). Most times this process is facilitated by aligning the retention time (RT) of features across multiple samples (top to bottom row). Note that features are almost never present across all samples and can display significant RT variability and (to a lesser degree) m/z variability.

Denoising

MS-omics produces inherently noisy data. Noise can consist of spurious data points or distortion of a data point's true value in retention time, m/z, or intensity. Denoising as used in MS-omics refers to the removal of spurious data points. Baseline subtraction is a common method in which signals with intensity lower than an adaptive threshold are considered to be noise and removed (see Figure 2.6). This is an active area of research, though most experiments in the literature have not made an explicit and dedicated study of different techniques, instead describing the denoising method applied as a data processing step in a larger experiment.

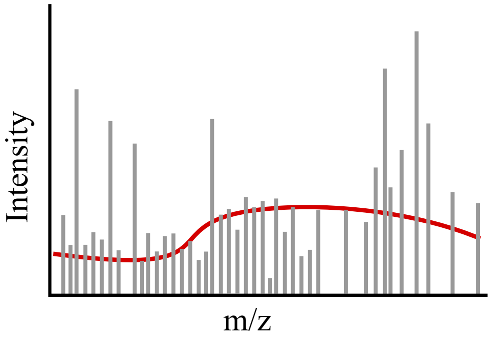


Figure 2.6: Baseline subtraction. Baseline subtraction is the functional estimation and removal of background noise.

Feature detection

The most important step of an MS-omics workflow is undoubtedly feature detection [14], a general term that can apply to the extraction of various signal elements from MS data. In chromatographic data, feature detection can refer to either extracting isotopic envelopes or isotopic traces from an MS sample output (see Figure 2.7). Many methods exist for isotope trace extraction, among them a promising new algorithm that performs well on existing evaluations [21]. Sometimes this process is called peak picking or peak detection, but those terms should be avoided since they are also used to refer to the conversion from profile data to centroid data. In direct injection data, feature detection is sometimes referred to as peak summarization, since each spectra (being an approximation of the latent content of the non-chromatographically separated sample) must be combined into a TIS through mitigating the variance inherent in m/z across spectra (see [92]).

Data structures

As described earlier, many data types exist for MS-omics data. New data formats continue to be proposed to meet unforeseen needs.

A recent prevailing expansion point has been the need to store the results of data processing tools in addition to the original data. Truly modular pipelines require data structures that contain all necessary data to be used by any tool in the pipeline, meaning

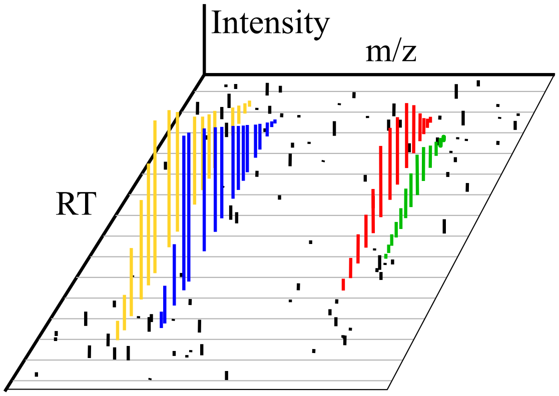


Figure 2.7: Feature detection. Feature detection consists of labeling data points which pertain to individual features (indicated by color here) while excluding noise points (in black).

previous modifications are annotated in addition to retention of the original data. APML is one attempted solution to this problem, but, so far, the community has not embraced it, as it appears that there are only two extant algorithms which use it [13].

There is still a need for compact, random access, and information rich data structures and access for MS data [102]. What's more, some proprietary formats can still only be converted to open formats on Windows platforms.

Identification

As discussed earlier, mass spectral identifications may be based on several factors, but two inputs, the precursor mass (the mass of the molecule) and the fragmentation pattern (through MS/MS) of the precursor mass, are by far the most common identifiers. This spectral information provides a fingerprint unique to most biological molecules; however, low quality spectra cause false positives and false negatives. While improving mass spectrometry will certainly improve spectral quality, improving spectral search algorithms and employing new identification inputs will allow for more confident identifications. This is particularly true for the relatively new fields of metabolomics and lipidomics.

Predicting RT

Retention time refers to the amount of time an analyte is delayed by chromatography before exiting and being detected by the mass spectrometer. Retention time is correlated with physical and chemical analyte characteristics; therefore, predicting analyte retention time provides another factor for positive identification. Many peptide retention time prediction strategies exist [5]. However, cross instrument retention times vary greatly due to changes in experimental parameters, creating a real need for retention time normalization as well as retention time prediction.

Mass variance correction

Mass variance, the difference between the theoretical and experimental (observed) mass of analytes is an open problem. One way of correcting mass variance is by using the weights of the elements of each analyte to predict m/z locations where a lack of signal is impossible, allowing for the identification of systematic deviation from theoretical masses in a sample [125]. A similar approach is to model such theoretical gaps via a sine curve fitted via a fast Fourier transform [28]. Accurate m/z values are essential to analyte identification.

Ontology

According to a recent survey of the field, the biggest problem in lipidomics is the need for a standardization of data acquisition and data processing, due to the huge variability in instruments, protocol and data processing for lipidomics[51]. The many options and permutations in the MS pipeline would make for a very long methods section if explicitly described in a paper—much too long for any journal's page limits. Although several partial ontologies exist (see [105, 119]), there is no concise way to uniquely identify an experiment from start to finish, including sample preparation, mass spectrometry protocol, and post-processing. Existing ontologies are particularly lacking in terms of data processing terms.

Absolute quantitation

MS signal intensity is related to but not equivalent to analyte quantity [57, 59]. Factors that influence this discrepancy include [36]:

- *Ionization efficiency.* Not all analytes in a sample are ionized.
- *Enzyme digestion rate.* When an enzyme—such as trypsin—is used to digest proteins into peptides, not all proteins are completely cleaved. This leads to less-than-expected signal abundance, as the true abundance will be diminished by whole proteins (which are not ionized and therefore not detected), and incompletely digested proteins (which will be detected at different m/z than the expected peptide components).
- *Ion suppression.* When the quantity of analyte entering the ionization mechanism at a given time exceeds the ionization capacity of the ionization mechanism, only a portion of the analyte is charged [3].

Accurate models of these effects would improve estimates of analyte population in samples, as well as further advance *in silico* simulation.

Currently, quantification methods generally fall into one of three approaches: label free spectral counting, quantification via differential stable isotopes, and label free quantification based on the precursor ion signal intensities [68]. Spectral counting is a method in which peptide signals are used to create a protein tally—the count of every protein containing a certain peptide is incremented each time one of its peptides is identified via MS/MS. Despite its prevalence, the accuracy of spectral counting is limited by its dependence on MS/MS acquisition rates, which, as mentioned above, are very low, and its propensity for false positives, since all proteins containing each detected peptide are considered as present when in reality only one need be. Stable isotope labeling methods (SILAC, ICAT, iTRAQ, and TMT) also have significant limitations (see [126]). Besides cost and sample prep complications, nearly all methods increase the number of co-eluting analytes, creating a bottleneck for the complexity of samples handled. What's more, because stable isotope methods target a

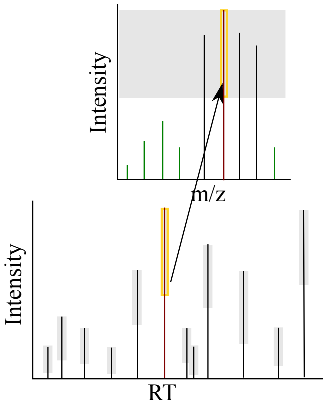


Figure 2.8: Dynamic range. Dynamic range is the window of intensities visible to the sensor at any given RT. The main chromatogram shows the signal of maximum intensity for each RT. The gray box indicates the dynamic range at that RT. The red peak is shown with the other signals at that RT. Note the green peaks will not be detected by the mass spectrometer because they lie outside the dynamic range.

small specific list of analytes *a priori*, they are not practical in terms of time and money for data-driven discovery, where sample composition is unknown [111].

Modeling dynamic range suppression effect

Dynamic range is a term that describes the minimum intensity of a detectable signal given a co-eluting analyte of a higher intensity (see Figure 2.8). All mass spectrometers have a dynamic range limitation. The current state of the art is 10^3 - 10^4, meaning that at a given RT if one analyte has an intensity of 1.3×10^5, any analyte with an intensity less than 1.3×10^2 would not be detected.

Fragment ion intensities

Because MS/MS acquisition captures not just the analyte of interest but also any surrounding precursor ions, and because fragmentation isn't a perfect process, fragment ion intensities are not as accurate as desired [10, 41]. Several machine learning approaches have been proposed for making more accurate fragment identifications [4, 31]. However, this is still an open problem.

De novo peptide sequencing

De novo sequencing is an alternative method to database matching that accommodates peptides that don't match up with the database (caused by mutations, polymorphisms, modified amino acids or simply a missing database entry) [37]. Here, the original peptide sequence—defined by a series of letters, each representing an amino acid—is reconstructed based on the MS/MS fingerprint and the chemical properties of the analytes. A recent tutorial provides more detail and resources [17].

Fragmentation patterns for lipids

Proteins have a known cleavage pattern, meaning that when peptides are fragmented by MS/MS, association to a peptide is straightforward. Lipids, on the other hand, have a much more complex form due to a wider vocabulary of building blocks and a more complicated fragmentation pattern. To date, no fragmentation rules have been published, making MS/MS much less helpful in lipidomics than proteomics. Because of the complexity of lipids, a machine learning approach could be appropriate in finding a solution to this problem.

Biomarker detection

Biomarker discovery is the use of comparative analysis (see Figure 2.9) in order to identify analytes that correlate with certain diseases or other conditions for diagnostics or drug development. It is an active area of research with a lot of published work; however the problem is still wide open due to limitations in mass spectrometry, pre-processing, and identification. Current methods struggle to highlight case/control differences in complex samples, requiring painstaking, time consuming, and error-prone manual detection.

Deisotoping

Deisotoping is the process of reducing several instances of the same analyte at different charge states into a single feature—usually a monoisotopic peak (see Figure 2.3). This is necessary

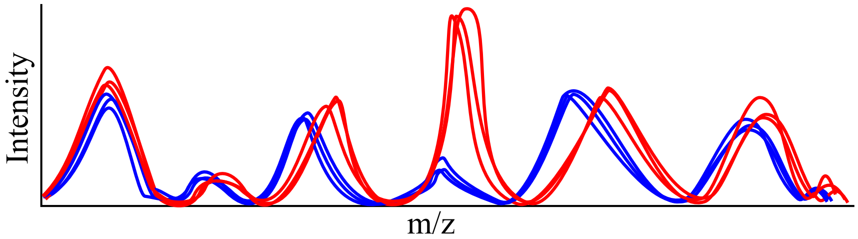


Figure 2.9: Difference detection. Comparative or differential MS-omics aims to identify possible differences between two sets of replicate studies. In this case the three red signals are cases—samples from individuals of interest—and the blue signals are controls—samples from baseline individuals. The center peak clearly indicates a differentially expressed analyte.

because the query to a data base search consists of only the single-charged feature m/z and (optionally) RT. Adding to the complexity of registering differently charged versions of the same analyte is the fact that, in complex samples, the isotopic envelopes of different analytes can and do overlap, requiring deconvolution (see below).

Deconvolution

Overlapping signals must be resolved prior to quantification (see Figure 2.10). RT overlaps occur when two isobaric analyte elute without a gap between them, and are more common in complex samples. Isotopic envelope overlaps occur in m/z where two analyte are not sufficiently separate in m/z at their current charge state. Ion overlaps occur when particular ions of two given analyte are too similar to be resolved in m/z. All m/z overlaps are less likely in high resolution machines, which by definition are capable of better resolving power evinced by more narrow signals in m/z. RT overlaps can be minimized to some extent by sample preparation and protocol designed to separate similar molecules into different RT areas.

Parameter reduction

In general, most algorithms require the user to optimize a host of parameters through manual tuning, which is time intensive. New algorithms should avoid free parameters. If included, they should also provide guidance or an automated method to fix them. Research

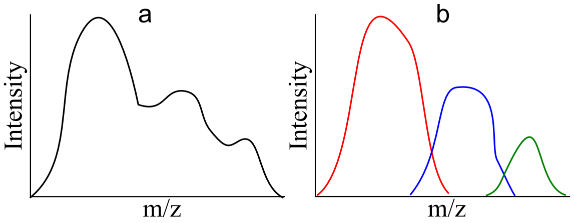


Figure 2.10: Deconvolution. Overlapping analytes create convoluted signals, which must be deconvoluted. This example depicts how three convoluted peaks in profile mode might look in the output of a low resolution mass spectrometer (a). In order to further process the data, they must be deconvoluted into their respective peaks (b).

opportunities include developing methods for automatically optimizing parameters on existing and popular methods.

2.3 Conclusions

MS-omics is an exciting, developing field with many research opportunities for mathematicians, computer scientists, and statitisticians. Although contribution to the field requires a functional understanding of many domain-specific concepts and terms, the open nature of most of the existing problems provides many opportunities for impact.

2.4 Competing interests and declarations

The authors declare that they have no competing interests. RS is supported by the NSF graduate research fellowship (DGE-0750759).

2.5 Author's contributions

RS, ADM, DV, and JTP all contributed in writing this manuscript.

Chapter 3

Novel Algorithms and the Benefits of Comparative Validation[1]

Bioinformatic research has produced a large volume of proposed algorithmic solutions to a host of problems. Whether presented as a processing step in a clinical experiment or treated in a stand-alone publication, novel bioinformatic algorithms are often not subjected to the thorough comparative evaluation endured by their counterparts in other closely related fields—such as computer science—where an algorithm unevaluated against extant methods is considered unpublishable. Two audiences are interested in algorithmic publications: the practitioner, who may use the algorithm, and the researcher, who will work to develop solutions superior to those extant. We argue that failure during the review/publication process to require comparative evaluation for novel algorithms is detrimental to both parties.

To demonstrate the dilemma, we conducted a case study of novel LC-MS alignment algorithms. Of the 48 publications from 2001 to 2012 that present alignment algorithms of which we are aware, 60% include no comparison to other methods. Another 20% compare their method to one or two others (see Figure 3.1). Only two papers compare performance against the state-of-the-art methods available at the time of publication. Interestingly, both of these, with 6 and 7 comparisons respectively, reuse comparative evaluation performance data and data sets from a stand-alone review paper of 6 methods [55].

It is natural to wonder if publication year correlates to the number of comparisons made. After all, earlier papers would have less methods to compare against. We found no correlation (r=0.397) between year of publication and number of comparisons (see Table 3.1).

[1] Smith, R., Ventura, D., and Prince, J.T.: **Novel algorithms and the benefits of comparative validation**, *Bioinformatics* 29(12), 15831585, 2013

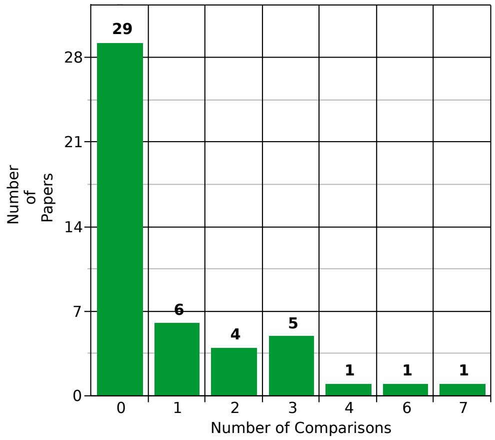


Figure 3.1: A comparison of the number of papers presenting MS alignment algorithms and the number of competing algorithms against with they compare. The majority of novel alignment method papers fail to compare against even one extant method.

Again, the correlation number would be even lower if it weren't for the fact that someone published a comparative evaluation of at least some of the extant alignment methods. Without the reuse of that survey paper data, the correlation coefficient would drop to 0.313. These data reinforce the prevailing paradigm that comparative performance of a new algorithm to existing ones is too time consuming for the author and reviewers and ought to be the subject of dedicated research [6]. At least for alignment, such dedicated comparison studies are few and far between—we are aware of only one such comparative survey paper, even though almost 50 new algorithm papers have been published over the last 11 years (see [55]). Even if these evaluative review papers were more numerous, there are many reasons why these evaluations ought to be primarily provided in the novel algorithm publications themselves.

A practitioner relies on the peer review process to ensure that the methods they are choosing have met a minimum standard of quality. Though a new method's description or performance may be convincing, these qualities alone are insufficient to weigh the usefulness of an algorithm. Without comparative evaluation, algorithms that under-perform against existing ones can easily flood a domain, making the practitioner's task of selecting an algorithm more difficult with every additional publication. Besides an extensive literature review caused by the inundation of papers on the subject, the practitioner must also perform

Table 3.1: A list of papers presenting novel -omics alignment algorithms. The data has a correlation coefficient of 0.397, suggesting there is no trend towards comparison against extant algorithms.

Publication	#Comp	Year	Venue
Fraga *et al.*	0	2001	Anal Chem
Hastings *et al.*	0	2002	Rapid Com in MS
Bylund *et al.*	1	2002	J Chrom A
Torgrip *et al.*	2	2003	J Chemometrics
Åberg *et al.*	0	2004	J Chemometrics
Lee *et al.*	0	2004	Anal Chim Acta
Tomasi *et al.*	0	2004	J Chemometrics
Eilers	0	2004	Anal Chem
Vorst *et al.*	0	2005	Metabolomics
Pierce *et al.*	0	2005	Anal Chem
Walczak *et al.*	4	2005	Chem Intel Lab Sys
Baran *et al.*	0	2006	BMC Bioinformatics
Smith *et al.*	0	2006	Anal Chem
Sadygov *et al.*	0	2006	Anal Chem
Fischer *et al.*	0	2006	Bioinformatics
Jaitly *et al.*	0	2006	Anal Chem
Prince *et al.*	1	2006	Anal Chem
Skov *et al.*	0	2007	J Chemometrics
Yao *et al.*	0	2007	J Chrom A
Kirchner *et al.*	0	2007	J Stat Software
Palmblad *et al.*	0	2007	ASMS
Lange *et al.*	0	2007	Bioinformatics
Wang *et al.*	0	2007	Biostatistics
Mueller *et al.*	0	2007	Proteomics
Listgarten *et al.*	0	2007	Bioinformatics
Fischer *et al.*	2	2007	BMC Bioinformatics
Csenki *et al.*	3	2007	Anal Bioanal Chem
Åberg *et al.*	0	2008	J Chrom A
De Groot *et al.*	0	2008	Proteomics
Suits *et al.*	0	2008	Anal Chem
Shinoda *et al.*	0	2008	Bioinformatics
Christin *et al.*	1	2008	Anal Chem
Podwojski *et al.*	2	2009	Bioinformatics
Befekadu *et al.*	3	2009	IEE EMBS
Christin *et al.*	3	2010	JPR
Daszykowski *et al.*	0	2010	J Chrom A
Tomasi *et al.*	1	2010	J Chrom A
Bloemberg *et al.*	1	2010	Chem Intel Lab Sys
Eliasson *et al.*	0	2011	Curr Pharm Biotech
Sinkov *et al.*	0	2011	Anal Chim Acta
Befekadu *et al.*	3	2011	IEEACM TCBB
Tang *et al.*	3	2011	Prot Science
Ballardini *et al.*	6	2011	J Chrom A
Voss *et al.*	7	2011	Bioinformatics
Zhang	0	2012	ASMS
Struck *et al.*	1	2012	J Chrom A
Hoekman *et al.*	2	2012	ASBMB
Kaya *et al.*	3	2012	Inform Sciences

a comparative evaluation of the existing algorithms since they have no mechanism for quantifying the comparative strengths or weaknesses of the methods from the publications themselves. As pointed out by a recent paper, this process is as time consuming as it is difficult, given the oft-encountered difficulties of obtaining and then successfully running someone else's software [6]. Extensive comparative analysis reduces the practitioner's overall time commitment by reducing the number of algorithms under consideration as well as by providing a realistic expectation of performance, hopefully justifying the inevitable inconvenience of obtaining and operating new software. Often, evaluation is made much more difficult (if not impossible) when open source code is omitted in submission. While English descriptions and pseudocode assist in building intuition about an algorithm, they are lossy definitions that leave out essential details needed for code implementation. Besides time savings, requiring source code facilitates more expansive comparison through automation as well as providing the reviewers an easy metric to determine whether the method is suitably formally defined to be distributed and replicated or whether it is an ad-hoc agglomeration.

There are also secondary consequences to consider. Publication is an incentive that can drive innovation. If novel algorithms are not required to outperform extant ones, then innovation—true forward progress not necessarily achieved by mere invention—is less likely to occur. Finding the best choice in an expanding sea of mediocre choices then becomes a Herculean task sure to exhaust any practitioner. The practical result is that practitioners stop short of exhaustively evaluating all the possible options and choose based on some other criteria (e.g., popularity, ease of use, or familiarity). The inevitable outcome of the algorithm selection crapshoot are results poorer than what may otherwise have been.

Researchers (the algorithm makers) also suffer when comparative evaluation is neglected. In the face of burgeoning publication numbers, they encounter the same exhaustive search problem faced by the practitioner, but they also face a moral dilemma—the current environment makes it easy to generate many publications, yet very difficult to perform the sort of due diligence comparison advocated in this letter. A good comparison requires choosing among the several existing evaluation methods, each of which highlight only specific behavior. The choice is non-trivial—in alignment, metrics include metrics that evaluate the alignment in isolation [18, 19, 109], in combination with other data processing steps [6, 55], globally, and locally. One must also find data sets, which should include sufficient data representative of the different typical performance-affecting real-world characteristics (e.g., complexity of the data, variability of peptide concentration, number of unique and common peptides, extent and form of retention time shift in the data, etc.). What's more, there is no disincentive provided for publishing work untested against existing methods. Thus, left to their own devices, will the researcher ever behave in a manner that is not in his best interest, though it is in the best interest of the field? Apparently, not very often. Our experience suggests that the pattern we found in alignment algorithms applies to algorithmic approaches in proteomics and metabolomics generally, and it may extend to other bioinformatics subfields where we have less experience.

So what is the solution? The problem, we have found, does not lie in the lack of venue requirements for performance demonstration against state of the art algorithms. Interestingly, many of the papers with zero-comparisons came from journals that explicitly require authors to provide quantitative comparison with state-of-the-art methods. Similarly, though an openly available group of standard data sets and metrics as described here would greatly facilitate the evaluations petitioned for, authors in other fields manage to provide comparisons even without standardized metrics or open frameworks for evaluation.

We suggest that greater care be taken by editors and reviewers to require novel algorithmic contributions to contain a reasonable comparative quantitative evaluation with existing methods. New contributions should also include necessary elements to facilitate future comparisons with other algorithms such as source code and parameter setting guidance. Such an effort will inevitably maximize the outcome of practitioner results, encourage the widespread use of the highest-quality tools, and provide researchers an incentive to truly innovate.

3.1 Funding

This work was supported by the National Science Foundation Graduate Research Fellowship [DGE-0750759].

Chapter 4

Controlling for Confounding Variables in MS-omics Protocol: Why Modularity Matters[1]

Abstract

As the field of bioinformatics research continues to grow, more and more novel techniques are proposed to meet new challenges and improvements upon solutions to long-standing problems. These include data processing techniques as well as wet lab protocol techniques. While the literature is consistently thorough in experimental detail and variable-controlling rigor for wet lab protocol techniques, bioinformatics techniques tend to be less described and less controlled. As the validation or rejection of hypotheses rests on the experiment's ability to isolate and measure a variable of interest, we urge the importance of reducing confounding variables in bioinformatics techniques during MS experimentation.

In science, we generate questions and design experiments to test possible answers to those questions. The ideal experiment involves a carefully controlled system where the impact of changes to a single variable may be measured. In practice, achieving full control over a system is difficult because the systems of most interest tend to be immensely complicated. Especially for complex systems, confounding variables—variables whose behavior can be spuriously attributed to the variable we are explicitly testing—may introduce hidden bias (referred to as omitted-variable bias) and therefore undermine an experiment. The degree to which omitted-variable bias is minimized is directly related to the accuracy of information which may be gleaned from an experiment. The MS-omics (proteomics, lipidomics, and

[1] Smith, R., Ventura, D., and Prince, J.T.: **Controlling for confounding variables in MS-omics protocol: Why modularity matters**, *Briefings in Bioinformatics*, 2013

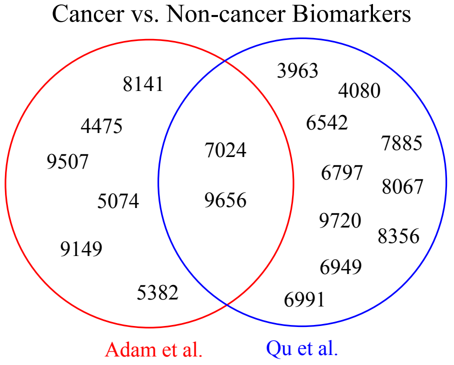


Figure 4.1: A comparison of m/z values of biomarkers for diagnosing cancer from two papers that used the same data set, with the only variability between them being the choice of bioinformatics tools used to post-process the data.

metabolomics prosecuted via mass spectrometry) community has been extremely careful to control for confounding variables in sample preparation/processing and mass spectrometry analysis (hereafter called lab protocol). But somewhat surprisingly, this meticulousness has not extended as uniformly to data processing protocols in these same experiments. It seems obvious that data processing can and will influence experimental outcomes, just as changes to lab protocol do, and this influence should be expected to grow as the complexity of algorithms and data sets increases.

Consider two experiments carried out on the same mass spec output files in search of drug biomarkers. In [2] and [77], the same group conducted two analyses of the same data set to predict cancer biomarkers. Despite the fact that the experimental variation was limited to the choice of post-processing bioinformatics tools (in this case, classification algorithms), the experiments yielded only two mutual m/z features (see Figure 4.1). The diagnostic biomarkers selected as well as the sensitivity and specificity of the diagnostic changed due solely to the data processing protocol details (see Table 4.1). Data processing protocol can and does influence experimental outcomes.

In data processing protocol, just as in lab protocol, limiting confounding variables boils down to limiting novel aspects under experiment. We suggest three guidelines to mitigate data-processing-related omitted-variable bias in MS-omics.

First, bioinformatics methods must be sufficiently described to permit replication, and parameters must be set to the established community standard independently demonstrated as effective in the literature. Although an explicit and careful protocol does not remove omitted-variable bias, it does make the potential confounding variables more obvious. It is hard to find a paper that does not include meticulous lab protocol details: sample preparation and source, sample storage conditions, machine manufacturer and calibration settings, etc. Unfortunately, when it comes to data processing, detailed descriptions are far too often replaced with descriptive snippets far too vague to reproduce the described protocol. In some papers, bioinformatic details are even simply relegated to a flow-chart box with a generic label like data pre-processing. No paper that compressed all the details of the source, composition, and preparation of a sample into a single flow-chart box labeled sample prep would ever pass peer review. Bioinformatic tools are unfortunately replete with free parameters that dramatically impact performance. If existing research suggests optimal parameter settings for a given situation, such settings should be employed. If not, a reasonable search of the parameter space ought to be conducted and reported. A suboptimal parameter setting can lead to lurking variable effects such as differential performance incorrectly attributed to the variable under experimentation. Although minimal reporting requirements suggested by HUPO-PSI (MIAPE, MIAMET, MIAME, etc.) are a step in the right direction, they do not require the reporting of all software parameters [3]. Consequently, a paper can meet the HUPO-PSI minimum reporting standards and still be completely unreproducible.

Second, new bioinformatic tools or unproven parameter settings ought to be presented and evaluated independently from studies designed for clinical outcomes. It is already

Table 4.1: A comparison of m/z values of biomarkers to diagnose cancer from two papers that used the same data set, with the only variability between them being the choice of bioinformatics tools used to post-process the data.

Study	Features	Sensitivity	Specificity
[2]	9	83%	97%
[77]	12	97-100%	97-100%

accepted that a new lab method deserves its own paper in which there is sufficient room to describe the method in reproducible detail as well as to ascertain its strengths and weaknesses in controlled experiments over a variety of data sets. The same standard ought to apply to bioinformatics methods. All too often, a paper whose focus is answering a chemical, biological, or clinical question is used as a vehicle to present a novel data processing method. Introducing a new variable in order to study another variable should at the least be somewhat disconcerting to any scientist. It is far more clear and appropriate to present novel methods in their own right.

Third—and most importantly—whenever possible, bioinformatics algorithms ought to be implemented following the single responsibility principle (SRP)—each module should have only one responsibility. In other words, algorithms ought to do one thing and do it well. This is not only a good programming philosophy but also a good experimental protocol philosophy that is at the heart of the scientific method—isolate and measure the variable of interest. New data processing methods, when coded modularly, are plug-in compatible with existing pipeline modules. Plug-in compatibility allows for a quick and comprehensive evaluation of new methods to ascertain downstream effects in the MS-omics pipeline. This approach has been implemented in frameworks for MS analysis (see, for example, mzMine 2 [73] and OpenMS [100]), yet new algorithms are consistently presented independently of these frameworks. Not only are these new contributions more difficult to use and evaluate due to their independent packaging, their non-modular interfaces and secondary functionality (visualizations, data import/export, etc.) are usually second rate to the full modular frameworks mentioned above. Packaging an alignment algorithm with yet another 2-D LC-MS display makes about as much sense as bundling a newly invented pipette with a second-rate centrifuge. Modularity not only facilitates the isolation of new variables but also decreases the learning curve by cutting down the need to install new software, learn new interfaces, deal with new file types, and facilitate transfer to the existing workflow. What's more, because of the number of confounding variables and added obfuscation, lack of modularity decreases the ease and transparency of evaluation against other methods, stifling innovation and community progress [98].

We suggest that practitioners treat confounding variables in the MS-omics toolkits with as much care as they do with confounding variables in the mass spec experimental protocol. A modular approach to bioinformatic tool development will help minimize omitted-variable bias, make bioinformatic tools interchangeable parts in the data processing pipeline, and facilitate extensive evaluation in controlled conditions before use in clinical application.

4.1 Key Points

- Choice of bioinformatic data processing algorithms and parameters affect the outcome of MS-omics experiments.
- Mitigation of confounding variables is just as important for data processing portions of the experiment as they are for lab portions and should be treated with as much care in practice and detail in publication. Each data processing variable necessary to reproduce the results ought to be reported in the article or supplemental information, including software choices and parameter settings.
- Novel algorithmic protocols ought to be introduced in their own dedicated article complete with either open source code or sufficient detail to reproduce the algorithm as well as sufficient evaluation with existing approaches to establish performance and detail shortcomings. It is not appropriate to introduce novel data processing techniques as a part of an experiment's protocol.

Chapter 5

Clarity in Concepts: A Novel, Unambiguous Nomenclature for MS-omics Data Structures[1]

Abstract

The comparison of analyte MS1 signal is central to many proteomic (and other omic) workflows. However, no standard vocabulary to describe the core data structures and algorithms involved exists. Without a standard, unambiguous nomenclature, literature searches, algorithm reproducibility, and algorithm evaluation for MS-omics data processing are nearly impossible. We propose a nomenclature which is constructed of a limited number of base terms along with qualifier terms allowing a vast number of data structures to be succinctly, precisely, and intuitively described. Using our terminology, we are able to show how terms from current official ontologies are too vague to explicitly map molecular entities to MS signals, and we illustrate the inconsistency and ambiguity of current colloquially used terms. We suggest this nomenclature as a beginning to, not the culmination of, the standardization process.

[1] Smith, R., Taylor, R., and Prince, J.T.: **Clarity in Concepts: A novel, unambiguous nomenclature for MS-omics data structures**, *Proteomics*, 2014 (in submission)

5.1 Introduction

Liquid-chromatography mass spectrometry (LC-MS) is a ubiquitously used platform for proteomic (and other ”omic”) investigations [93]. MS signal from hundreds to millions of ions can be quantitatively compared across experimental conditions in a fairly robust and repeatable way [64]. Analyte quantities are captured directly in MS signal (aka MS1), while analyte identities are often elucidated or confirmed using MS/MS (aka MS2) fragmentation spectra [64].

Confidently matching MS1 analyte signal between runs (”correspondence”) is difficult with complex samples[97], so a variety of approaches to circumvent this problem have been explored. Multiple reaction monitoring (MRM) can be effective for monitoring a relatively small number of pre-selected analytes with a high degree of confidence, but it is unsuited to discovery-based experiments. MS/MS based approaches (e.g. iTRAQ [81] and spectral counting [61]) are also popular alternatives. However, due to low MS/MS capture rates (10–20%) and low database match rates (<60%) [64], MS/MS driven approaches lack sensitivity compared to MS1-based approaches. And, although a data independent acquisition (DIA) approach may address some of the sensitivity deficiencies of MS/MS for *identification*, DIA does not of itself address difficulties in correspondence and quantitation. Hence, despite the availability of alternative approaches, the ability to match MS1 signal across experimental conditions is still highly desired.

Numerous efforts, large and small, have focused on using MS1 signal to compare analyte quantities. The most well-known of these are packages that implement the necessary algorithms for comparison of signal in an end-to-end fashion (e.g., SuperHirn [67], MaxQuant [22], XCMS [21], and Skyline [62]). However, the entire process is complex, and many algorithms have been developed which focus on individual steps such as extracting signal from the different data structures involved [128] and performing the final correspondence step [97]. Although many methods exist, very few have actually been compared against one another [98]. Some of this may stem from the lack of modularity of popular solutions (i.e., it is difficult to isolate and test subcomponents in monolithic software) [96], but a larger portion may be due to a deeper and more troubling problem: *few algorithms in this domain share any common nomenclature for the core data structures and processes needed for MS1 correspondence.*

The result of not having any kind of standard terminology for this domain is that LC-MS data processing is stagnated. Without consistent, clear terminology researchers have no handles for searching the literature, so they are unable to easily take advantage of modularized components that each solve a discrete problem. This leads to massive duplication of effort and few cross-tool evaluations since researchers are unaware of related efforts. A well defined vocabulary and problem domain also encourages and aids new-comers to the field resulting in more and better solutions to difficult data processing challenges. It is also much easier to re-implement solutions when both the *what* and *how* of a process are clearly understood. Hence, an increase in term clarity has immediate impact on reproducibility—a requirement firmly enforced for sample preparation and wet-lab processing protocols but which is almost completely unenforced for data processing descriptions [98].

What about using terms from existing HUPO-PSI [105] and IUPAC [69] standards and ontologies? As they stand, there are no terms in existing ontologies with enough granularity to precisely and reproducibly describe a data processing pipeline. Standards committees are best at crystallizing and refining accepted practice, but the onus to invent or select appropriate terminology lies foremost with the community itself, at least initially. A good example of this was the creation of a standard spectrum exchange format. The mzXML format was created and published by a small group of researchers [72]. After several years of use the HUPO PSI mass spectrometry working group produced the mzML format which was able to draw upon the experience gained from use of the mzXML format. An official nomenclature culminates with IUPAC [69] and HUPO-PSI [105] standards but the community cannot realistically expect nomenclature to begin there.

As a small but critical first step towards eventual standardization, we have identified core data structures and algorithms necessary for MS data processing. We also propose a nomenclature to describe them which is constructed of a limited number of base terms along with qualifier terms. This combinatorial design allows a vast number of data structures to be succinctly, precisely, and intuitively described. With precise terminology in hand, we are then able to show how terms from current official ontologies are too vague to explicitly map molecular entities to MS signals, and we illustrate the inconsistency and ambiguity of current colloquially used terms.

5.2 Proposed Terms

In order to maximize the information communicated in a term, we have created base terms, which describe the general concept under consideration, as well as qualifier terms, which specify any additional information possible about the genesis of the data concept. An overview of all terms is presented in Figure 5.1.

5.2.1 Base Terms

Generic terms allow us to refer to a specific data structure without necessarily adding detail about how it was processed. These terms are useful for algorithms that will take data structured in a certain way, no matter how it came to be in the current format.

Molecule - The unit that accepts charge. For instance, a lipid in a lipidomics experiment or a protein in a proteomics experiment.

Isotope - An isotope in this context consists of a molecule, at a particular charge state, with a certain number of neutrons (no distinction is made in this context among molecules where the neutron is associated with different atoms or kinds of atoms) (see Figure 5.1).

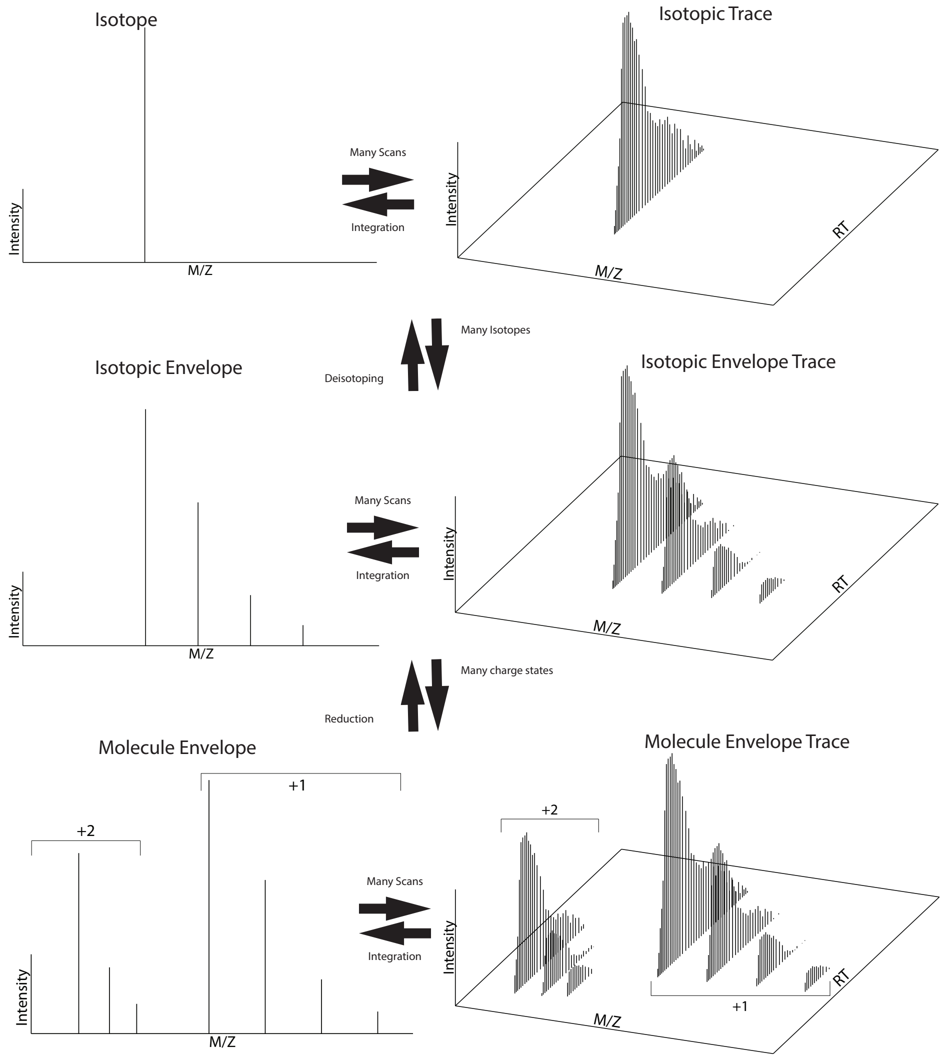


Figure 5.1: In this partial overview of the proposed nomenclature, the relationship between base concepts and some of qualifier terms is demonstrated. The qualifier *trace* adds a time dimension to a base concept. An *envelope* is a set of related instances across the m/z dimension. An *isotope* is a *molecule* at a particular charge state with a certain number of neutrons. An *isotopic envelope* is the unique impulse signal (at a specific RT) generated by one molecule/charge state combination consisting of one or more *isotopes* equally spaced m/z 1/z apart. A *molecular envelope* is the set of unique *isotopic envelopes* generated by one *molecule* across multiple charge states. An *isotopic trace* is the unique whole (meaning throughout RT) signal generated by the accumulation of instances of a given *molecule* at a given charge state whose molecular formula contains the same isotopic composition. An *isotopic envelope trace* is the unique whole signal generated by one *molecule*/charge state combination consisting of one or more isotopic *traces* equally spaced m/z 1/z apart. A *molecular envelope trace* is the set of whole isotopic envelopes generated by one molecule across multiple charge states.

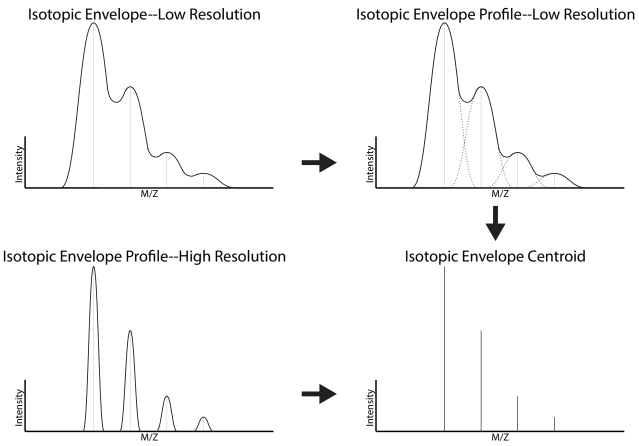


Figure 5.2: The terms *profile* and *centroid* in combination with the other terms proposed allow distinction between a low resolution convolved signal created by instances of a single *molecule* / charge state combination and the same concept in data from a high resolution instrument. These distinct concepts are indistinguishable under the IUPAC, HUPO-PSI, and colloquial term *peak*.

5.2.2 Qualifiers

Obviously, the most specific term possible should be used in each instance. For this purpose, we have introduced a set of qualifying terms that add specificity to the above-defined generic terms. The use of qualifiers allows us to encode previous processing steps into the term used to identify a data structure.

Profile

A *profile* is the data distribution generated by a single isotope (see Figure 5.2).

The qualifier *profile* allows us to specify concepts that are otherwise conflated between low-resolution and high-resolution data. For instance, in 2-d terms, an *isotope profile* is the data distribution thought to be a single *isotope*, and is found in high resolution *profile* data, while an *isotope envelope profile* is the convolution of several *isotope profiles* as found in low resolution profile data (see Figure 5.2).

Centroid

A *centroid* is the result of consolidating a *profile* into a single representative impulse containing the center of the previous *profile* and possessing the cumulative intensity of that profile (see Figure 5.2). This qualifier allows us to disambiguate between distinct concepts such as a *centroid isotopic envelope* and a *profile isotopic envelope*. Note that the assumption is that all data is centroided unless otherwise stated.

Envelope

An *envelope* connotes a discrete collection of things across the m/z dimension. For example, when we couple *envelope* with *isotope*, we get *isotopic envelope*, the unique impulse (meaning at a specific retention time (RT)) series generated by one molecule / charge state combination consisting of one or more isotopes equally spaced m/z 1/z apart (see Figure 5.1). By coupling *molecule* with *envelope*, we get *molecular envelope*, the set of unique isotopic envelopes generated by one molecule across multiple charge states (see Figure 5.1).

Trace

A *trace* implies a signal that extends into the RT dimension. For example, when we combine *isotopic envelope* and *trace*, we get an *isotopic envelope trace*, which is the unique whole (meaning throughout RT) accumulated (meaning throughout a run) signal generated by one *molecule* / charge state combination consisting of one or more *isotopic traces* equally spaced m/z 1/z apart (see Figure 5.1). Likewise, an *isotopic trace* the unique whole (meaning throughout RT) signal generated by the accumulation of instances of a given *molecule* at a given charge state whose molecular formula contains the same isotopic composition (see Figure 5.1). A *molecular envelope trace* is the set of whole (meaning throughout RT) isotopic envelopes generated by one molecule across multiple charge states (see Figure 5.1).

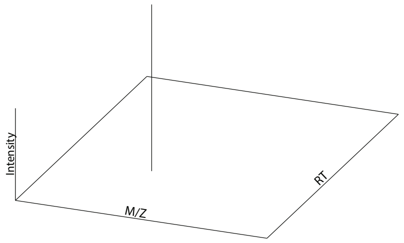


Figure 5.3: *Integrated isotopic trace* - the isotope produced by summing all constituent points in an isotopic trace.

Integrated

An integrated object has been summed through the RT dimension. For example, if we take an isotopic trace and sum its constituent centroids (or profile points), we will end up with a single 3-tuple consisting of m/z, RT, and intensity that can accurately be called an isotope (see Figure 5.1). However, by calling it an *integrated isotopic trace*, we retain a unique description of the original data structure as well as the transforming process used (see Figure 5.3). An *integrated isotopic envelope trace* is the sum of the constituent points in the *isotopic traces* contained in the *isotopic envelope trace*. In appearance, it is identical to the isotopic envelope in Figure 5.1.

Average

The data structures described by the qualifier *average* are, in appearance, the same as those in *integrated*, however the process to generate them involves taking the average of the intensity of the composite points, not the sum.

Instantaneous

The qualifier *instantaneous* implies that this object is a spectral slice of a trace object at a given RT. The *instantaneous* objects look exactly like those with that are *integrated*, however, this qualifier indicates that we are looking at a slice of the data structure in time, not a summation or average of the data through time.

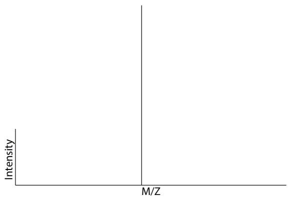


Figure 5.4: *Deisotoped isotopic envelope* - the composition of all *isotopes* in an *isotopic envelope*.

Max

The qualifier *max* implies that this object is the spectral slice of a trace object at a the RT of greatest intensity. Max objects look exactly like those with that are *integrated*, however, this qualifier indicates that we are looking at a slice of the data structure in time, not a summation or average of the data through time.

Deisotoped

The qualifier *deisotoped* implies that the object has been combined through the m/z dimension, such as a *deisotoped isotopic envelope*, the consolidation of all *isotopes* in an *isotopic envelope* (see Figures 5.4 and 5.5).

Reduced

The qualifier *reduced* implies that the object has been combined through reducing charge states to the lowest common charge state. For instance, a *reduced molecular envelope* is the set of the composition of all *isotopic envelope* in the *molecular envelope* (see Figure 5.1, bottom left to middle left).

Candidate

A *candidate* signal is one which is suspected to be a certain data structure, but has not yet been processed. For example, a mound of signal could be an *isotopic envelope trace*, so we call it a *candidate isotopic envelope trace*.

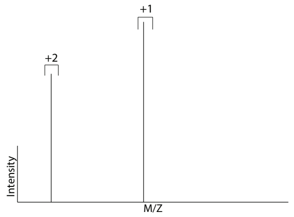


Figure 5.5: *Deisotoped molecular envelope* - the set of the composition of all *isotopes* in each *isotopic envelope* in a *molecular envelope* (see Figure 5.5).

Detected

A *detected* signal has been extracted from the full data set. Since this is the default assumption, the qualifier is rarely if ever used except to distinguish from a *candidate* or *theoretical* signal component.

Theoretical

A signal constructed *in silico* is said to be *theoretical*, as it is not an observation derived from observed experimental measurements.

5.3 Current Terms and Usage

To those unfamiliar with the field of MS-omics, it may be worth asking why it is that a new vocabulary is needed for the basic data structures of MS-omics. Surprisingly, there has been no effort of which the authors are aware to generate an unambiguous nomenclature for these concepts.

5.3.1 Why Current Official Terms are Incomplete

IUPAC [69] and HUPO-PSI [105] are organizations that specialize in standardizing nomenclature. Their significant and useful controlled vocabularies address all aspects of MS experimentation—including wet lab protocol and instrumentation—and have done much to improve reproducibility. The motivation at the heart of our proposed nomenclature is to suggest a nomenclature that explicitly maps causal molecular entities to the signals they produce. However, the current HUPO-PSI and IUPAC data concepts terms do not yet capture this level of granularity and clarity, which is an absolute must in order to achieve data processing reproducibility.

The HUPO-PSI-MS OBO has more MS data processing terms than IUPAC. Most are extremely generic. For example, they provide a term *mass spectrum* to refer to any segment of data with m/z and abundance axes: "a plot of the relative abundance of a beam or other collection of ions as a function of the mass-to-charge ratio (m/z)." They also provide a complementary term to refer to the time and abundance axes: *chromatogram*. However, these terms are sufficiently generic that one or both terms can accurately be applied to all of our above provided concepts. Additionally, the term *profile spectrum* as defined could apply to a combination of signals of any m/z width. Likewise, the term *peak* as defined could refer to any signal in the entire run.

In addition to these generic terms, the HUPO-PSI-MS OBO provides two specific data concepts: *deisotoping* and *charge deconvolution*. *Deisotoping* is referred to as "the removal of isotope peaks to represent the fragment ion as one data point and is commonly done to reduce complexity. It is done in conjunction with the charge state deconvolution." The concept described is worthy of a definition, but the one provided can be improved upon. A fragment ion is not a data signal, but a molecular object. However, deisotoping is an operation on a data signal. Additionally, this term should not be specific to fragment ions, but also applies to non-fragmented MS1 data, such as an MS1 *isotopic envelope*. Our nomenclature expands this term to include the logical wider use. *Charge deconvolution* is defined as "the determination of the mass of an ion based on the mass spectral peaks that represent multiple-charge ions." We have named this concept *reduction*. *Deconvolution* is already a widely used signal processing term (also used in MS processing e.g. in [15]) for resolving two overlapping signals into their constituent parts (see 5.2, top right). What's more, the definition focuses on mass determination. However, the real task at hand is transforming the charge of each *isotopic envelope* in the *molecular envelope* to match that of the lowest charged *molecule* in the *molecular envelope*, and then combining those signals in order to assess the mass and quantity of the entire *molecular envelope*. In addition to avoiding overloading, we consider *reduction* to be more intuitive than *deconvolution*.

Both HUPO-PSI and IUPAC use a similar definition of the term *ion*: an atomic, molecular, or radical species with a non-zero net electric charge. We agree that this concept needs a name. The problem with this term is that it cannot distinguish between the instances of interest (e.g. proteins in a proteomics experiment, lipids in a lipidomics experiment) and the molecular charged units of any scale that are detected in a mass spectrometer experiment. In other words, while our term *molecule* is limited to a lipid or a protein, the term ion could also correctly refer to any of the constituent points in a *profile*, centroids derived from a *profile*, *isotopes*, and *molecular envelopes*.

IUPAC terms are similarly ambiguous. The four data processing terms we are aware of in the most recent specification include two terms whose distinction is unclear to us: 1) *isotope cluster* - "group of *peaks* representing *ions* of the same elemental composition, but different isotopic compositions;" 2) *isotope pattern* - "set of *peaks* related to *ions* with the same chemical formula but containing different isotopes that have a particular pattern associated with the relative abundance of the isotopes." As defined, *isotope cluster* and *isotope pattern* can correctly be used to refer to all combinations of *isotopic envelope* and its qualifiers.

Finally, IUPAC shares a similar definition as HUPO-PSI for *peak*, "a localized region of relatively intense detector response in a mass spectrum when *ions* of a specified m/z are detected." As a catch-all, this term is useful, however it does not relieve the need for specific terms for each different concept that peak could refer to.

5.3.2 Ambiguous Colloquial Terms

The industry has not yet adopted a standard for MS data structure concepts. Consider, for instance, the usage of two of the most common labels for MS-omics data structures. These lists are by no means exhaustive in references or instances.

The term *feature* is used for:

- An *isotopic envelope* [56, 110, 128].
- A *candidate isotopic envelope* [67].
- A *deisotoped integrated isotopic envelope* [55, 107].
- An *integrated isotopic trace* [64].

The term *peak* is used for:

- A *profile* [22].
- A *centroid* [25, 54, 128].
- An *isotopic trace* [25, 25, 27, 110, 128].
- A *deisotoped integrated isotopic envelope* [64, 129].
- An *isotopic envelope* [128].
- An *integrated isotopic trace* [107, 128].
- An *isotope* [11].
- An *instantaneous isotopic envelope* [132].

It should be abundantly clear that these terms convey very little useful information—certainly insufficient information for reproducibility. These terms are used with so little care that even the attachment of a very specific qualifier does not relieve the lack of specificity. For example, adding the qualifier *monoisotopic* to *peak* could mean:

- An *isotopic trace* where the signal is generated by a light isotope [110].

- The most abundant *isotopic trace* in an *isotopic envelope* [105].
- An *integrated isotopic trace* where the signal is generated by a light isotope [132].
- A *deisotoped integrated isotopic envelope* [114, 128].

These examples briefly illustrate the ubiquity of overloading (using one term to mean more than one concept). Overloading treats a term as a variable, whose meaning must be defined in detail for the scope of each publication it appears in. An adequate definition takes significant thought, some space, and usually a descriptive image. There simply isn't ample space in each manuscript to define a custom set of terms for MS-omics data processing. This results in insufficient definitions for terms or no definition at all.

For example, the terms *isotopic peaks* and *isotopic multiplets* do not convey a clear meaning and are undefined in the manuscript where they appear [130]. It is unclear if a *peak/multiplet* is dealing with an *isotopic trace*, an *integrated isotopic envelope trace*, a *max isotopic envelope trace*, or an *instantaneous isotopic envelope trace*. The paper describes a decharging algorithm for *isotopic envelopes*, but depending on what definition you adopt for these terms, you will get a very different result.

As another example, consider a review paper that describes the algorithmic composition of several approaches to data processing problems [25]. To allow for the use of mathematical algorithm descriptions, the author provides a key where symbols are defined for certain MS data constructs. These include symbols for *peak* area, number of *chromatograms*, *peak* maximum, *peak* end, *peaks* detected in a mass channel, raw height of *peak*, and *peaks* detected in a *chromatogram*. But what is a *peak*? What is a *chromatogram*? As seen from the citations in this paper, these terms are not universally defined, and the author does not define them. Subsequently, the algorithms in the paper are irreproducible unless the reader is able to correctly guess the definition of these terms intended by the author.

Reproducibility is, in fact, at the heart of the nomenclature problem. An algorithm description is rendered useless if the data structure terms used within it are ambiguous or undefined. In a modular approach to pipeline algorithm creation and testing [96], data processing methods prior to the pipeline module of interest have to be exactly describable with concise terms. If an algorithm says it operates on *monoisotopic peaks*, how does the practitioner know to which format it refers? What's more, in evaluating algorithms, knowing the exact format of the input data informs interpretation of the algorithm. For instance, we know that any alignment algorithm that takes as input *deisotoped candidate isotopic envelopes* is subject to the error introduced by convolved *isotopic envelopes* mistakenly assumed to be a single *isotopic envelope.* As another example, any alignment algorithm that uses *deisotoped isotopic envelope* data as an input is not putting to use the added information available in the complete *isotopic envelope trace.*

Our suggested vocabulary eliminates most if not all of the ambiguity in the current naming schemes employed in the literature. The following examples illustrate how the proposed vocabulary untangles the currently obfuscated terms in use.

Isotopic envelope trace describes a concept for which the following terms have all been used: an empheluting isotopic distribution [7], a emphchromatogram [25], an *isotope series* [25], an *isotope pattern* [22], an *isotope-resolved mass spectrum* [103], an *ion series* [108], and an *isotopic cluster* [15, 114]. None of these terms differentiate between the concepts we refer to as *isotopic envelope trace, instantaneous isotopic envelope, max isotopic envelope*, etc.

Isotopic traces have been referred to as *eluting isotopes* [7], *single ion chromatograms* [25], *peaks* [22, 25, 27], *mass spectra* [11], and *peak hills* [22]. Each of these terms are unclear. The problem with the term *chromatogram* is that is does not specifically refer to the elution profile of a single *isotope.* For example, an *extracted ion chromatogram* is an m/z slice of data that can extend across an entire run's RT. Any term that uses *peak* is bound to be confusing due to the overuse of the term. Like *chromatogram*, a *mass spectrum* can technically stretch across an entire m/z range and therefore does not specifically describe the m/z window of a specific *molecule.*

Integrated isotopic envelope has been called an *isotope pattern* [27]. However, many other concepts can accurately be called *isotope patterns*, such as a *max isotopic envelope* or an *averaged isotopic envelope.*

5.4 Conclusion

The ever-increasing number of MS-omics experiments drives a thriving MS-omics data processing algorithms field. However, the lack of an unambiguous vocabulary for MS-omics data structures has created serious challenges for reproducibility and evaluation of data processing algorithms.

In this paper, we have highlighted the ambiguity of current vocabulary for MS-omics. We propose an unambiguous vocabulary together with a visual lexicon for the proposed terms. By adopting these terms, authors can facilitate reproduction of their algorithms succinctly by providing a crystal-clear set of meanings for terms they use, vastly improving the reproducibility of their work.

5.4.1 Acknowledgements

RS was supported by the National Science Foundation Graduate Research Fellowship [DGE-0750759]. The authors declare no conflicts of interest.

Chapter 6

A Coherent Mathematical Characterization of Isotope Trace Extraction, Isotopic Envelope Extraction, and LC-MS Correspondence[1]

Abstract

Liquid chromatography-mass spectrometry is a popular technique for high-throughput protein, lipid, and metabolite comparative analysis. Such statistical comparison of millions of data points requires the generation of an inter-run correspondence. Though many techniques for generating this correspondence exist, few if any, address certain well-known run-to-run LC-MS behaviors such as elution order swaps, unbounded retention time swaps, missing data, and significant differences in abundance. Moreover, not all extant correspondence methods leverage the rich discriminating information offered by isotope envelope extraction informed by isotope trace extraction. To date, no attempt has been made to create a formal generalization of extant algorithms for these problems. By enumerating extant objective functions for these problems, we elucidate discrepancies between known LC-MS data behavior and extant approaches. We propose novel objective functions that more closely model known LC-MS behavior. Through instantiating the proposed objective functions in the form of novel algorithms, practitioners can more accurately capture the known behavior of isotope traces, isotopic envelopes, and replicate LC-MS data, ultimately providing for improved quantitative accuracy.

[1] Smith, R., Prince, J.T., and Ventura, D.: **A coherent mathematical characterization of isotope trace extraction, isotopic envelope extraction, and LC-MS correspondence**, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2014 (in submission)

6.1 Introduction

Liquid chromatography-mass spectrometry (LC-MS) is a popular technique for elucidating the composition of liquid samples. Data processing considerations are essential to accurately determine the identity of molecules (analytes such as lipids or peptides) contained in the sample (a process called identification), as well as their quantity in sample (a process called quantitation).

Information about sample quantity is captured directly in survey scans, or MS (aka MS1) data. Fragmentation spectra of one or more analytes constitute MS/MS (or MS2) data, and this information is typically used to corroborate or ascertain the identity of a molecule. Partitioning/clustering MS1 signal from complex samples and mapping the signal to other analyses (correspondence) is challenging. Some quantification strategies bypass these challenges by using information derived directly or indirectly from MS/MS data. These methods include spectral counting [16] and isobaric tags for relative and absolute quantitation (iTRAQ) [122]. Though these methods have been successful, the amount of quantifiable signal embedded in MS1 data is estimated to far exceed what is currently available by MS/MS [64]; however, most MS1 data remains unused by current software. Hence, improving methods for partitioning and mapping MS1 signal stands to significantly (˜10 fold) increase the sensitivity of a typical label-free or isotope-labeling MS-omics experiment, both for experiments currently being run and for past experiments where raw data is still available.

Subdivision of raw mass spectrometer output data into smaller signal partitions attributed to specific analytes in the sample is critical prior to achieving analyte identification and quantification. The larger partition unit, called an isotopic envelope trace, is the signal pattern generated by each analyte/charge combination (see Figure 6.1).[2] Because mass spectrometers can only detect charged analytes, the sample must be subjected to an ionization method, which imputes a charge on each detected analyte. Since multiple instances of each component exist in the sample, and since each instance is charged independently,

[2] To avoid ambiguity, this manuscript uses the nomenclature described by Smith *et al.* [95]

there exist in each output the signals of multiple analytes, each with (potentially) multiple charge states. These create a distinct signal—the isotopic envelope trace—for the total signal detected for each analyte/charge state combination. Each isotopic envelope trace is composed of a series of isotope traces, which are manifestations of the fact that each analyte is composed of chemically similar compounds that differ in the weight of certain isotopes (such as ^{12}C vs ^{13}C). At each charge state, each molecular variant of the analyte is detected at a particular m/z offset, creating one isotope trace per molecular variant/charge-state/analyte combination.

Mass spectrometry data, in its raw form, is not ideal for isotope trace extraction or subsequent processing. After internally accumulating signal over discrete time slices, the mass spectrometer outputs raw data condensed into the form of many narrow profiles wherever signal is present. Conversion to centroid mode integrates the abundance of each of these profiles into a single tuple called a centroid. This is considered a routine conversion for which ample software is readily available. We adopt the typical convention of using centroid data.

Despite the ubiquity of LC-MS experiments, to the best of our knowledge, no concise, complete description of the LC-MS isotope trace and isotopic envelope extraction problems exists. Here, we describe constructs for isotope traces and isotopic envelopes, as well as formally describe the relationship of centroids, isotope traces and isotopic envelopes. In this context, we review extant objective functions for isotope trace extraction, isotopic envelope extraction, and correspondence. Finally, we propose novel objective functions for each of these tasks that address shortcomings in current approaches.

6.2 Isotope Trace Extraction

The most important data processing step in a typical quantitative LC-MS pipeline is isotope trace extraction [14]. Clustering centroids into isotope traces is a non-trivial problem due to the many sources of noise affecting centroid mass and abundance. Sources of noise affecting centroids include chemistry effects due to chromatography, abundance inaccuracy due to

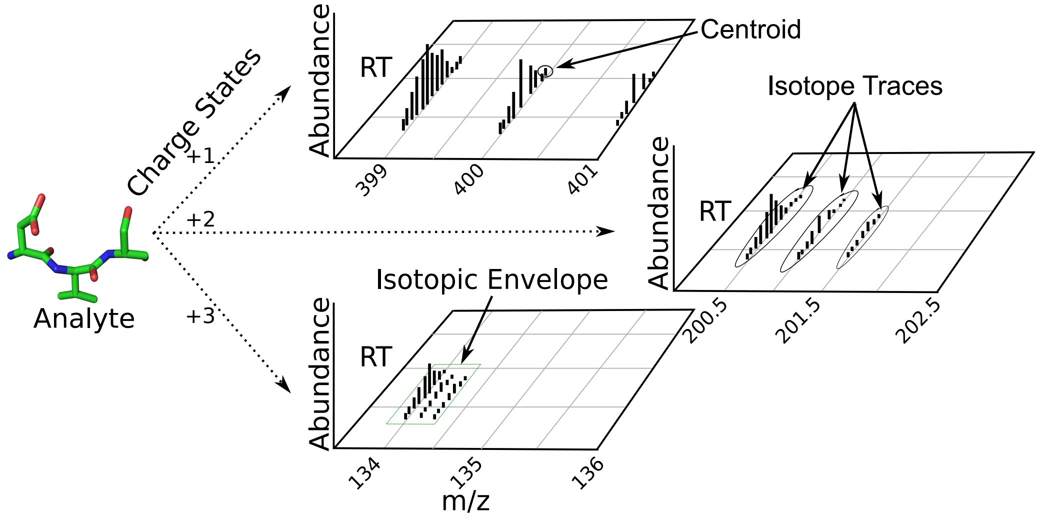


Figure 6.1: An LC-MS sample is composed of many instances of many classes of analyte. Each detected instance of an analyte is ionized to a charge state. The signal produced by each charged analyte is accumulated as a function of the mass of the analyte, its charge (together composing the mass-to-charge ratio (m/z)), and the time at which it is detected (dictated by the chromatographic system in use).

ionization efficiencies, m/z deviation due to machine calibration, occlusion/adulteration of low-abundance signal due to dynamic range limitations, and compounded inaccuracies in mass-to-charge ratio (m/z) and abundance due to centroid construction. Of course, these complications are propagated from the clustering of isotope traces to the clustering of isotopic envelopes to the identification of cross-experiment correspondence.

A centroid is denoted as $c = (\mu, \tau, \alpha)$ where μ, τ, α are values for m/z, retention time (RT), and abundance, respectively. A single MS run produces a set of centroids $C = \{c_i\}_{i=0}^{n}$, where n can readily reach into the millions.

An isotope trace $F \subset C$ is defined as a set of centroids: $F = \{c_i\}_{i=0}^{m}$, with each set F constrained so that all members of a given isotope trace F are within a distance threshold θ from other centroids in their neighborhood Υ (see Figure 6.2):

$$\max_{j \in \Upsilon_i} \; \delta_F(c_i, c_j) < \theta^{\mu,\alpha,\tau} \tag{6.1}$$

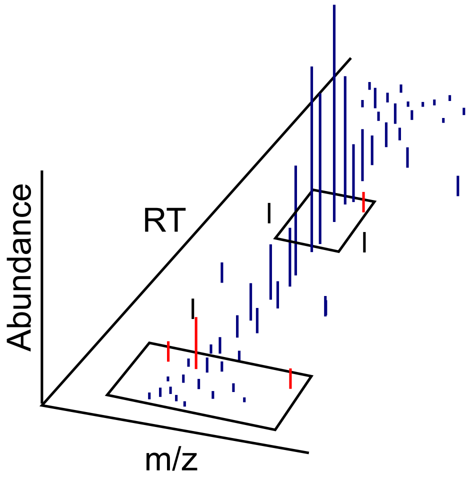


Figure 6.2: Each box illustrates an example candidate local neighborhood Υ defined by an algorithm-specific m/z and RT window. Blue centroids indicate the centroids pertaining to the isotope trace, while red centroids have been rejected due to differences in abundance, m/z, and/or RT compared to other centroids in Υ.

where θ is a function of centroid m/z, RT, and abundance, δ_F is a distance function based on m/z, RT, and abundance, and Υ is a neighborhood demarcated by m/z, RT, and abundance. Additionally, the slope of a (abundance-weighted) linear regressor estimate for an isotopic trace is very nearly infinite (in the $m/z, RT$-plane). One way to formalize this is to use a weighted, inverse variant of the Theil-Sen estimator as follows (see Figure 6.3):

$$\frac{\sum_{c_i,c_j \in F} \frac{c_j^\mu - c_i^\mu}{c_j^\tau - c_i^\tau} c_j^\alpha c_i^\alpha}{\sum_{c_i,c_j \in F} c_j^\alpha c_i^\alpha} \approx 0 \qquad (6.2)$$

where c^α is the abundance of centroid c and c^μ is the m/z of centroid c.

Note that the behavior of isotope traces are dependent on all three MS dimensions although many common approaches to isotope trace extraction ignore one or more of these dimensions. For example, most proprietary MS software uses hard m/z bins for isotope trace extraction.

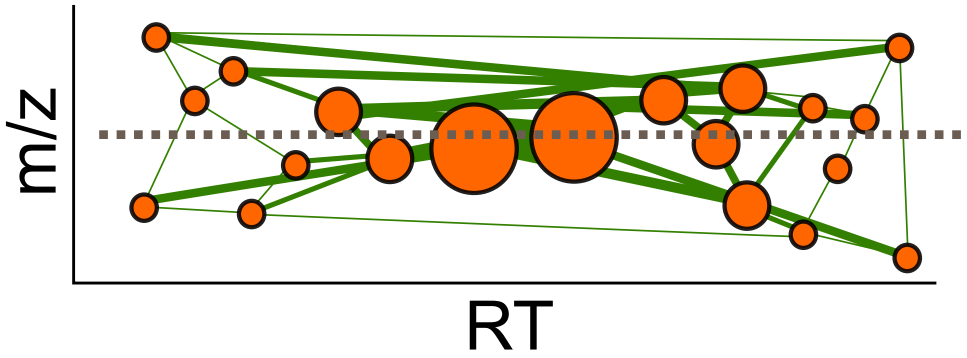


Figure 6.3: One way to characterize the relative lack of variance in m/z (compared to RT) of an isotopic trace is by using an inverse variant of the Theil-Sen estimator—a fully-connected graph is constructed with edges connecting each pair of centroids (circles whose radius indicates abundance), and weighted by the abundance of its connected centroids (represented by line thickness). An isotopic trace will have a weighted average (inverse) slope of approximately zero (not all connections shown).

6.2.1 Extant Objective Functions

The prominent algorithms for isotope trace extraction include centWave [104], MatchedFilter [104], centroidPicker [73], massifquant [21], and MaxQuant [22].

MatchedFilter operates on the simplifying assumptions that 1) isotope traces are completely contained within pre-processed hard m/z bins and 2) the shapes of all isotope traces in a run can be fit to the same shape. MatchedFilter minimizes the error of a Gaussian fit over prospective isotope traces, by attempting to find the set of isotope traces $\mathcal{F}$, a scaling factor b_F, and mean retention time F^t for each isotope trace that minimizes the summed abundance error over all isotope traces. Note the use of a single, global variance σ, an average RT width for all $F \in \mathcal{F}$:

$$\lambda_F = \sum_{F \in \mathcal{F}} \sum_{c \in F} \left| b_F e^{\frac{-(c^\tau - F^t)^2}{2\sigma^2}} - c^\alpha \right| \tag{6.3}$$

The centWave algorithm extracts isotope traces that fit a scaled and translated Ricker wavelet ζ (commonly called a Mexican hat function). The fit is calculated as a convolution between the shape function and the signal intensity (abundance), so the goal is to maximize the objective function:

$$\lambda_F = \sum_{F \in \mathcal{F}} \sum_{c \in F} c^\alpha \zeta(c) \tag{6.4}$$

where

$$\zeta(c) = \left(\frac{1}{\sqrt{b_F}} \frac{2}{\sqrt{3}\pi^{\frac{1}{4}}} \left(1 - \left(\frac{c^\tau - t_F}{b_F}\right)^2\right) e^{\frac{-\left(\frac{c^\tau - t_F}{b_F}\right)^2}{2}} \right) \tag{6.5}$$

with isotope trace-specific scaling parameter b_F and translation parameter t_F chosen to maximize the convolutional fit over isotope trace F.

The algorithm centroidPicker uses heuristic operations on a neighborhood graph to separated the data into connected components. It connects an undirected graph $G = (C, N)$ of centroids where the edges N are constrained such that:

$$N = \left\{ (c_i, c_j) \;\middle|\; \begin{array}{c} \delta_c(c_i, c_j) < \delta_c(c_i, c_k) \forall_{k \neq j} \\ c_i^\alpha > \theta \text{ and } c_j^\alpha > \theta \end{array} \right\} \tag{6.6}$$

for some intensity threshold θ and centroid distance function δ_c, resulting in G being composed of one or more connected components, each considered one isotope trace. Thus, $\mathcal{F} = \{F_i | \forall c_k \in F_i, \exists_{c_l \in F_i} \{c_l \in \Upsilon(c_k)\}\}$, where the neighborhood function $\Upsilon(c)$ returns the set of nodes connected to c (and is symmetric because G is undirected).

The objective functions for massifquant and MaxQuant define $\mathcal{F}$ as the set of all F formed by iterating over values of time t, and adding c if $c^\tau = t$ and $|c^\mu - c_*^\mu| < \epsilon$, where $c_* \in F$ and $c^\tau - c_*^\tau \leq c^\tau - c_j^\tau$ for all $c_j \in F$. For massifquant, ϵ is prescribed by a Kalman filter induced from the variance in c^μ and c^α for all $c_j \in F$ such that $c_j^\tau < t$, with the added constraint that c^τ be unique in F. MaxQuant defines ϵ simply as a distance threshold of 7ppm m/z.

6.2.2 Proposed Objective Functions

We define F^μ, the m/z of isotope trace F, given by the weighted m/z of its component centroids:

$$F^\mu = \frac{\sum\limits_{c \in F} c^\alpha c^\mu}{\sum\limits_{c \in F} c^\alpha} \tag{6.7}$$

and using it propose an alternative objective function for isotope trace extraction:

$$\lambda_F = \sum_{F \in \mathcal{F}} \sum_{c \in F} \left| b_F(\tau) e^{\frac{-(c^\tau - F^t)^2}{2\sigma_F^2}} a_F(\alpha) e^{\frac{-(c^\mu - F^\mu)^2}{2h(\alpha)^2}} - c^\alpha \right| \tag{6.8}$$

where, again, centroid clustering $\mathcal{F}$ and retention time means F^t are chosen to minimize the Gaussian fit error; however, rather than using a single global variance in the RT dimension, each isotope trace F has a local variance σ_F; in addition, the scaling factors have become time-dependent scalar functions $b_F(\cdot)$. The second Gaussian factor, parameterized by mean F^μ and variance function $h(\cdot)$, models the m/z width of the isotope trace, which is a function of the abundance α. Isotope traces splay at low abundance and narrow at high abundance; thus, both the variance $h(\cdot)$ and the scaling factors $a_F(\cdot)$ are modeled as functions dependent on the abundance α. Note that while variance is trace-independent (depending only on abundance), each isotope trace has its own scaling function (which in turn is dependent on abundance).

6.2.3 Alleviating Current Limitations in Isotopic Trace Extraction

Current objective functions for isotopic trace extraction fail to capture isotopic trace behavior formalized in this section: namely, a pattern of centroids forming a generally tight distribution through time around a specific m/z, with variation occurring as a factor of abundance, with normal abundance traces splaying at the beginning and end of elution, and lower abundance traces displaying high m/z variance in general. Moreover, isotope traces are skewed in time, with sharp onset of intensity followed by a post-peak long tail. The shape of traces is almost never strictly Gaussian (or even symmetric), as chromatography almost always deviates from the Gaussian in heading (which is more steep) and in tailing (which is less steep). Our objective functions account for each of these behaviors.

6.3 Isotopic Envelope Extraction

The LC-MS clustering problem is defined as a two-step partitioning problem. In the first step, isotope trace extraction, we require a partition ϕ of the set of all centroids C into the set of isotope traces $\mathcal{F}$, $\phi(C) = \{F_i\}_{i=1}^{r} = \mathcal{F}$ with the properties:

$$\bigcup_{i=1}^{r} F_i = C \text{ and } F_i \cap F_j = \varnothing \quad \forall_{F_i \neq F_j \in \mathcal{F}} \tag{6.9}$$

In other words, 1) all centroids are assigned to an isotope trace; 2) isotope traces can't share centroids. Because any sensor's detection of a physical system will deviate somewhat from the true physical system, we can expect MS detections to contain extraneous centroids. However, all signal ought to be accounted for (even if some identified "traces" eventually are identified as noise) and, in a platonic model, ought to be assigned to an isotope trace.

In the second step, isotopic envelope extraction, we require a partition ψ of the set of isotope traces $\mathcal{F}$ into the set of isotopic envelopes $\mathcal{E}$, $\psi(\mathcal{F}) = \{E_i\}_{i=1}^{p} = \mathcal{E}$ with the property

$$\bigcup_{i=1}^{p} E_i = \mathcal{F} \tag{6.10}$$

The choice of partitions ϕ and ψ is guided by a set of distance functions Δ that define distances between centroids, isotope traces, isotopic envelopes, etc. and objective functions λ_F and λ_E that describe "good" isotope traces and isotopic envelopes, respectively. The choice of distance and objective functions, along with choice of optimization procedure, characterizes an algorithmic approach for solving this clustering problem. A defining general property of isotopic envelopes, however, is the regular spacing between component isotope traces. In addition, for virtually all molecules from biological sources we expect that if there is an isotope with index j and an isotope with index $j+2$, then there exists an isotope with index $j+1$.

An isotopic envelope E is the set of isotope traces F_i that are produced by a given analyte/charge state combination: $E = \{F_i\}_{i=0}^{q}$ subject to the constraint that the m/z difference between each consecutive (assuming an ordering of centroids from least mass to greatest mass) isotope trace in E must be equivalent to $\frac{k}{z_E} + \epsilon$, where k is the mass of a neutron, z_E is the integer charge of E and ϵ is a noise tolerance parameter. That is, assuming an indexing function $\iota^{\mu} : \mathcal{E} \times \mathcal{N} \to \mathcal{F}$ that returns the ith least massive isotope trace in an isotopic envelope:

$$\iota^{\mu}(F, i+1) - \iota^{\mu}(F, i) = \frac{k}{z_E} + \epsilon, \quad 1 \leq i \leq |E| - 1 \tag{6.11}$$

The m/z m of the jth isotope trace in E must be roughly equivalent to

$$m = \frac{\tilde{m} + jk}{z} \tag{6.12}$$

where $\tilde{m}$ is the uncharged molecular weight of the ion.

Every isotope trace consists of signal from at least one isotopic envelope, and, in the case of overlapping isotopic envelopes, an isotope trace may be composed of signal from more than one isotopic envelope.

6.3.1 Extant Objective Functions

FeatureFinder [115] is an isotopic envelope extraction algorithm in OpenMS that searches directly for E. Although the details are not completely clear, it appears that the algorithm attempts to minimize

$$\lambda_E = \sum_{E \in \mathcal{E}} \sum_{c \in E} G_E(c) \tag{6.13}$$

where the G_E compute a comparison between the (μ, τ, α) values for a centroid and the expected centroid values obtained from a heuristic isotopic envelope shape. Note that isotopic trace extraction is ignored.

MSInspect [7], another approach to isotopic envelope extraction, groups all co-eluting signals and compares them to a simulated envelope calculated from a Poisson distribution parameterized by m/z, with the goal being to minimize the KL divergence between the Poisson distribution and the "distribution" of abundance in an instantaneous profile of the envelope at time τ:

$$\lambda_E = \sum_{F \in E, c \in^\tau F} \hat{P}(c^\alpha) \log \frac{\hat{P}(c^\alpha)}{P_m(c^\mu)} \tag{6.14}$$

where the notation $c \in^\tau F$ means that $c \in F$ at time τ, E is the maximal intensity (instantaneous) isotopic envelope (at time τ), $\hat{P}(\cdot)$ is the ratio of the intensity of isotope trace F (at time τ) to the total intensity of all isotope traces $F \in E$ (at time τ), and $P_m(\cdot)$ is the value of the Poisson distribution at c^μ.

6.3.2 Proposed Objective Functions

We propose an alternative objective function for isotopic envelope extraction:

$$\lambda_E = \beta I(E) + (1 - \beta) J(E), \quad 0 \leq \beta \leq 1 \tag{6.15}$$

where β is a relative importance weighting coefficient. The first term computes the deviation of member isotope traces from the expected charge-based m/z interval—we want the isotope traces in envelope E to fit expected m/z spacing:

$$I(E) = \sum_{\substack{F_i, F_j \in E \wedge \\ F_i^\mu < F_j^\mu \wedge \\ \forall_{F_k^\mu \in E} F_k^\mu > F_i^\mu \implies F_k^\mu > F_j^\mu \vee F_k = F_j}} |(F_i^\mu - F_j^\mu) - \frac{k}{z_E}| \tag{6.16}$$

The second term computes the deviation in elution time of member isotope traces—we want all the isotope traces in isotopic envelope E to co-elute within a small time window:

$$J(E) = \sum_{F_i, F_j \in E} F_i^\tau - F_j^\tau \tag{6.17}$$

where F^τ could be defined analogously to Equation 6.7, could be the maximum intensity for isotopic trace F or could be some other reasonable definition for isotopic trace elution time.

We want to optimize $\mathcal{E}$ and the z_E so that λ_E is minimized; that is, we want to find charge-state/isotopic-envelope pairs such that the errors in expected m/z and co-elution time are minimized.

The isotopic envelope extraction segment of the MaxQuant [22] algorithm is one of the possible instantiations of this objective function, though many possibilities exist for how to set the allowable m/z and RT error and how to generate the prerequisite list of isotope traces.

6.3.3 Alleviating Current Limitations in Isotopic Envelope Extraction

Isotopic envelopes are rich with data: the expectation of contiguous isotope traces with a uniform m/z charge gap, and similar maximal abundance across all isotope traces. Accounting for this behavior is not possible without adopting an isotope trace-centric approach to data extraction. Reliance upon maximal elution time alone—an approach that is susceptible to conflation with overlapping envelopes in complex samples—is not a sensitive approach in envelopes of lower abundance, where maximal elution times are not pronounced. Moreover, by first finding the isotope traces, the exact m/z of each isotope trace can be calculated using a weighted average, alleviating the need for larger than theoretically justified isotope trace gaps, which will not be sensitive in complex samples with overlapping isotopic envelopes. Instead, the proposed objective functions leverage a precise and reliable m/z charge gap and adjacency of isotope traces along with maximal elution times, using all the information in the data.

6.4 Correspondence

The final objective of almost every MS experiment is the differential analysis of more than one MS run. This comparison allows the identification of significant quantity and component differences, useful for applications such as drug design, disease treatment, biological processes research and chemical forensics. Correspondence yields a mapping between isotopic envelopes in different runs (see Figure 6.4), a prerequisite for differential analysis.

The combination of noise from within one run (enumerated above) and noise from run to run—most notable in retention time shifts, where an isotopic envelope appears at a different retention time or with a compressed or stretched RT length compared to another run—make LC-MS correspondence non-trivial.

The correspondence mapping should again optimize an objective function which, in turn, characterizes an algorithm choice for solving the correspondence problem.

6.4.1 Extant Objective Functions

According to a recent review on LC-MS correspondence algorithms [97], all extant approaches use either centroid data or a reduction of isotopic envelope traces into a single centroid. Of the almost sixty algorithms reviewed there, nearly all use the same objective function—finding a family of one-to-one partial functions $\chi_r : \mathcal{E}_r \rightarrow \mathcal{E}_*$ (a different function for each experimental run r), where $\mathcal{E}_*$ is the set of envelopes from a reference run, that minimizes global RT and m/z distance between isotopic envelopes (in any of their reduced forms, according to the authors):

$$\lambda_{corr} = \sum_{E \in \mathcal{E}_r} \delta(E, \chi_r(E))^{\tau,\mu} \qquad (6.18)$$

where $\delta()^{\tau,\mu}$ is a distance function defined over RT and m/z.

The continuous profile model (CPM) [58] uses a different objective function, and thus is free from the reference requirement that most other algorithms have, allowing for a

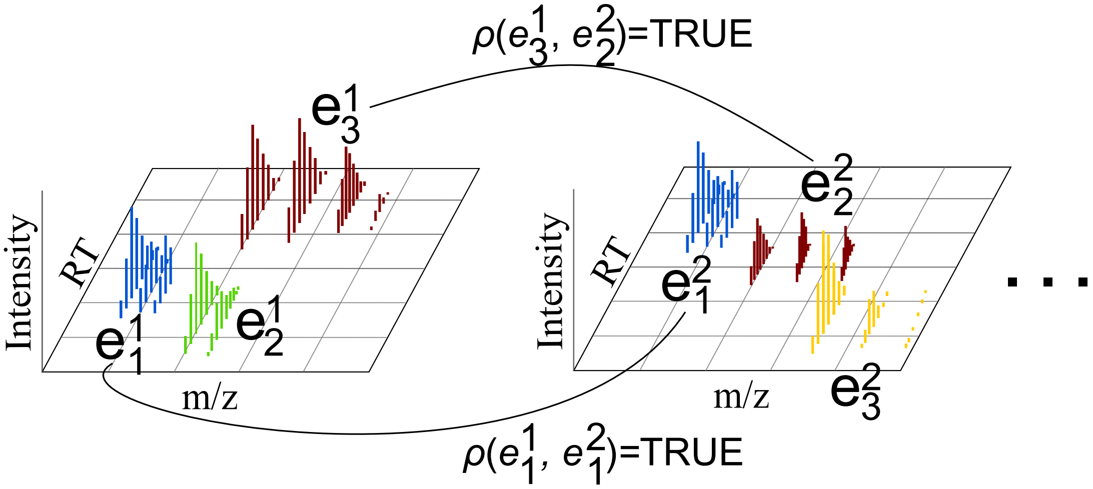


Figure 6.4: Objective functions for correspondence must allow a mapping from an isotopic envelope in one run to an envelope in another, or to none, if there is no corresponding isotopic envelope. Here, the unillustrated relations would yield FALSE.

symmetric solution (one that is not dependent on the choice of a reference run). Additionally, the mapping is somewhat more localized than that of most correspondence algorithms. CPM minimizes the log likelihood of differences between a hidden Markov model m^τ of the RT of a latent run and observed runs:

$$\lambda_{corr} = \log p(D|m^\tau) \tag{6.19}$$

where D is the set of observed runs.

6.4.2 Proposed Objective Functions

In contrast to existing LC-MS correspondence objective functions, the objective functions suggested here use the entire isotopic envelope. This allows greater discrimination by using isotope trace quantity and spacing to match isotopic envelopes from different runs. This extra discrimination is essential given the amount of RT variance and (to a lesser degree) m/z variance present in the data.

Let R be a set of runs, each of which has an associated set of isotopic envelopes $\mathcal{E}_r = \{E_i^r\}_{i=1}^{p_r}, 1 \leq r \leq |R|$ and let $\tilde{\mathcal{E}} = \bigcup_r \mathcal{E}_r$. We seek to find a binary equivalence relation ρ

that induces a set of *correspondence classes* over $\tilde{\mathcal{E}}$ that is reflexive (an envelope corresponds with itself), symmetric (if envelope E_1 from run 1 corresponds with envelop E_2 from run 2, then E_2 also corresponds with E_1) and transitive (if envelope E_1 from run 1 corresponds with envelope E_2 from run 2 and envelope E_2 corresponds with envelope E_3 from run 3, then E_1 corresponds with E_3); and if $\rho(E_i^r, E_j^s) = \text{TRUE}$, then for $k \neq i$, $\rho(E_k^r, E_j^s) = \text{FALSE}$ and for $k \neq j$, $\rho(E_i^r, E_k^s) = \text{FALSE}$ (an envelope from one run may have 0 or 1 matches from any other run; note that due to reflexivity, this also means that two non-identical envelopes from the same run never correspond).

This relation should minimize

- The difference in charge state between corresponding isotopic envelopes, δ_{charge}.
- The difference in m/z between isotope traces in corresponding isotopic envelopes, $\delta_{mz_{it}}$.
- The difference in elution duration between isotope traces in corresponding isotopic envelopes, δ_{dur}.
- The difference in isotope abundance ratios between corresponding isotopic envelopes, δ_{ratio}.
- The difference in m/z between corresponding isotopic envelopes, $\delta_{mz_{ie}}$.
- The number of singleton correspondence classes, δ_{orphan}.
- The difference in retention time between corresponding isotopic envelopes, δ_{rt}.

An objective function incorporating all of these variables can take many forms, with perhaps the simplest generalization being a weighted linear combination, with weighting coefficients ω allowing relative prioritization:

$$\begin{aligned}\lambda_{corr} = \sum_{\rho(E_1,E_2)} & \omega_{charge}\delta_{charge}(E_1,E_2) + \omega_{mz_{it}}\delta_{mz_{it}}(E_1,E_2) \\ & + \omega_{dur}\delta_{dur}(E_1,E_2) + \omega_{ratio}\delta_{ratio}(E_1,E_2) \\ & + \omega_{mz_{ie}}\delta_{mz_{ie}}(E_1,E_2) + \omega_{orphan}\delta_{orphan}(E_1,E_2) \\ & + \omega_{rt}\delta_{rt}(E_1,E_2) \qquad (6.20)\end{aligned}$$

with the summation over $\rho(E_1, E_2)$ meaning a summation taken over all pairs of envelopes $E_1, E_2 \in \tilde{\mathcal{E}}$ for which $\rho(E_1, E_2) = \text{TRUE}$. Given the weighting coefficients ω, the most desirable correspondence would be that induced by the relation ρ^* that minimizes λ_{corr} (see Figure 6.4),

$$\rho^* = \operatorname*{argmin}_{\rho} \lambda_{corr}$$

6.4.3 Alleviating Current Limitations in Correspondence

Recently, several ubiquitous shortcomings were identified in a review of over 50 LC-MS correspondence algorithms [97]. The most significant of these shortcomings was the fact that all current LC-MS correspondence algorithms make model assumptions that fail to capture common behavior. In other words, each algorithm is constructed in such a way that the algorithm is guaranteed to get the wrong answer under certain conditions that are common to real LC-MS data. The behaviors discussed included the ideas that:

- Not all analytes appear in all replicates.
- Elution order can swap.
- Shifts occur in m/z as well as in RT.

Some correspondence methods reduce isotopic envelopes to a single point representation. This deprives the method of a rich source of distinguishing data found in full isotopic envelopes—the expectation of contiguous isotope traces with a uniform m/z charge gap, number of isotope traces, and relative abundance ratio of isotope traces. Similarly, most correspondence algorithms conduct an initial RT alignment, where signals (almost always much-reduced from the full isotopic envelope, and rarely built up from isotope traces to isotopic envelopes) are shifted up or down in RT (preserving original order) in order to most closely match a reference run. This is invariably followed by direct matching. The problem is that the initial warping is a lossy procedure that adulterates the original RT time, which would be useful to probabilistically ascertaining the closest corresponding isotopic envelope.

The proposed objective function does not force matches between runs, as it is very common for species to either not be present or fall below the signal-to-noise ratio in differential studies. Instead, the proposed objective function leverages the full breadth of isotope envelope information, allowing a rigorous direct comparison of candidate correspondences based on all available data to select the most likely correspondence (in the sense of minimizing error), or no correspondence at all if that is the most likely case given the data.

6.5 Conclusion

We present a concise attempt to formalize LC-MS data clustering problems, describing the constructs of isotope traces and isotopic envelopes and their relational structure. We provide a review of current approaches to isotope trace extraction and LC-MS correspondence, and propose novel objective functions for both tasks that address shortcomings in current methods.

6.6 Acknowledgements

This work was supported by the National Science Foundation Graduate Research Fellowship [DGE-0750759] to R.S.

Chapter 7

Mspire-Simulator: LC-MS Shotgun Proteomic Simulator for Creating Realistic Gold Standard Data[1]

Abstract

The most important step in any quantitative proteomic pipeline is feature detection (aka peak picking). However, generating quality hand-annotated data sets to validate the algorithms, especially for lower abundance peaks, is nearly impossible. An alternative for creating gold standard data is to simulate it with features closely mimicking real data. We present Mspire-Simulator, a free, open source shotgun proteomic simulator that goes beyond previous simulation attempts by generating LC-MS features with realistic m/z and intensity variance along with other noise components. It also includes machine learned models for retention time and peak intensity prediction and a genetic algorithm to custom fit model parameters for experimental data sets. We show that these methods are applicable to data from three different mass spectrometers, including two fundamentally different types, and show visually and analytically that simulated peaks are nearly indistinguishable from actual data. Researchers can use simulated data to rigorously test quantitation software, and proteomic researchers may benefit from overlaying simulated data on actual data sets.

[1] Noyce, A.B., Smith, R., Dalgleish, J., Taylor, R.M., Erb, K.C., Okuda, N., and Prince, J.T.: **Mspire-Simulator: LC-MS shotgun proteomic simulator for creating realistic gold standard data**, *Journal of Proteome Research*, 12(12), pp. 5742-5749, 2013

7.1 Introduction

A single liquid chromatography-mass spectrometry (LC-MS) run is inherently capable of quantifying upwards of 100,000 peptides [64]. Unfortunately, in a typical analysis the majority of this data is discarded due to difficulties in identifying and accurately picking chromatographic peaks, especially those of lower abundance. Increasing the accuracy of peak picking results in the detection of more features that can be compared across runs. More accurate peak picking can also influence mass estimates and therefore yield an increase in the number and quality of identifications [22]. It ultimately simplifies cross-run comparisons of feature abundances and increases the overall accuracy of those quantitative comparisons. In other words, peak picking quality influences the entire downstream analysis.

For these reasons, it is undoubted that the most important step of a proteomic workflow is feature detection, for which many algorithms exist[14, 22, 75]. However, very little has been done to fully test or compare the performance of these algorithms. In large part, this is due to the challenging nature of creating gold-standard data. Fully annotating actual complex proteomic data sets, or even small portions, is extremely time consuming, difficult, error prone, and subjective. Because MS/MS annotation is rare for small peaks and because they have intensities near the signal to noise threshold, accurate human annotation of small peaks in a complex sample is very likely impossible.

Simulation is routinely used in related fields when gold standard data is difficult to come by (e.g., systems biology network simulation) [80] or the cost of performing each experiment is high (simulated ion movement in MS fields) [83]. For quantitative mass spectrometry, an attractive alternative to using hand-labeled data sets is to simulate actual data using noise parameters derived from experimental data. An ideal simulator would generate data sets where all aspects of the data are known, the various noise components are adjustable, and the peak characteristics conform to those found in biologically derived data sets. Such data sets would be invaluable for comparing algorithms because accuracy can be comprehensively and quickly ascertained programmatically. The speed of this feedback will also aid in the creation of new, more sophisticated algorithms.

Because simulators can produce fully defined peaks of any size, data sets produced by simulation are particularly well-suited to test algorithms for their ability to detect and accurately quantify small LC peaks. Small peaks are highly desirable targets for identification and quantitation because: 1) seminal biological events may occur at low quantities (e.g., upstream signal transduction) 2) a change in state to low quantity may be as significant as an increase in quantity 3) post-translational modifications may manifest themselves as a drop in the unmodified peptides concentration 4) lower abundance peaks constitute the majority of peaks in an LC-MS run and these are inaccessible by current MS/MS regimes. By quantifying low abundance peaks, proteomic and individual protein coverage may improve, and intra-protein variation can be tracked.

Many proteomic workflows allow users to examine their experimentally derived fragmentation spectra alongside a representation of the theoretically matched spectrum (i.e., an MS/MS fragmentation view). With a simulated LC/MS data set in hand, a somewhat analogous view could be generated for the user where simulated MS1 data is layered on top of actual data. This view would encourage a researcher to examine their MS1 output in order to reconcile what they can observe with what they expect to observe. A peak that went unidentified may still be present, and researchers would then know where to look within their MS1 data. Alternatively, a peak that should have been present may be absent prompting researchers to simulate data with conjectured post-translational modifications in an effort to locate the modified peak. Simulated data sets have the potential to augment the traditional proteomics workflow in which researchers often neglect to thoroughly examine their MS1 data.

While previous MS1 proteomic simulators [9, 87] have been created, a simulator that mimics the intensity and m/z variance found in real data sets is critical for testing peak-picking/quantitation algorithms. Here we present a full featured LC/MS shotgun proteomics simulator, Mspire-Simulator, that generates peptide peaks with realistic m/z and intensity variance and elution profiles. Machine learning is used to generate peaks with a realistic retention time distribution as well as peak heights reflecting peptide ionization efficiency.

7.2 Methods

Mspire-Simulator takes as its input FASTA files containing the protein sequences that are to be in the simulated run. Using one of 16 proteolytic enzymes and relevant digestion parameters each protein sequence is in-silico digested into peptides. Each peptide's charge, mass and theoretical spectrum, including the isotopic distribution, is calculated. These calculations are currently used to create centroided data. The simulator will be extended to create profile data in the future. Centroided data will be most useful initially because most analytical software deals with this type data. The simulator is implemented in the Ruby programing language and makes use of and extends the mspire (mass spectrometry proteomics in Ruby) library [76]. It is available under the MIT license and works out of the box with sensible defaults. Customization to data from different machines is achieved through an included Ruby script which uses a genetic curve fitting algorithm. This script produces SVG files that visualize the fits as well as the necessary parameters for Mspire-Simulator to adapt its simulations.

The actual data used to create our default simulation model was obtained from our in house LTQ-Orbitrap mass spectrometer coupled with reverse phase liquid chromatography using nanospray ionization. The data is derived from an LC-MS shotgun proteomic run of complex Human Embryonic Kidney (HEK-293T) cells. We used a Waters Nano Acuity column (15cm long). A solvent used was 95:5 water to acetonitrile and 0.1% formic acid and B solvent was acetonitrile and 0.1% formic acid. Gradient was formed by 5% - 60% solvent mix over 70% of the run. This data along with all files produced and used is deposited at https://chorusproject.org/anonymous/download/experiment/-17116340021687089. The

MM14 data is already available at http://msbi.ipb-halle.de/msbi/centwave/, and the Orbitrap-Velos data which is available upon request.

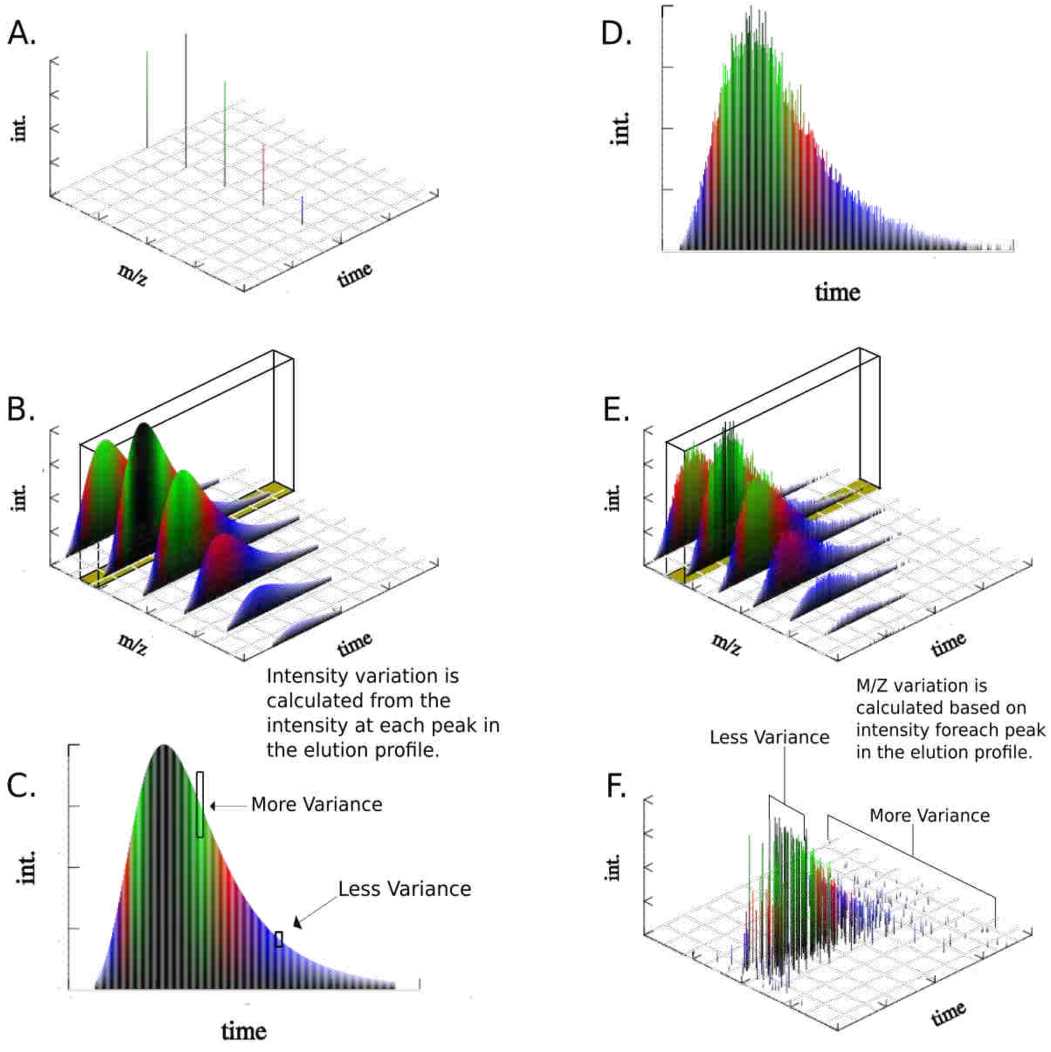
Orbitrap-Velos data was generously provided by the Christine Vogel lab and was from a ubiquitin pulldown from Saccharomyces Cerevisiae (Eksigent NanoFlow Plus, LC gradient 2-90% acetonitrile over 4.5 hrs at flow rate 400nL/min). MM14 data is from the Bruker MicrOTOF-Q instrument, is described by Tautenhahn et al. and details can be found in that publication [104].

7.3 Results

Mspire-Simulator models elution, variance in intensity and variance in the mass to charge ratio (m/z) and predicts retention times and intensities for peptides. We follow the convention of Cappadona et al. and refer to a peptide feature as the full chromatographic profile of a peptide (at a given charge state) and a peptide peak as an individual isotopic component of a feature [14]. Figure 7.1 outlines the overall process of simulation, and we consider each component in turn.

7.3.1 Retention Time and Intensity Prediction

A peptide is first situated along the retention time axis (see Figure 7.1A). Both retention time and intensity are predicted for each peptide using a machine learned model that was built in WEKA [39]. We used the M5Rules [48] algorithm for retention time prediction and the M5P [112] algorithm for intensity prediction, both of which gave the best correlation coefficients for our test data, 0.96 and 0.74 respectively using the internal WEKA ten-fold cross validation technique. The test data can be downloaded using the above hash. These prediction models were trained on in house data which contained amino acid counts, average m/z value, the charge state, mass, retention time, and a binned intensity value for 1484 peptides. The intensity values were binned into ten bins based on magnitude ranges. This allowed for a better prediction of intensities. The user may replace the default models with



Intensity variation is calculated from the intensity at each peak in the elution profile.

M/Z variation is calculated based on intensity foreach peak in the elution profile.

Figure 7.1: Overall process of simulation from theoretical spectrum to realistic peaks. The underscored 3d box in part B and E designates the specific peak shown in following parts. A: The theoretical spectrum calculated for a certain peptide. B: Ideal elution profiles are given to the spectrum. C, D: Intensity variance is calculated for each peak in the elution profile. E, F: Mass to charge variance is calculated for each peak in the elution profile.

custom/better ones in order to mimic other configurations and machines. For each peptide a single retention time and intensity is predicted; these values are used as starting points from which the retention times and intensities of all the centroids related to a particular peptide are generated. The retention times are coerced into times that conform to a user specified sampling rate (e.g. one scan per two seconds). Elution profiles are generated by sampling from the normal distributions of parameters t and f.

7.3.2 Feature Shape

Peptide features are modeled along the m/z axis (see Figure 7.1A) by predicting the charge states and isotope distribution of a peptide. For charge state prediction, a user specifies a pH, after which a standard iterative procedure is used to determine the ratio of charge states that would be observed (e.g. for the peptide DRVYIHPF at a pH of 2.0, 29.045% of this peptide would have a charge of +2 and 70.959% a charge of +3). We label this parameter ionized pH to indicate that it represents the acidity of the peptide as it enters the mass spectrometer and not necessarily in LC buffer. Isotope distributions are calculated by FFT convolution [78].

The elution profile (see Figure 7.1B) is produced by function composition of a dynamic standard deviation with a Gaussian function. The standard deviation (σ) is based on the relative position along the elution curve:

$$\sigma = tx + f \tag{7.1}$$

where x is the relative retention time index from the starting retention time of the feature, t is the tailing factor and f influences the shape at the front of the profile. The elution profile is then given by substitution of σ into a Gaussian function:

$$i = he^{-0.5(\frac{x-\mu}{\sigma})^2} \tag{7.2}$$

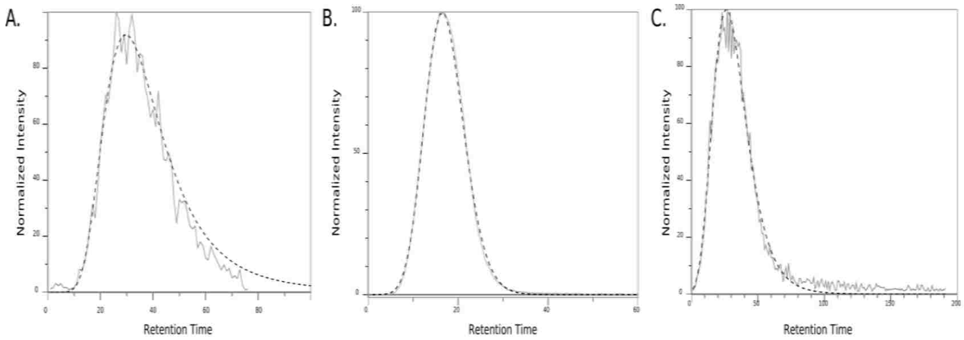


Figure 7.2: Elution Peak Shape. Max intensity normalization was used in each case. The noisy gray line shows a peak from actual data. Dashed line shows the function that the simulator uses. The simulators model can be modified to fit many elution profiles present in real data. A: LTQ-Orbitrap. B: Orbitrap-Velos. C: Qq-TOF.

where i is the intensity at that point in the elution, μ is where the apex of the curve is located, x is as above, and h is an initial height factor which determines the maximum height of the peak and is the same for each peptide. Thus, i is a generalized intensity which is later modified by a variance model and predicted intensity values as mentioned above. This produces a skewed elution profile that fits peaks derived from a wide variety of elution conditions as shown for data an LTQ-Orbitrap (see Figure 7.2A), a Quadrapole Time-of-Flight (Qq-TOF) (see Figure 7.1B) and an Orbitrap-Velos (see Figure 7.1C).

For the mass spectrometer types we examined there was a global relationship between the intensity of a peak and the variance of its measurement. We observed larger intensity variance in more intense features and thus also nearer the apex of an eluting peak (see Figure 7.1C,D). An inverse exponential function captures this relationship:

$$\sigma = m * (1 - e^{-c*i}) + d \tag{7.3}$$

where σ defines the standard deviation in intensity given the intensity value, i. The c, d, and m parameters represent experimentally derived constants that can be used to fine-tune the function for different mass spectrometers or run conditions (see Figure 7.3A,B). σ is then composited into a Gaussian function for each peak, again like above, and the ideal intensity is modified by drawing stochastically from this distribution. Our intensity variance model

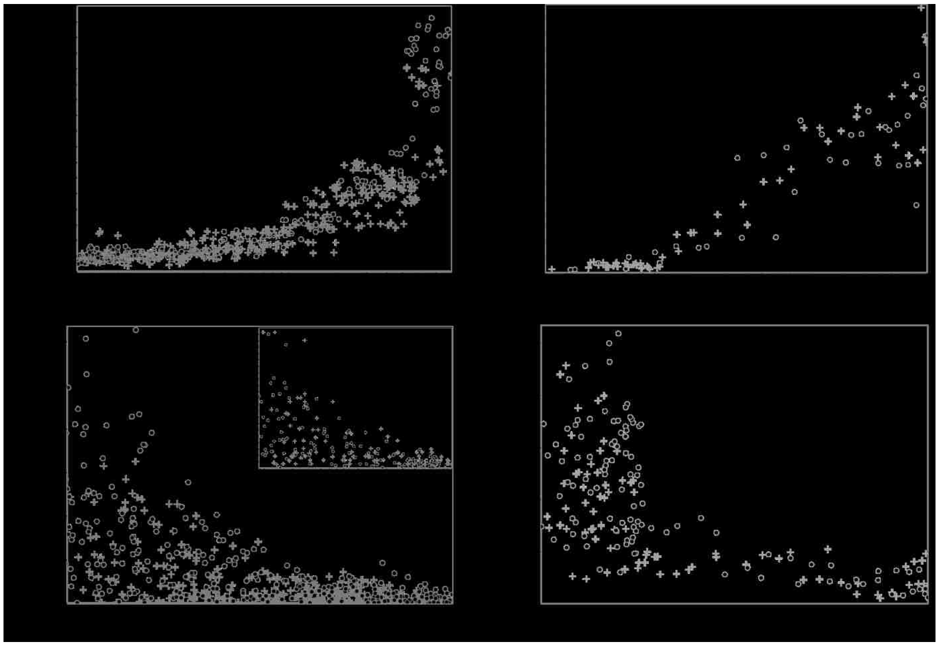


Figure 7.3: Intensity and m/z variance. Circles show simulated standard deviation variance and pluses show actual standard deviation variance. Max intensity normalization was applied in each case. The x axis is on a log10 scale. A-B: Standard deviation is calculated along intervals of ten peaks across the elution profiles. A: The simulator models this behavior from a LTQ-Orbitrap accurately with a small RMSD between actual and simulated of 0.9051. B: It can also model Qq-TOF data accurately; RMSD of 1.1167. C: Inset is one actual elution peak and the large image is four combined (RMSD = 0.1153). D: M/z variance from Qq-TOF data. C-D: This shows the general trend of measured m/z values varying more at low intensity signals and less at high intensity signals as well as the simulators ability to mimic this observation.

adequately mimics real data. When compared to actual data we observe a RMSD of 0.9051 (see Figure 7.3A,B).

The m/z variance is a function of intensity and therefore may vary between peptide features, and also within each elution profile contained in a feature (see Figure 7.1E,F). This is modeled by the following function:

$$\sigma = m * i^{-y} \tag{7.4}$$

Where σ is the standard deviation, i is the relative intensity of the feature at that point in its elution profile and y is another experimentally derived constant that can be fit to different data types. The standard deviation function is composited with a Gaussian function, similar to the above elution functions, and is then randomly sampled from to give the quantity of m/z variance in either direction. The m/z variance model produces realistic results based on the comparison of simulated m/z variance to actual m/z variance (see Figure 7.3C,D).

Feature shape is further modeled by given protein abundances. The abundances can be specified in the FASTA file header by a '#' symbol followed by a value representing the percentage of that protein in the sample. If no abundances are given, equal molarity is assumed. These values are then used to modify the total area under the function that determines feature shape by a simple scaling procedure.

7.3.3 Drops and Noise

At certain retention times, in real LC-MS runs, entire scans where very few if any peaks are observed are referred to as drops (e.g. PeptideAtlas accession PAe000142 contributed by S. Markey). Our model also simulates drops at random retention times by a specified percentage of the total run time. This elevates the realism in the simulation and adds another dimension of control when using simulated data to test analytical software.

The simulator has the ability to add white noise to the spectra based on density and intensity factors specified by the user. The higher the density factor, the more white noise there is in each spectrum. Intensities are pulled from a flat random distribution that varies between a maximum and minimum value given by the user. These parameters, along with the option to turn off the white noise completely, give the user complete control for testing purposes.

7.3.4 Merging Overlapping Peaks

As a final processing step, overlapping peaks are detected and merged. This is accomplished by using a ppm range to define whether two peaks are sufficiently close to be overlapping or not. The intensities of the peaks to be merged are summed and the new m/z value is calculated by a weighted average of the original m/z values weighted by the intensities of the respective peaks. We use 1/4 of the m/z variance in ppm to define the range that we use to detect overlapping peaks, and this parameter is adjustable by the user.

7.3.5 MS/MS

Theoretical fragmentation spectra are produced by generating fragment ion formulas for all possible cleavages and calculating the mass for each ion, at the predicted charge states. The ion types are configurable, and masses can be average or monoisotopic. The fragmentation spectra are produced by the MS-fragmenter gem, freely available from Rubygems.org.

7.3.6 Modifications

Mspire-Simulator has the ability to add modifications to specified residues and termini. Modifications are read in by the user specifying a modification ID from the PSI-MOD.obo and which residue/terminus to apply it too. These modifications are then used in the calculation of each spectrum. Since there will always be modifications in peptide samples this is an important part of simulation.

7.3.7 Output

The simulated run is written to an mzML file that can be visualized with any mzML file viewer. The mzML format is the standard de jure and is quickly becoming the standard de facto for mass spectrometer data. Cross-platform converters like Proteowizard [50] can convert mzML into mzXML [72] or other formats. Alternatively, the code base itself could easily be extended to directly output other representations of the data as well. The program also creates an SQLite, XML or CSV file which contains information on all of the data in the simulated run which can then be used to validate peak picking and quantitation software.

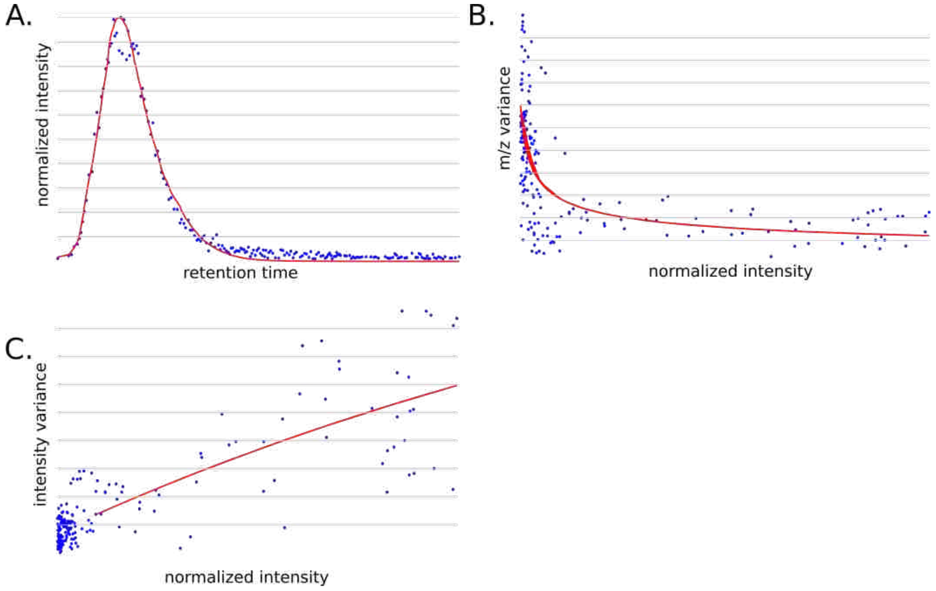


Figure 7.4: Demonstrating the visual output from the curve fitting program. Max intensity normalization was used for each. This is a fit of Orbitrap Velos data. The blue dots show the actual data and the red smooth lines represent the curve fit. This shows the ability to quickly generate parameters needed to simulate different types of data. This output took 5 min.

7.3.8 Parameter Fitting Automation

Simulating data from different mass-spectrometers and operating conditions requires some customization of noise and variance parameters. We developed a genetic algorithm to discover parameters from actual data. Figure 7.4 demonstrates the automatic fitting of Orbitrap Velos data, and it works equally well on the many peaks and instrument types we have tested.

7.3.9 Using the Simulator to Assess Quantitation Performance

The lack of quantitative comparison of data processing and wet lab protocol is due in large part to the daunting task of obtaining labeled data [96, 98]. The size and complexity of MS data sets precludes obtaining labeled data without a significant outlay of resources. Mspire-Simulator provides a facile method for generating any quantity of labeled simulated data. As a case study, consider Smith et al, where Mspire-Simulator data used in conjunction with hand labeled real data allowed the use of qualitative metrics to evaluate the accuracy of a peak summarization, a data processing step in non-chromatographic studies [92].

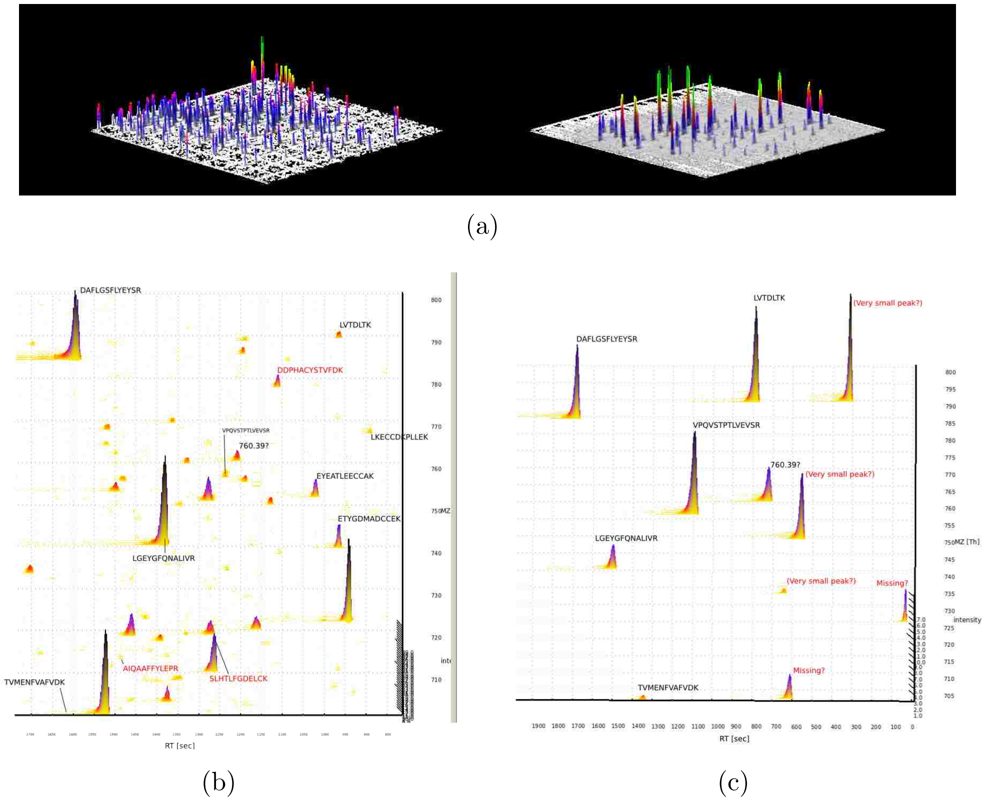
7.4 Discussion and Conclusion

Mspire-Simulator succeeds at creating highly realistic LC-MS peptide features as demonstrated by the comparison between actual and simulated data shown in Figure 7.6. Under macro- and microscopic inspection, analytically and visually, the two features are virtually indistinguishable (Table 1). An entire simulated run of Bovine Serum Albumin (BSA) Figure 7.5 shows the similarity between the simulated data and what is commonly observed in performing an actual BSA digest during quality control runs. Mspire-Simulator can also produce highly complex runs (see Figure 7.7) as well as simulate data from different mass-spectrometers (see Figures 7.2 and 7.3).

Table 7.1: Statistics comparing the two features shown in Figure 5. Normalized values were calculated by using a max intensity normalization. Isotope index (I.I.) from least to most abundant.

Statistic	Actual	Simulated	Difference
m/z Var I.I. 1	0.095	0.215	0.120
m/z Var I.I. 2	0.070	0.137	0.066 (ppm)
mz Var I.I. 3	0.239	0.373	0.134 (ppm)
m/z Var I.I. 4	0.255	0.203	0.043 (ppm)
m/z Var I.I. 5	0.032	0.296	0.264 (ppm)
Intensity Var	26.3	25.8	0.52
Elution Length (s)	43.384	43.360	0.025
Normalized Mean Intensity	21.90	21.35	0.55
Normalized Median Intensity	8.20	8.08	0.12
Num Samples / Num Peaks Used	73	85	12
Num Peaks in Quartile 1	21	21	0
Num Peaks in Quartile 2	27	25	2
Num Peak in Quartile 3	15	21	6
Num Peaks in Quartile 4	10	16	6

With Mspire-Simulators abilities, layering simulated data next to or on top of actual data, visually or analytically, could become standard practice in proteomicsmuch the way MS/MS spectra are layered onto predicted b and y ion series to identify potential database matches. By comparing actual data with the model and then refining the model, a feedback loop is created that has utility not only in affirming what is known but in pointing out what



(a)

(b)

(c)

Figure 7.5: A simulated BSA run ((a), left) is compared to an actual BSA run ((a), right). As can be seen, there are differences in retention times and intensities between the two runs indicating that refinements can be made to these two prediction models. A detail segment of the simulated ((b)) and real ((c)) BSA runs show that for each there are labels not found in the other (red) while there are many that are found in both (black).

is missing. Is an expected peptide missing because it has been modified? Are changes in the ratios of charge states indicative of pH or electrospray voltage aberrations? These and other aspects of a run can now be queried, and this process will inevitably result in more complete, more refined models of shotgun proteomics.

Refinements to Mspire-Simulator will focus initially on technical aspects of a LC-MS proteomic experiment. These include: a more explicit model of a peptides ionization efficiency [12, 63]; the pH of a solution as buffer concentrations change and as influenced by the electrospray process; the relative rates of tryptic digestion as a function of adjacent amino acid residues [79]; profile data simulation; exploring the relationship between variance

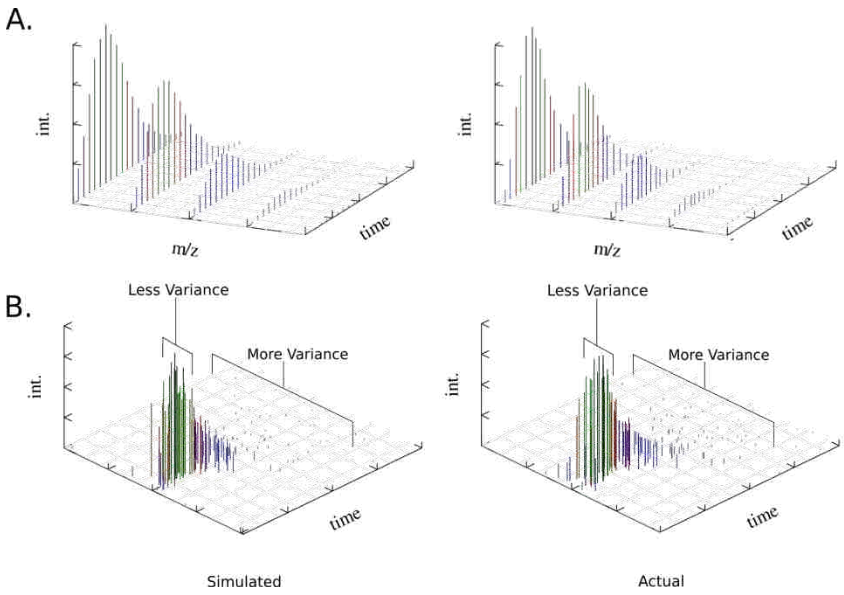


Figure 7.6: Left side shows simulated features and right side shows the actual features. A: Visual comparison of LC-MS feature from the peptide: HLVDEPQNLIK (single letter code amino acids). See Table 1 for analytical comparison. B: Detail of a single elution profile showing m/z variance characteristics. Simulated m/z variance is very similar to actual (see Table 1; row 1-5).

parameters and m/z and retention time; and improvements in peak merging. Future efforts will be devoted to these refinements.

Mspire-Simulator could also be extended with more sophisticated modeling of biological phenomenon. More rigorous post-translational modification or splice-variant prediction would alter the landscape of predicted peptides. Protein level enrichment could easily be added in, reflecting predictions about localization in a fractionated sample for instance. While biological questions are appropriately addressed after analysis of the raw data, it is nonetheless intriguing to consider mapping the biology as a simulated data set onto the raw data in an effort to generate putative identities for unanticipated peaks.

As simulated data becomes more sophisticated, we are aware of the possibility of its inappropriate use. The mzML file format is open and completely editable. As it currently stands, we see no way to prevent a simulated mzML file from being tampered with in order to be passed off as actual data. But, the problem is not as hopeless as it might seem at first glance: the mzML format encourages use of a file hash tag audit trail, so instrument produced

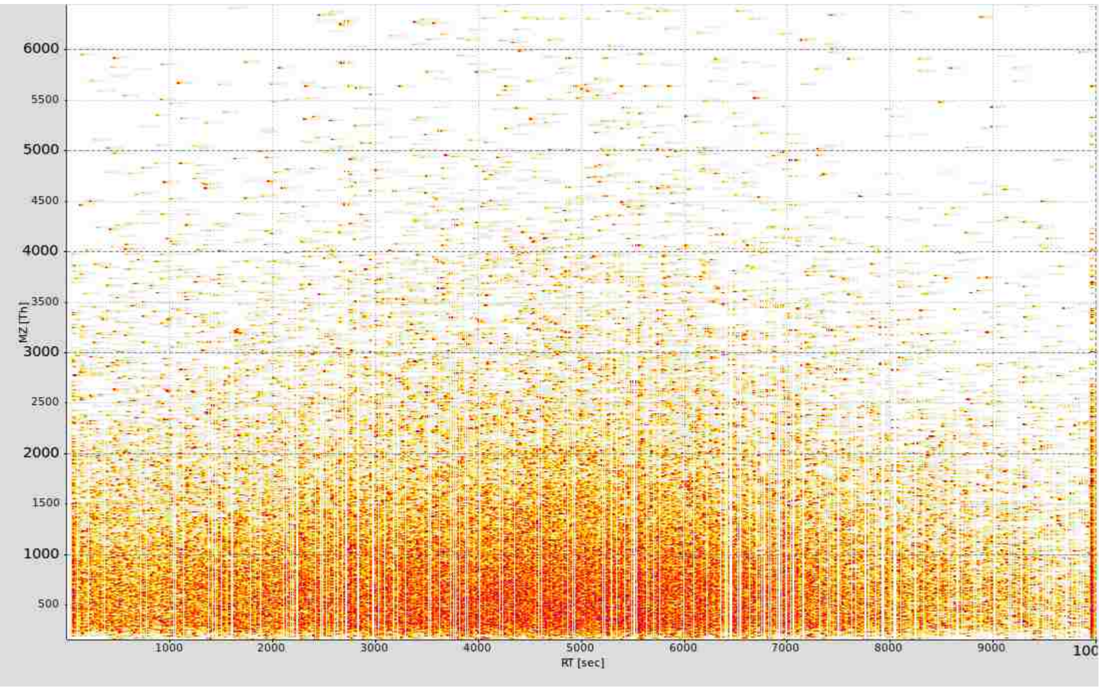


Figure 7.7: Birdseye view of a simulated complex human cell run. 50,000 peptides were taken from the human FASTA database and simulated in two charge states creating 100,000 features. The run was generated in 31hrs on a single 2.50GHz core and used 1.9Gb of RAM. White vertical lines represent dropped/lower signal and are intentionally included. The run demonstrates the simulators ability to generate highly complex runs. Purple peaks are the highest intensity, then red, yellow, and gray the lowest. Viewed in TOPPView [101].

data should always point back to a vendor produced raw data file. Deciding whether a file was simulated is now roughly equivalent to deciding whether a file was tampered with, and that is checking against a vendor produced raw data file. The potential for the fraudulent use of simulated data should serve, then, to encourage what researchers should be doing anyway: providing access to raw data and using audit trails. In any event, we suggest that the potential benefits of simulation software far outweigh the challenges presented by potential misuse.

Mspire-Simulator will be useful initially in testing and developing algorithms for peak picking and quantitation. Simulated data is not meant to replace testing on actual data, but to facilitate more rigorous testing of algorithms. Data may be simulated with a range of peak and noise characteristics, and strengths or flaws in algorithms uncovered. Mspire-Simulator will be especially useful in testing algorithms for their ability to accurately detect and quantify small peaks because the provenance of every centroid is known. Simulated data may ultimately facilitate workflows that find and quantitate an order of magnitude more peptides than is currently possible.

7.5 Acknowledgement

We would like to thank Brigham Young University for generous funding. We thank Christine Vogel for the Orbitrap-Velos data set, and we thank Adam Swensen for annotating BSA peaks for Figure 6. Institutional startup funds from Brigham Young University and multiple BYU Undergraduate Research Awards. RS is funded by NSF GRF [DGE-0750759].

Chapter 8

JAMSS: Proteomics Mass Spectrometry Simulation in Java[1]

Abstract

Countless proteomics data processing algorithms have been proposed, yet few have been critically evaluated due to lack of labeled data (data with known identities and quantities). Although labeling techniques exist, they are limited in terms of confidence and accuracy. *In silico* simulators have recently been used to create complex data with known identities and quantities. We propose JAMSS: a fast, self-contained *in silico* simulator capable of generating simulated MS and LC-MS runs. JAMSS improves upon previous *in silico* simulators in terms of its ease to install, minimal parameters, graphical user interface, multi-threading capability, retention time shift model, and reproducibility. The simulator is open source software licensed under the GPLv3. The software and source are available at https://github.com/optimusmoose/JAMSS.

[1] Smith, R. and Prince, J.T.: **JAMSS: Proteomics Mass Spectrometry Simulation in Java**, *Bioinformatics*, 2014 (in submission)

8.1 Introduction

Proteomics studies require the prediction of the quantity and identity of proteins in sample. The accuracy of the determination relies wholly on the accuracy of the data processing pipeline modules that systematically extract and process the components of the sample output file [93, 96]. Despite the criticality of data processing accuracy, very few published algorithms have quantitative comparisons against other algorithms using labeled data—data where the correct protein quantity and identity are known [98].

Common strategies for labeling data are limited in terms of confidence and accuracy. For example, MS/MS identifications are biased towards the approximately 16% most intense signals, and have an approximately 50% false positive rate [64], leading to evaluative results that are not representative of the data set, particularly among the more biologically significant but less-abundant peptides. Hand labeled data sets exist [21], but due to the complexity of labeling by hand usually consist of very small segments of data within an intensity threshold. Using existing tools, hand labeling consists of many subjective decisions, and creating a set of replicates would take years.

Construction of *in silico* data sets is an attractive alternative to labeling, as these data sets automatically include labels. *In silico* simulation consists of emulating the physiochemical processes involved in the mass spectrometry analysis of a sample in order to produce an mzML (or equivalent) output file similar to what would be generated in a real run but without any material or instrument time cost. Although more research is needed before an exact replicate of a real sample run can be simulated, the overall characteristics of the output data in terms of density, noise, signal shape, etc. are similar enough to be valuable as labeled data for LC-MS data processing algorithmic evaluation.

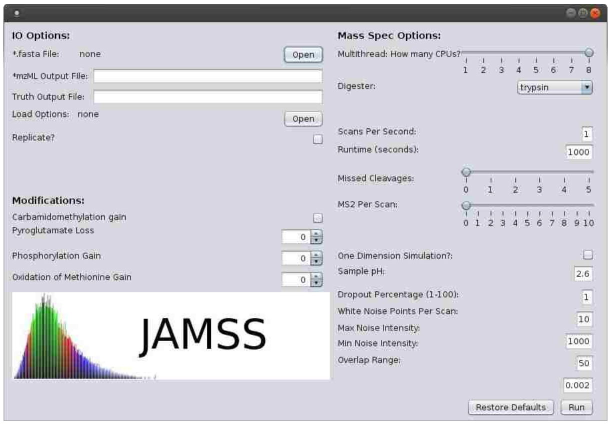
LC-MS simulation is still in its infancy. LC-MSsim, an incorporated module of OpenMS, was the first simulator to produce full-featured MS simulated data [87]. It has since been replaced by MSSimulator, featuring more realistic isotope trace variance (in both intensity and m/z) and MS/MS simulation [9]. Most recently, Mspire-simulator, a stand-alone simulator in the Ruby programming language, provided automatic charge modeling, realistic hourglass-shaped isotope traces (increased variance at lower intensities), direct control over post-translational modifications (PTMs), and the ability to extract simulation parameters from existing mzML files using machine learning [70].

There remains much to be done in MS simulation. Existing simulators have involved installation processes, requiring installation of their parent libraries and a sometimes onerous degree of dependency management. They are also both command-line programs with very little documentation. Neither program is multi-threaded. Both programs feature many parameters, some of which significantly alter the simulation outcome in unclear ways. Although Mspire-simulator produces run-to-run variation (unlike MSSimulator), it does not vary the RT of eluents across runs and cannot produce a clone of a previous run when fed the same input and parameters. Although Mspire-simulator's isotope trace generation features more realistic variance in m/z and intensity than MSSimulator, it is many times slower and has no bound on RAM requirements. Both programs seem limited in regards to PTMs: both seem to render all PTMs as static, even variable ones. MSSimulator produces the same isotope trace shape (scaled for intensity) for every peptide, meaning its utility for generating data sets for evaluating data processing algorithms is limited.

This paper describes the Java Mass Spectrometry Simulator (JAMSS), a novel simulator designed to address each of the above-mentioned drawbacks of current simulation software.

8.2 Methods

JAMSS takes any protein .fasta file as input. Optionally, users can specify the quantity of each protein as a percentage of the total sample content (see program documentation in README file). The GUI provides several clear options to modify the output. For example, the user can specify how many cores to use if using a multi-core machine. They can select one of 16 digestion enzymes. They can select how many scans per second, how many missed



IO Options:

*.fasta File: none

*mzML Output File:

Truth Output File:

Load Options: none

Replicate? □

Modifications:

Carbamidomethylation gain □

Pyroglutamate Loss 0

Phosphorylation Gain 0

Oxidation of Methionine Gain 0

Mass Spec Options:

Multithread: How many CPUs? 8

Digester: trypsin

Scans Per Second: 1

Runtime (seconds): 1000

Missed Cleavages: 0

MS2 Per Scan: 0

One Dimension Simulation?: □

Sample pH: 2.6

Dropout Percentage (1-100): 1

White Noise Points Per Scan: 10

Max Noise Intensity: 1000

Min Noise Intensity: 50

Overlap Range: 0.002

Figure 8.1: JAMSS has a straightforward GUI interface to facilitate parameter selection for MS simulation.

cleavages to allow, how many MS2s per scan to generate, how many noise points to include and at what intensity range, and the pH of the sample. They can control the resolution of the simulation through a merge parameter. Additionally, the GUI can be set for a one-dimensional (non-chromatographic) simulation, which is useful in modeling direct injection experiments. There are also settings for PTMs. The program includes options for carbamidomethylation, pyroglutamation, phosphorylation, and methionine oxidation. Although these options do not include all possible PTMs, limiting them by explicit mention allows for treating variable PTMs as they should be treated: that is, the combinatoric possibilities of all selected PTMs are calculated, and the total quantity of each protein is split according to the percentage of the proteins each PTM combination will affect.

After reading the .fasta file, the simulator instantiates N mass spectrometer objects, one for each CPU core selected by the user. The program delegates each protein sequence to one of the MS objects until all proteins are processed. Inside the MS object, each protein sequence is digested. For each peptide, atom counts are calculated from which the isotopic envelopes and charge are calculated. If PTMs are selected, this process is executed for each PTM applicable to the peptide. From there, the amino acid profile of the peptide (or PTM-modified peptide, if applicable) is fed into the same machine learning model used in [70] to predict the retention time of the peptide (see [70] for details). The intensity of the peptide is determined by the user-provided intensity or, if none is provided, an inverse exponential sample. The shape and variance of each isotope trace in a molecular envelope's isotopic envelopes is modeled using the same mechanisms in [70]. Isotope trace shapes are determined via a modified Gaussian function with sufficient variation so that no two isotope traces are identical, providing variation for replicate runs. Further variation is achieved by modeling RT shifts as normally distributed events. From there, each centroid is subject to general noise in m/z, as well as intensity-specific noise in m/z (providing splayed isotope traces in the head/tail regions).

Memory usage is bounded by having the MS objects periodically writing their produced centroids onto disk. Frequency of writing out is determined automatically as a factor of the size of the JVM selected by the user and the CPUs in use by the user. For faster processing, more RAM can be selected by the user at runtime. After all centroids are created, the program finishes each RT scan by merging points within the resolution set by the user and generating an mzML output file, as well as .csv files to facilitate labeling of each centroid, isotope trace, and isotopic envelope.

8.3 Results

JAMSS is a GUI-based MS simulator in Java (see Figure 8.1). It creates fully annotated complex proteomic data sets in both mzML and a convenient .csv format. It can be used to generate LC-MS and MS data, allowing for the evaluation of a wide range of data processing algorithms such as isotope trace extraction (both in chromatographic and non-chromatographic [92] applications), isotopic envelope extraction, molecular envelope extraction and reduction, and correspondence [97]. It has a limited number of intuitive parameters, a self-contained one click installation with no external libraries or dependencies, and supports multi-threading. It creates isotope traces with realistic variance in both m/z and intensity and has an user set memory upper bound. JAMSS handles variable PTMs and static PTMs. JAMSS features controlled randomized trace shape generators to create run-to-run variation in replicates but uses controlled random seeding so it is possible to produce a clone of a previous run. It maintains relative protein abundance in isotope traces accounting for isotope trace variability and abundance distribution over PTMs. JAMSS also models RT shifts in order to provide more realistic replicates for generating data sets to test LC-MS correspondence algorithms.

Funding: This work was supported by the National Science Foundation Graduate Research Fellowship [DGE-0750759] to R.S.

Chapter 9

Massifquant: Open-Source Kalman Filter Based XC-MS Isotope Trace Feature Detection[1]

Abstract

Motivation: Isotope trace detection is a fundamental step for XC-MS data-analysis that faces a multitude of technical challenges on complex samples. The Kalman filter application to isotope trace detection addresses some of these challenges; it discriminates closely eluting isotope traces in the m/z dimension, flexibly handles heteroscedastic m/z variances and does not bin the m/z axis. Yet the behavior of this Kalman filter application has not been fully characterized since no cost-free open-source implementation exists and incomplete evaluation standards for isotope trace detection persist. **Results:** Massifquant is an open source solution for Kalman filter isotope trace detection that has been subjected to novel and rigorous methods of performance evaluation. The presented evaluation with accompanying annotations and optimization guide sets a new standard for comparative isotope trace detection. Compared to centWave and matchedFilter—two alternative isotope trace detection engines in the XCMS software—Massifquant detected more true isotope traces in a real LC-MS complex sample, especially low-intensity isotope traces. It also offers competitive specificity and equally effective quantitation accuracy. **Availability:** Massifquant is integrated into *XCMS* with GPL license ≥ 2.0 and hosted by Bioconductor: `http://bioconductor.org` Annotation data is archived at `http://hdl.lib.byu.edu/1877/3232`. Parameter optimization code and documentation is hosted at `https://github.com/topherconley/optimize-it`.

[1] Conley, C., Smith, R., Torgrip, R.J.O., Taylor, R.M., Tautenhann R., and Prince, J.T.: **Massifquant: open-source Kalman filter based XC-MS isotope trace feature detection**, *Bioinformatics*, 2014

9.1 Introduction

The most important automated data-analysis step in a typical quantitative -omics XC-MS analysis pipeline is isotope trace (IT) detection [14][2]. In liquid or gas chromatography mass spectrometry (LC-MS or GC-MS, with either specified as XC-MS) analytes elute with chromatographic separation and are subsequently measured by the mass spectrometer. IT detection is the first and essential step in enumerating the signals of these analytes.

IT detection is a trivial task when performed on data derived from simple mixtures, but can be highly challenging for complex mixtures because there are 1) large numbers of analytes which co-elute, many show interlocking or overlapping isotope envelopes; 2) an unknown number of analytes; 3) an abundance of ITs with low signal to noise ratio; 4) significant intensity variation in the signal composing lower abundance ITs due to dynamic range limitations of the spectrometer; and 5) heteroscedastic m/z variance as a function of intensity for most mass spectrometers. Unisotropic m/z variance results in that the data comprising the tails of a IT have larger m/z variance than the data around the mode, and that low abundance ITs have a larger m/z variance than high abundance ITs.

Though difficult to achieve, increasing the sensitivity and accuracy of IT detection software influences the entire downstream analytical pipeline [96]. An example: Vast numbers of peptides go unidentified in proteomic analyses [64]; a more sensitive IT detection would allow researchers to track and quantify these peptides, leveraging identifications acquired in other samples. It goes without saying that accurately determining IT boundaries and distinguishing signal from noise improves quantitation results. Furthermore, accuracy in IT detection can also result in more accurate precursor mass estimates and therefore yield an increase in both the number and quality of peptide identifications.

Most IT detection software, such as matchedFilter, relies on the creation of fixed width m/z bins (buckets) to facilitate finding and quantifying eluting analytes. Though bucketing is computationally efficient, for complex data sets it is impossible to find a bin size and

[2] We adopt the MS-omics terminology specified in [95]

position that excludes closely co-eluting ITs while also being broad enough to fully capture the IT of interest. To address this shortcoming, Tautenhahn *et al.* [104] developed a software package, centWave, which uses a binless pre-scan to first identify regions of interest composed of centroids. A **centroid** is a (m/z, intensity) measurement pair at a given time scan of the chromatographic dimension. Once a region is specified, the centroids are then collapsed into a one-dimensional chromatogram and wavelet-based curve fitting is performed to separate closely eluting ITs. The approach is appealing because the initial algorithm identifies zones of interest in a binless way and because the algorithm used for detecting ITs using intensity fluctuation in the time domain is sophisticated. However, in this approach subtle shifts in m/z value are ignored when data are combined into a one-dimensional chromatogram. ITs which are very close in m/z or with poor chromatographic profiles may not be properly resolved.

The same year Aberg *et al.* [1] developed TracMass, a binless IT detection algorithm which fully utilizes m/z information in distinguishing ITs. TracMass uses a chromatographically traversing 2-dimensional Kalman filter model (KF)—one dimension focused on m/z values and the other on intensity values—to determine which centroids belong with each extending IT. The decision to incorporate a centroid is made by carefully weighing all previous m/z and intensity evidence of that IT, so mis-incorporation of centroids is rare as the KFs incorporate more data. Furthermore, the KF accounts for the heteroscedastic variance within the same IT as intensity values change. The KF approach can disentangle even the most closely eluting chromatographic ITs. Furthermore, for the non-expert user, TracMass requires few user parameters for effective operation.

Despite its apparent promise for IT detection in complex samples, no peer-reviewed publication had compared TracMass performance to leading options [128, 131] until just recently with TracMass2 [106]. This is not an isolated deficiency—most IT detection algorithms are not rigorously evaluated because of the difficulty of establishing ground-truth data, especially for lower abundance ITs [98, 131]. Indeed, other compelling binless methods for quantitation may benefit from a similar evaluation as presented here [22, 127].

Here, we make available an open source implementation of the TracMass algorithm, called Massifquant, and integrate it into the popular XCMS software suite [90, 104]. Like TracMass, Massifquant uses a two-dimensional Kalman filter to quickly, accurately, and adaptively find ITs in highly complex samples without resorting to binning, and its open license (GPL $\geq$ 2.0) enables further extension and inspection. We indicate how the KF adapts to m/z variance and describe two major modifications which mitigate known limitations of TracMass. We detail novel metrics for evaluating XC-MS IT detection and use these metrics with manually annotated data to perform a detailed evaluation of Massifquant, centWave, and matchedFilter performance on different LC-MS platforms.

9.2 Methods

9.2.1 Description of the Massifquant algorithm

Massifquant relies on 2D Kalman filters to identify ITs in XC-MS data. A single KF's purpose is to track the m/z and intensity coordinates of a IT over the chromatographic dimension. A **track** is an instance of a KF model, which predicts the existence of a centroid in the next time scan. If the prediction is close enough to a real centroid, it incorporates the real centroid to the track. Closeness is determined by quasi-confidence intervals centered about the prediction. The KF then updates its estimate of the underlying "true" centroid and predicts again. When the signal of the IT disappears (i.e., we have reached the end of a chromatographic IT) the KF will fail to predict a centroid on successive scans and tracking will be terminated.

With many ITs to be discovered, Massifquant manages a host of active KFs. For a given scan, each active KF claims the centroid that best fits its predicted location. Unclaimed centroids trigger new instances of KF tracks in the expectation that these are the beginning of new ITs. The process is then repeated on the next scan until all scans have been examined.

In this way, every centroid is either claimed by an existing KF or triggers the creation of a new KF. After an entire sample has been parsed, spurious KFs are discarded based on simple filters for minimum length, intensity, expected m/z deviance, or consecutive missed predictions.

We will describe the **Kalman gain** to highlight the model's adaptive nature and how it can be tuned. After the KF predicts a centroid, it refines the prediction by carefully weighting the model prediction error through a modeling device known as the Kalman Gain. This device is largely a function of (i) the estimation error covariance, which is initialized by the modeler, but evolves over time based on prediction performance; (ii) and the assumed measurement error of the Mass Spectrometer, also defined by the user. So the modeler may tune the Kalman gain based on these parameters. A smaller Kalman gain means that the model prediction, which is based on previous observations, is trusted to be closer to the true centroid location than the newly acquired observation. The default settings of Massifquant create a Kalman gain that places more trust in early acquired observations (i.e. the first 4-30 scans) as illustrated in Figure S1 in the supplementary materials. The idea is to find the IT's location quickly and not deviate once it has been found; the default works for a variety of situations, but can also be tuned to a particular dataset. The fact that the KF continuously adapts its centroid prediction estimates based on the information it has previously amassed and the variance it encounters makes it an effective tool for identifying ITs with their own specific heteroscedastic variance. For a more mathematical discussion, an introduction to the theory behind the discrete Kalman Filter/Gain are described in Welch and Bishop [117] and section 2 of the supplementary materials.

Massifquant implements most of the core of the TracMass algorithm; however, it is difficult to determine how much the two algorithms differ since the latter's source code is not available. There are a few known major differences. The initialization of the P is likely different. Moreover, the intensity component of the Scheffe-type quasi-confidence intervals—used to classify whether a next-scan centroid belongs to a KF prediction–was not found to be sufficiently discriminatory. Massifquant only uses the m/z dimension to determine a successful prediction. Retaining the intensity estimation in the KF does seem to aid in resolving competing KFs that claim the same centroid (by virtue of comparing their two dimensional prediction distances).

Massifquant also implements a function to ensure continuity of identified ITs that is not found in TracMass (discussed in section 3 of the supplementary information). We found that a KF will periodically lose the position of the IT, stop tracking it en route, triggering a new KF track which will finish estimating the IT's other data points (see supp. file Figure S2). Since each KF track corresponds to an IT, we call the undesirable phenomenon "segmentation". The segmentation problem was addressed by an ad-hoc t-test comparing the m/z locations between these problematic KF. The conservative test combines many of the segmented tracks into a unified IT.

A more thorough description of the Massifquant implementation is given in the supplementary material (see the section "Reimplementing the Kalman filter model"). The supplement highlights some differences with TracMass and a discussion of the logic behind specific design decisions. The description will be useful to anyone seeking to modify or extend the algorithm. Massifquant was written in C++ and has been integrated into the XCMS pipeline available through Bioconductor [40, 90]. It plays the same role as centWave or matchedFilter in the differential analysis workflow.

9.2.2 Annotation

Data sets

We chose two very different LC-MS data sets to assess IT-detection flexibility. The first annotated data set, MM14, is a subset from a UPLC-ESI-QTOF analysis of 14 plant metabolites resulting in 46 annotated ITs. The centWave developers originally showcased their method of parameter optimization on the entire data set, and its provenance is detailed in Tautenhahn *et al.* [104].

The second data set, MOUSE, is one fraction from a larger mouse brain phosphoproteomic analysis. Briefly, 408.8 mg of brain tissue was homogenized/boiled in SDS-lysis buffer and clarified. Proteins were then digested and peptides purified following the FASP protocol [124] to yield an estimated 7.3 mg of peptides. 25 mg of Titanspere TiO2 beads (GL Sciences) were used to enrich for phosphorylated peptides. 3M Empore Anion Exchange disks were packed into a 200 l pipette and Britton & Robinson buffer was used to elute at pH 11 (the fraction termed 'MOUSE' in this work), 6, 5, 4, and 2. MS analysis was perfomed with an LTQ-Orbitrap XL fed by an Eksigent NanoLC UHPLC system. A Nano Acquity (1.7m, 130 C18 bead BEH, 75m m x 150mm) column run at 375 nL/min in a linear gradient from 2.5% to 10% ACN (with water and 0.1% formic acid as the second buffer) for 60 minutes, then to 28% ACN for an additional 220 minutes. The complete raw file is available upon request, and virtually all parameters may be accessed using the cross-platform unfinnigan software (see `https://code.google.com/p/unfinnigan/`). The relevant parameters are: MS1 data collected between 375–1800 m/z at 60,000 resolution with an MS/MS data dependent scan collected after each MS scan. The section chosen for hand-annotation generally spans retention time 5429.5–7306.2 seconds and 600.0003–637.3923 m/z. In total, this area contained 589 annotated ITs which show variation in length, shape, and variance.

Data annotation

The MOUSE and MM14 datasets were manually-annotated to be used as ground truth for assessing the automated IT detection abilities. A tuned LC-nanoESI system is capable of producing consistent chromatographic IT shapes. However, when running complex samples, even on the best tuned system, fundamental dynamic range limitations will unavoidably produce IT shapes that are far from ideal. The lack of characteristic IT shapes among lower abundance ITs, the number of overlapping ITs (in m/z and time), and their sheer number and density makes manual annotation difficult. For the MOUSE data, any IT that did not exceed a maximum intensity of 1×10^5 was ignored to preserve the integrity of the annotation.

Because IT annotation in complex data sets is challenging, we established guidelines for what is called a true IT. These guidelines consider within-IT and between-IT characteristics to ensure the best annotation possible. To be defined as a IT, a series of centroids should typically exhibit the following properties:

Within

1. The m/z error variance structure is influenced by intensity. Toward the tails of a IT, the m/z observations show mostly symmetric and increasing deviations from the mean. The body and apex centroids deviate less. From a bird's eye view (i.e., looking down the intensity axis), the m/z-time projection has the shape of a string fraying at the edges.
2. The collective centroids should have a chromatographic IT shape. Dramatic oscillations in intensity from scan to scan could disqualify an annotation.

Between

1. The detected ITs should have approximately the same m/z ppm variance.
2. Within an isotopic envelope, ITs should have very similar mode and shape, although length typically varies.

In each case, great effort was made to balance the benefits of the systematic application of these rules with human judgment. Each IT was individually annotated (based on all criteria) and then wrapped into appropriate isotopic distributions where possible.

We executed this annotation scheme on the MM14 and MOUSE data sets using ToppView [101] as follows: From a global 2-D view, the annotator identified mass traces satisfying mentioned properties. After zooming, a 3-D inspection confirmed similar chromatographic length and shape for a given isotopic distribution. Once confirmed, the IT's centroids were selected and collectively saved into an .mzML file. Candidate mass traces that did not sufficiently satisfy all the criteria, but still had some resemblance to a IT, were labeled as questionable and saved as .mzML files; these were excluded from the algorithm performance analysis since they were deemed liable to interfere with true algorithmic specificity and sensitivity. Objectively determining an IT's chromatographic boundaries is difficult, especially since there is so much diversity among IT shape and length. Generally, we tried to include as much of each IT tail as possible and to be as consistent as possible across each data set.

9.2.3 Performance Evaluation

Different algorithms select different portions of a IT when attempting to identify ITs (any attempted IT classification we call a **candidate**). Because the extent and location of the mapping from a candidate to the true IT may vary widely, gauging the success of a candidate can be challenging. For example, a method that identifies 30 centroids directly in the middle of the high intensity region of a IT should be given more credit than one that identifies 35 centroids but that are all in the very low intensity tail region. In another example, credit should be given to an algorithm that successfully captures an entire IT with three distinct candidates, but it should not receive as much credit as an algorithm that identified the IT with a single candidate. These examples motivated the development of two ways of examining success: at the IT-level and at the entire sample-level.

Isotope trace-level evaluation

Classifying the success of an algorithm at the IT-level requires the classification to be general enough to handle a variety of IT shapes and yet still be precise. To classify the successful identification of a IT, we defined metrics that consider how a candidate's centroids individually contribute to the overall intensity of the annotated IT, namely, the true area under the curve (AUC_A). The centroids clustered into a candidate are either true positives, false positives, or false negatives. Restricting attention to the true positives, a candidate's true area under the curve is denoted as AUC_{TP}. Naturally, a candidate's relative correct identification of a IT within the context of intensity is defined to be $\alpha := \frac{\mathrm{AUC}_{TP}}{\mathrm{AUC}_A}$. Now, an algorithm is said to sufficiently identify the i^{th} annotated IT if $\alpha_i \geq 1 - r$, where $0 \leq r \leq 1$. For the following analysis, we took $r = 0.5$ because requiring a candidate to capture more than 50% of an IT's total intensity ensures that the main body of a IT has been identified, while still allowing for differences in opinion on exact IT boundaries. In short, this criterion abstracts away the difficulty of varying shapes and algorithmic-selection bias.

Conversely, the false positive and false negative centroids contain precise information as to where a candidate is accurate and by how much. To be clear, the AUC quantitation error is taking evaluation precision beyond classification. Let AUC* be the quantification reported by the algorithm, which includes true and false positive centroids alike and excludes false negative centroids. Then the AUC percent error is simply $\epsilon := \frac{|\mathrm{AUC}_A - \mathrm{AUC}^*|}{\mathrm{AUC}_A} \times 100\%$. Dramatic variation in IT intensity motivated the percent error representation.

Another issue is that true negative ITs are impossible to define. So an algorithm's IT-identification accuracy was measured by the commonly used metrics of precision and recall (sensitivity) for information retrieval. **Isotope trace sensitivity** (s_f) is the number of ITs correctly identified by the algorithm *divided by* the number of true ITs. **Isotope trace precision** (p_f) is the number of ITs correctly identified by the algorithm *divided by* the number of algorithm-claimed ITs. High sensitivity means the algorithm successfully identifies most true ITs, while high precision is a measure of identification reliability. The harmonic mean of these is the F_1 score $:= 2\frac{s_f p_f}{s_f + p_f}$; it summarizes the overall identification performance.

Sample-level evaluation

Finally, sample-level metrics allow us to define how much of the entire sample AUC was correctly identified without regard for individual ITs. It is a way to quantify the level of intensity information found by an IT detection without regard to how the centroids are actually clustered into ITs. The **sample sensitivity** is defined as $\frac{\sum_i \mathrm{AUC}_{TP_i}}{\sum_j \mathrm{AUC}_{A_j}}$. This is the total algorithm-identified true raw intensity *divided by* total true raw intensity. On this global level, a true negative can be defined as the sample noise, or the centroids that don't contribute to any real ITs. Thus, the **sample specificity** equals $\frac{\sum_i \mathrm{AUC}_{TN_i}}{\sum_j \mathrm{AUC}_{FP_j} + \sum_k \mathrm{AUC}_{TN_k}}$. This taken to be the total correctly algorithm-ignored raw intensity (true negative signal) *divided by* total noise raw intensity of the sample (including false positives of the algorithm) . These last two metrics are useful as a global measure of accuracy in contrast to the IT-specific accuracy in the preceding metrics.

Evaluation by IT type

An evaluation should indicate how certain IT types influence performance. Simpson's paradox further motivates an evaluation by type since conclusions based on the aggregate annotation are sometimes reversed when analyzed by type [8]. We classified ITs by intensity, ppm error, and length. Annotated ITs were grouped by the variable of interest into 8 percentile categories $\{[0, 12.5\%), [12.5\%, 25\%) \ldots, [87.5\%, 100\%]\}$. For example, the longest IT was categorized in [87.5%, 100%]. The recall was computed for each category; precision was approximate because mapping the algorithm-identified ITs to the right annotation-based category was not always right. For instance an algorithm-identified IT length might be shorter or longer than the annotation length and the mapping can only be corrected if the IT identification is correct.

Optimization

With the goal of maximizing the F_1-score, we optimized parameters for the two algorithms on each dataset. Initial values for centWave on MM14 were selected from the paper Tautenhahn *et al.* [104]; the manual annotations provided a baseline of minimum IT length, height, and ppm deviation. Where prior knowledge was absent, liberal parameter grids were explored for parameters like *snthresh* for centWave, or *criticalValue* for Massifquant. Paired parameters, or parameters that were thought to have interactions, were explored simultaneously in two dimensions. For instance, the (min, max) IT length form a natural pair and exhibited interactions in F-score performance for centWave. The most important parameters for both algorithms, (*snthresh*, *ppm*) in centWave, and (*criticalValue*, *consecMissedLim*) in

Table 9.1: centWave optimization on MM14 improved with identification performance and the parameters are in the same vicinity.

version	ppm	snthresh	peakwidth	peakfilter	F_1-score
original	30	2	(5,10)	(2, 400)	.8936
our evaluation	18.4	2.5	(3,11)	(1, 511)	.9438

Massifquant were searched simultaneously. Their respective F-score surface plots exhibited near-concavity, a desirable property for parameter tuning. It appears unique to Massifquant that all F-score surface plots had near-concavity. The optimizations were conducted with R (`http://www.r-project.org`) and Matlab scripts (MATLAB version 7.14.0.739, The Mathworks Inc., Natick, Massachusetts). Scripts and detailed procedures to reproduce all results are provided upon request. Other details of the optimization are included in the supplementary file. Table 9.1 compares centWave performance on MM14 based on reported optimized parameters from the original centWave publication and the optimized parameters resulting from this new evaluation. The two different evaluation settings yield similar parameters and F_1-scores, suggesting this new annotation and evaluation effort is valid. For matchedFilter, all combinations of the suggested ranges for each parameter were exhaustively evaluated.

9.3 Results

9.3.1 Overall Evaluation

As detailed in the methods section, we developed an independent, open-source implementation of Aberg et al.'s TracMass algorithm, and call it 'Massifquant'. The algorithm uses 2-dimensional Kalman filters to adaptively find chromatographic ITs in the m/z domain without bucketing the data. We compared Massifquant's ability to sensitively and accurately find ITs with centWave, a sophisticated and well-known algorithm used in the XCMS platform for label-free IT detection, and matchedFilter, the original binning-based XCMS method for IT detection.

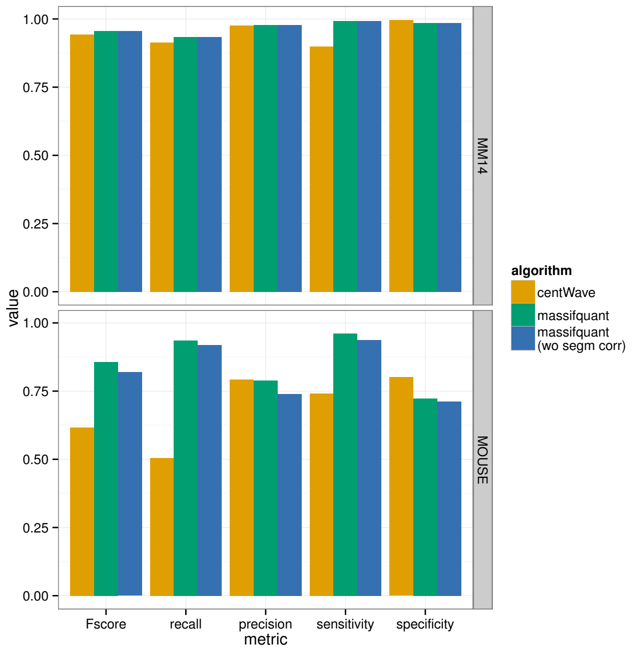


Figure 9.1: Optimized performance metrics by dataset and algorithm. Massifquant is the performance of Massifquant without correcting IT segmentation. This and other figures used reshape2 and ggplot2 R packages [120, 121]

We manually annotated ITs in two data sets, chosen to have different characteristics, following a set of rational guidelines. The MM14 data set is a run of 14 plant metabolites on a lower-resolution UPLC-ESI-QTOF. The MM14 reveals the performance of an IT finder under close to ideal circumstances (viz. low sample complexity, good signal-to-noise, good chromatography). The MOUSE sample was run on an Orbitrap mass spectrometer and is typical of many highly complex proteomic analyses. While chromatographic IT shapes are smooth for high abundance ITs, the intrinsic dynamic range limitations result in greater m/z and intensity variability for lower abundance analytes. The heterogeneity of IT sizes and shapes encountered in the MOUSE data is ideal for discovering the limitations of an IT detection algorithm.

Figure 9.1 shows that Massifquant reported uniformly higher sensitivity values than centWave and the t-test union of segmented ITs improves Massifquant performance on MOUSE. As for identification reliability, precision was in the same neighborhood for both

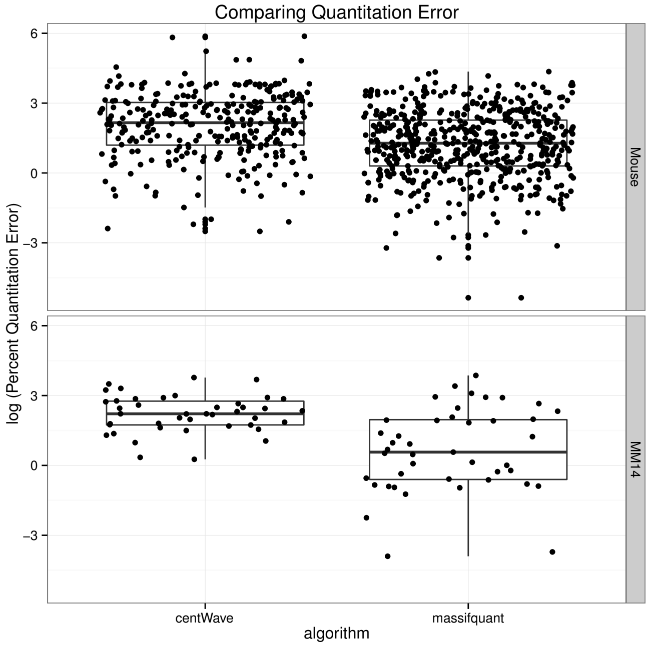


Figure 9.2: A comparison of log-transformed percent quantitation errors (log ϵ) for successfully identified ITs. Massifquant outperforms centWave's quantitation error on both data sets.

datasets, yet centWave shows higher sample specificity in MOUSE since it rarely found a false IT. Massifquant exhibited a better F_1-score on MOUSE since it identified substantially more ITs than centWave. Both algorithm's MM14 performance is effectively equal for all metrics but sensitivity. The matchedFilter algorithm was only able to identify 33 of the 589 ITs in the MOUSE dataset after optimization over 215 parameter settings. Because matchedFilter performs so poorly compared to centWave and Massifquant, we omit the results from the charts in this paper.

Comparing algorithms' quantitation accuracy is controversial because defining IT boundaries is not clear-cut and in this analysis most error comes from the tails—knowledge afforded because of the evaluation criterion. No statistical test comparing the two algorithm's was done since the spatial components, length, shape, m/z variance, etc. likely create dependence among ITs. Nonetheless, Figure 9.2 illustrates that Massifquant and centWave quantitation errors are generally in the same small neighborhood.

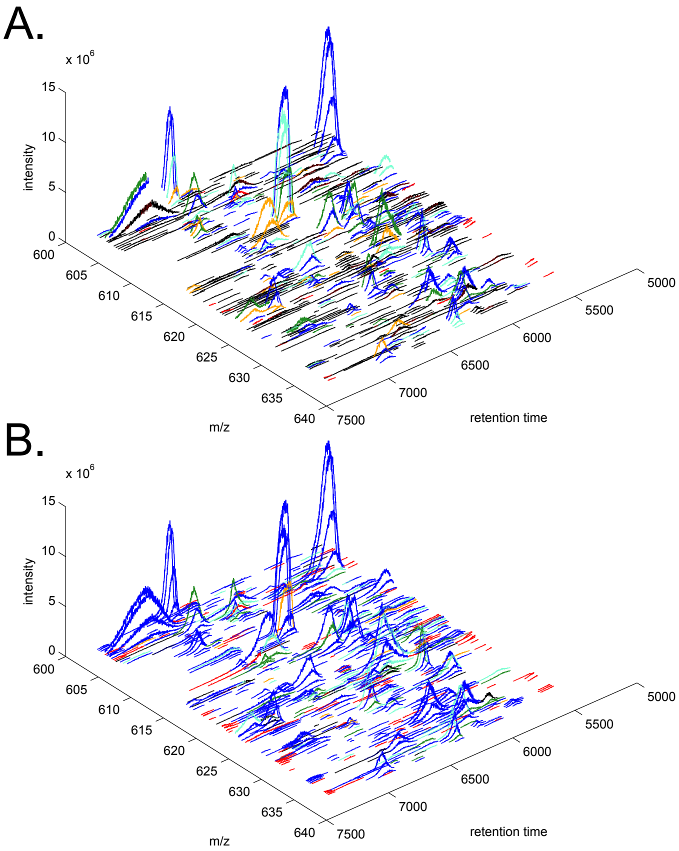


Figure 9.3: A comprehensive view of manually annotated ITs on the MOUSE data set and detected ITs, for A) centwave and B) Massifquant. Correctly identified ITs are color-coded according to the percent quantitation error (ϵ): dark blue $< 10\%$, aqua $< 20\%$, green $< 40\%$, orange $> 40\%$. False ITs are labeled in red; all other noise was excluded. ITs missed by the algorithm (i.e., false negatives) are labeled black.

9.3.2 Evaluation by IT Type

An evaluation is incomplete without identifying what types of ITs were missed within certain types of samples. For example, both algorithms are perhaps equally excellent at detecting ITs in a simple sample like MM14 with high signal-to-noise (see supp. data Figure S3). On the other hand, Figure 9.3 shows that Massifquant excels at finding low-intensity type ITs in the MOUSE complex sample and quantifies them very well, whilst these are not identified by centWave.

The “Evaluation by IT Type’ strategy’, described in section 2.3, addresses whether the high number of low-intensity ITs relative to high-intensity ITs in the MOUSE data unfairly benefited Massifquant in aggregate statistics (viz. F_1-score). Figure 9.4 summarizes the results of IT-typed performance for characteristics thought to vary widely within MOUSE. centWave's IT sensitivity improves as the intensity increases and the estimated ppm error decreases, both in a linear fashion. *Massifquant's* sensitivity varies little across all categories, irrespective of the variable, and without a doubt outperforms centWave. With respect to IT precision, the effect of each variable seems present for both algorithms. Both have similar approximate precision results. Not surprisingly, Massifquant shows improved precision as length, narrowness, and max-intensity increase.

9.4 Discussion & Conclusions

In Massifquant, we have implemented an open-source Kalman filter-based IT detection algorithm based on Aberg *et al.* [1]. We have evaluated its performance using two manually-annotated data sets, and compared the performance of Massifquant with centWave, a wavelet-based IT finder, and matchedFilter, a binning-based IT finder. A protocol for how IT detection algorithms should be evaluated has not yet been established, so we first discuss the evaluation process; then, we address algorithmic performance and suitability for use, and finally conclude with some thoughts about the use of m/z information in MS IT detection generally.

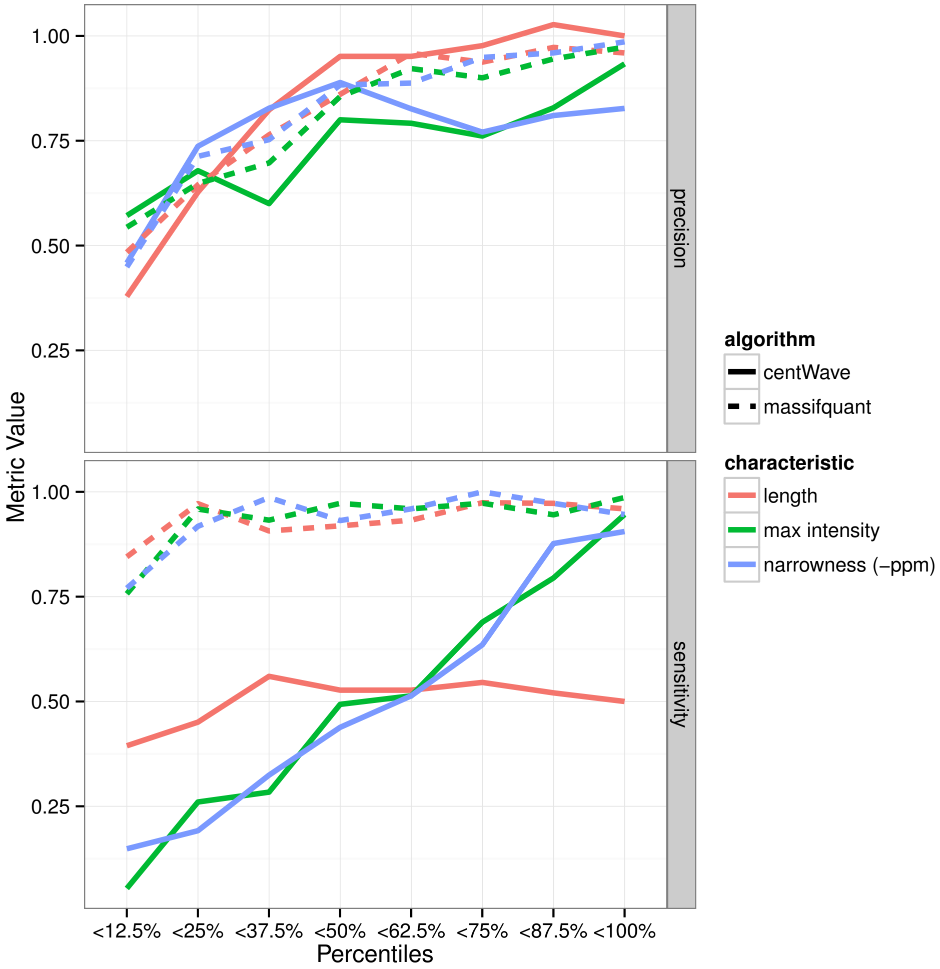


Figure 9.4: Isotope trace detection performance across various quantiles for different IT characteristics of the MOUSE data set. The left-most percentile bins generally represent the hardest cases for IT detection algorithms (short, low intensity, broad ITs) while bins on the right are generally easier (long, high intensity, narrow ITs). The sensitivity panel is at the IT-level.

9.4.1 The evaluation process

Comparative evaluation of algorithms in MS-omics is often lacking Smith *et al.* [98], and Zhou *et al.* [131] recently suggested that the quantitative evaluation of IT detection algorithms is long overdue. We believe the general lack of evaluation is related to the difficulties associated with creating data sets to effectively test these algorithms and also to a lack of clear and explicit metrics for assessing success. In order to facilitate further efforts in this area, we discuss some of the challenges and successes we met using a manually annotated data set approach.

Hand-annotation, especially of low abundance ITs, is extremely challenging. It requires concerted effort over a long period of time. The authors spent several weeks of dedicated effort in order to annotate the two data sets, and the MOUSE data set is only a small subset of the complex LC-MS sample from which it was derived. Despite our best efforts to be accurate and consistent, we conclude that the manual annotation process is still somewhat subjective. Indeed, we simply had to exclude the evaluation of ITs below a certain threshold because we felt human judgment was inadequate for the task. Despite these challenges, the annotation data itself is a useful model for future validation efforts. Moreover, it contains isotopic-level information that could be of use in other projects.

We validated the manual annotation efforts through a holistic visual inspection (see Figure 9.3 for example) and analysis of histograms of ppm deviation (see supp. Figure S4 for example) to ensure that there were no outliers. So, despite the inherent difficulty of manual annotation, we conclude that the endeavor was largely successful. Several aspects of the process are worth considering in more depth: 1) We used semi-rigid guidelines for annotation that we believe worked well across a variety of ITs with different characteristics. We could have generated and applied very strict rules for annotation at the outset, but this may have resulted in even worse systematic bias considering the highly variable ITs we encountered. The proposed guidelines should serve useful for future annotation efforts. 2) We used a single annotator for both data sets to eliminate person-to-person variability in the interpretation and application of IT criteria. However, tools for community-sourcing annotations would be an interesting solution and has been already been discussed in genomic contexts [42]. 3) We used ToppView, the MS viewer associated with OpenMS, to help us find and annotate ITs [52, 101]. Additional add-ons such as color-coding and flagging of already-annotated ITs and producing a community-based validation would also improve the annotation process.

Among the previous efforts to evaluate IT detection algorithms, we found that most of them focused solely on questions of identification, but lacked in detail of what constituted an ’identified’ IT. For IT detection, the identification criterion is critical for fair evaluation—and we additionally argue that the evaluation should probe quantitation accuracy if possible. We evaluated identification at IT and sample levels, and also calculated the percent quantitation error for each IT. The precisely defined metrics may now be more easily employed, modified, or improved.

This multi-metric evaluation exposes two risks other evaluations take when relying purely on the F_1-score. 1) Precision values show that Massifquant does at least as well if not better at IT identification reliability for MOUSE at low intensity. However, the sample specificity, along with Figure 9.3, provide stronger evidence that centWave effectively discriminates low-intensity non-ITs better than Massifquant. Hence, precision and consequently the F_1-score can be misleading. To our knowledge, this is the first evaluation that has proposed a true specificity measure for IT detection, which helps avert wrong conclusions. 2) By our evaluation standards, and likely others, accurate quantitation does not always imply a favorable IT detection F-score and vice-versa. On the MOUSE dataset centWave ignores many low-intensity ITs, giving it a low F-score; however, the ITs that it does identify are generally quantitatively accurate with a median $\epsilon = 8.663\%$. Thus, quantitative accuracy is somewhat distinct from IT detection sensitivity or precision.

9.4.2 Algorithm performance

On the simple data set MM14, Massifquant showed similar performance to centWave. On a highly complex sample, MOUSE, Massifquant performed much better. In particular, Massifquant excels at finding ITs with a variety of characteristics such as differing intensity, widths, and lengths. Massifquant outperforms centWave in IT detection sensitivity across every size and shape of ITs in the complex sample tested. As for reliability, Massifquant is competitive with centWave with the exception that it finds more false low-intensity ITs; the excess false positives and multi-modal artifacts are two deficiencies of Massifquant which can complicate downstream analysis in sample-to-sample comparisons. Future extensions of Kalman Filter IT detection will need to make intensity estimation more robust. An attempt to combine centWave's wavelet intensity estimation with Massifquant has not proven to be effective (see supplement section 4). In spite of these deficiencies, both algorithms reported similar quantitation accuracy for the quantified ITs; Massifquant just found far more ITs.

A possible objection to our general comparison is that a large number of small ITs might bias the evaluation in Massifquant's favor. However, Figure 9.4 removes any suspicion of unfair advantage; even if low-intensity or very broad ITs (e.g. first four bins) were removed from the analysis, Massifquant still identifies ITs better on the MOUSE data set.

As shown in Figure 9.1, our effort to address the problem of IT segmentation with Massifquant was successful—on the MOUSE and MM14 data set, the precision increased from 0.7391 to 0.7894 and 0.9185 to 0.9355, respectively. However, some ITs were erroneously combined (see supp. data Figure S2) . For algorithmic simplicity, future efforts should attempt to address the IT segmentation problem from within the framework of the Kalman filter. Ideally, such an approach would also be more effective than the ad-hoc method we applied in this study to treat IT segmentation.

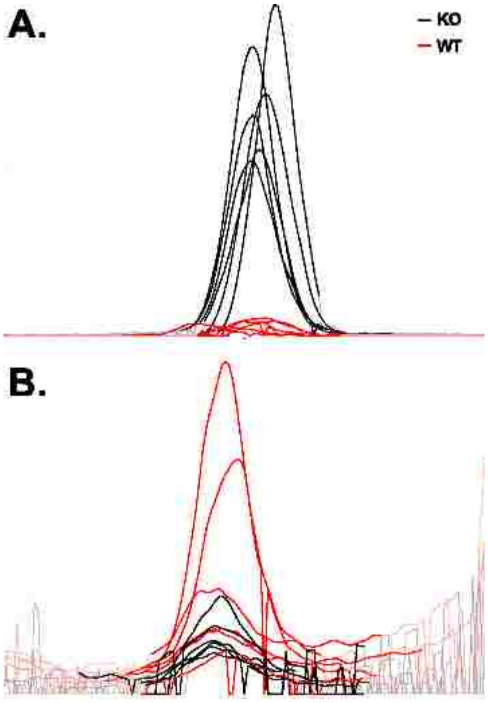


Figure 9.5: Massifquant identifies differentially expressed ITs between wild-type (WT) vs. knock-out (KO) conditions in the faahKO dataset for (A) trivial cases and (B) non-trivial cases.

9.4.3 Ease of use

Massifquant parameters can be readily optimized through visual confirmation instead of score-based methods (e.g. f-score) that require an annotation. Visual optimization is more time efficient, intuitively simple, and almost as accurate. Similar in purpose to Tengstrand *et al.* [106], the visualization tools at `https://github.com/topherconley/optimize-it` illustrate precise changes in IT detection induced by differing parameter input. The documentation offers a step-by-step guide how to optimize Massifquant to new data sets, especially controlling the number of false positives. Further, the score-based method shows a concave f-score surface when varying Massifquant's parameters, indicating a very predictable parameter behavior (see supp. data Figure S5,S12,S13,S14). Massifquant's appeal is due, at least in part, to the the fact that several internal KF parameters are learned from the data—in an initial prescan, and then later for each individual IT being tracked.

Massifquant operates on centroided MS data, which means it can analyze data taken in centroid mode or profile mode (after centroiding), whereas algorithms requiring profile data cannot operate on centroid data because the centroiding process is not readily reversible. Further, running Massifquant is as easy to run and modular as other XCMS IT detection options. The same differential abundance (DA) workflow applies. Figure 9.5 illustrates a Massifquant-based DA analysis in on the FAAH knock out LC/MS data set [82], (see `http://bioconductor.org/packages/devel/data/experiment/manuals/faahKO/man/faahKO.pdf` for details).

9.4.4 The use of m/z information in IT detection

Can the success of Massifquant on a complex sample be generalized? ITs in a highly complex sample—particularly low abundance ITs—are different from ITs derived from a simple mixture: limitations in a mass spectrometer's dynamic range produce much greater intensity variability for ITs from a complex sample. Because of this, at least for mid-to-high mass accuracy/resolution mass spectrometers, m/z measurements will tend to be far more helpful at distinguishing closely eluting species than IT shape. Indeed, we found that Massifquant performs at a high level because of its m/z estimation (despite extremely poor intensity estimation). Most IT detection algorithms focus on IT shape, but we suggest that on highly complex samples an algorithm should be focused mainly on subtle changes in m/z. Algorithms that bin data from closely related ITs in order to do IT shape analysis lose the richest information available for distinguishing those ITs. Distinguishing convolved isobaric compounds and near-isobaric compounds will, of course, require chromatographic IT shape analysis, but new algorithms will likely see the greatest improvement gains by working to fully utilize the m/z information found in closely eluting analytes.

9.5 Acknowledgement

We greatly appreciate assistance from Jeffrey Humpherys (BYU, Dept. of Math), William F. Christensen (BYU Dept. of Statistics) and Steffen Neumann (Leibniz Institute of Plant Biochemistry). This work was supported by National Science Foundation grant No. 0639328, institutional startup funds from Brigham Young University, and NSF GRF [DGE-0750759] to RS.

Chapter 10

Statistical Agglomeration: Peak Summarization for Direct Infusion Lipidomics

Abstract

Motivation: Quantification of lipids is a primary goal in lipidomics. In direct infusion/injection (or shotgun) lipidomics, accurate downstream identification and quantitation requires accurate summarization of repetitive peak measurements. Imprecise peak summarization multiplies downstream error by propagating into species identification and intensity estimation. To our knowledge, this is the first analysis of direct infusion peak summarization in the literature.

Results: We present two novel peak summarization algorithms for direct infusion samples and compare them with an off-machine ad-hoc summarization algorithm as well as with the propriety Xcalibur algorithm. Our statistical agglomeration algorithm reduces peakwise error by 38% (m/z) and 44% (intensity) compared to the ad-hoc method over 3 data sets. Pointwise error is reduced by 23% (m/z). Compared to Xcalibur, our statistical agglomeration algorithm produces 68% less m/z error and 51% less intensity error on average on two comparable data sets.

Availability: The source code for Statistical Agglomeration and the data sets used are freely available for non-commercial purposes at `https://github.com/optimusmoose/statistical_agglomeration`. Modified Bin Aggolmeration is freely available in MSpire, an open source mass spectrometry package at `https://github.com/princelab/mspire/`.

10.1 Introduction

Direct infusion (injection) lipidomics, sometimes called "shotgun" lipidomics for it's similarity to shotgun genomics, is an emerging but well studied field [29, 30, 113]. Here, a liquid sample is injected into a mass spectrometer, yielding a set of (mass/charge (m/z), intensity, retention time (RT)) 3-tuples [45]. For our purposes, we define a data point as a single m/z and intensity observation of a given isotope at a particular RT and a peak as the data points that comprise the observation of a distinct isotope. (Hereafter, we will more accurately use the term *ridge* instead of *peak* due to the fact that direct injection lipid intensity does not vary as a function of time.) Since there is no chromatographic separation in direct infusion lipidomics, each RT scan represents an independent measurement of the sample. Ideally, the species in the sample would be uniformly distributed across RT and measured in near identical intensities across RT, making reduction to a single two-dimensional vector of unique ridges trivial. Unfortunately, there are several noise factors that appear in real world direct infusion samples. Sample distribution heterogeneity results in inter-scan variance in both m/z and intensity. What's more, technical and mechanical limitations in the mass spectrometer inculcate even more error into the output. Accurately estimating the true ridge values from the resulting output file is a nontrivial challenge (see Figure 10.1).

In order to identify and quantify each lipid, it's component ridges must somehow be isolated one from another, and the additive noise ridges removed. We will call this process *ridge summarization*. Only after ridge summarization can the isotopic envelopes be compared with theoretical databases in order to identify and quantify the individual lipids in the sample.

The necessity of a solution for the ridge summarization problem in every direct infusion lipidomics application and the presumed effect of the results of such a solution on downstream quantitation would suggest that a description of ridge summarization be found in every shotgun lipidomics study [84]. However, it is frequently left unmentioned (e.g. [71], [99], and [29]). Although direct infusion methods have been around since the mid-1990s, we are only aware of two published solutions to this segment of the quantitation pipeline. The first is that of treating a survey scan as a true ridge measurement [88]. From a glance at a typical shotgun lipidomics plot, it should be clear that treating any single RT scan of data as a representative set of true ridges would be less than ideal, as the scan would include many ridges with incorrect m/z and intensity and exclude many other true ridges (see Figure 10.1). The second, a more robust approach, applies to shotgun lipidomics a technique that has been used in several proteomics studies [38, 60]. This approach, which we label the fixed width algorithm, averages scans across the retention time dimension to yield an estimation of the true contents of the sample [47]. Though this approach is simple to code and runs in linear time, it is non-statistical and does not take into account the data densities along the m/z axis.

Here we present two statistical approaches to solving the ridge summarization problem and evaluate them against both synthetic and real-world ridge summarization problems. We also provide the first comparative performance analysis of Xcalibur and the fixed width algorithm on the ridge summarization problem.

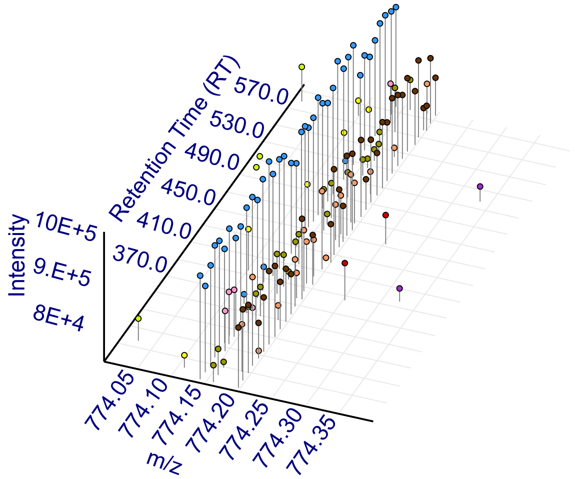


Figure 10.1: A typical direct infusion lipidomics sample. The lack of consistent repetition in data points in the RT dimension and the abundance of noise in each of the three dimensions make accurate ridge summarization difficult. The colors delineate observed ridges.

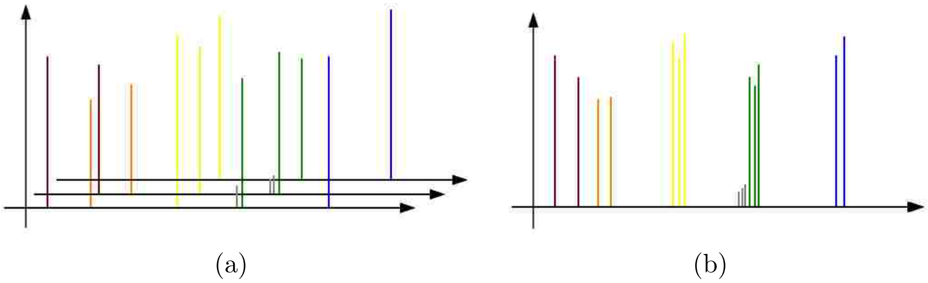


Figure 10.2: Scan Combination. Here (a) multiple scans are combined into (b) one list of (m/z, intensity) pairs by removing the retention time (RT) dimension. Each data point (an m/z and intensity observation of a distinct isotope at a given RT) is depicted here with a pinhead, while the collection of pinheads of one color denotes a ridge.

10.2 System and Methods

We use a representative sample of three labeled data sets to test the capabilities of the methods we present as well as the baseline results from the widely used Xcalibur software shipped with Thermo mass spectrometers.

10.2.1 Data

The methods presented in this paper were evaluated on one synthetic data set and two real world, hand labeled data sets.

The Noyce data set is a synthetic data set constructed as described in [70] with sampling rate 1, noise density factor 500, and one dimension mode.

Sample_3_750-800 and Sample_3_1000-1050 are two m/z intervals of a rat soleus lipid extract. Each ridge in these data sets were isolated and labeled by hand using TOPPView [101] and an exhaustive list of all (m/z, intensity, RT) triplets in the file.

Each of the data sets used in lipidomics can be represented as a list of points where each point is an (m/z, intensity, RT) triplet. For the purposes of the algorithms detailed here, points are reduced to m/z and intensity values (see Figure 10.2).

Sample_3_750-800 and Sample_3_1000-1050 are two m/z intervals of a rat soleus lipid extract. The total lipids from about 20 mg of the mouse muscle cells grown in

DMEM + 10% fetal bovine serum medium were extracted using a modified Bligh and Dryer method. Lipids were dried under a nitrogen stream and 14 micrograms of the internal standard (cer 18:1/17:0) was added before the analysis. The whole lipids were dissolved in Chloroform:Methanol:Isoproponal (2:1:1) containing 11 mm Ammonium acetate and analyzed for 10 minutes in an LTQ-orbitrapXL mass spectrometer with positive polarity, 4.2KV ESI ionization voltage, 35V capillary voltage, and 110V tube lens voltage. The MS1 settings were FTMS analyzer, a mass range of 450 -2000 m/z, resolution of 100,000, full scan, centroid data.

All data sets were obtained in centroid mode. Designing algorithms to treat data in centroid mode is more general than choosing to only handle profile mode due to the lack of ability to convert centroid data to profile data when the opposite conversion is readily possible. This design decision is reflected in industry software standards such as CentWave [104].

10.2.2 Metrics

Each of the following metrics measures a different quality of ridge assignment. Since each algorithm has different strengths, these metrics allow a ranking of algorithms based on what is important for the practitioner. Since we cast the ridge selection problem as a clustering problem, all of the following metrics are established clustering metrics, with the exception of normalized true ridge distance, which is a metric devised specifically for measuring the quality of summarized ridges.

In what follows, we define $\mathbf{R}$ as the set of observed ridges, $\hat{\mathbf{R}}$ as the set of predicted ridges, and $\mathbf{D}$ as the set of data points. We define the intensity, $\hat{r}_{int}$, of a predicted ridge $\hat{r}$ as the sum of the intensities of the ridge's assigned points:

$$\hat{r}_{int} = \sum_{d \in \hat{r}} d_{int} \tag{10.1}$$

and the m/z value, $\hat{r}_{m/z}$, of $\hat{r}$ as the intensity-weighted mean of the m/z value of the ridge's assigned points:

$$\hat{r}_{m/z} = \sum_{d\in\hat{r}} d_{m/z}\frac{d_{int}}{\hat{r}_{int}} \tag{10.2}$$

NORMALIZED TRUE PEAK DISTANCE (NTPD). NTPD is a metric we developed for this task which indicates the normalized m/z or intensity difference between the predicted ridges and the nearest observed ridges. The nearest observed ridge, $\tilde{r}$, to a predicted ridge $\hat{r}$ is always calculated using m/z/ value as:

$$\tilde{r} = \underset{r\in\mathbf{R}}{\operatorname{argmin}}(|\hat{r}_{m/z} - r_{m/z}|) \tag{10.3}$$

Using this closest observed ridge, the m/z NTPD is calculated as

$$\mathrm{NTPD}(\hat{\mathbf{R}},\mathbf{R}) = \frac{1}{\min(|\hat{\mathbf{R}}|,|\mathbf{R}|)}\sum_{\hat{r}\in\hat{\mathbf{R}}}(|\hat{r}_{m/z} - \tilde{r}_{m/z}|) \tag{10.4}$$

while the intensity NTPD is calculated using the same equation (Eq. 10.4) with $\hat{r}_{m/z}$ and $r_{m/z}$ replaced with $\hat{r}_{int}$ and r_{int}.

The normalizing term controls score inflation whether the error is in predicting too many or too few ridges. The significance of this metric is reflected in its analytical relevancy. This per-ridge metric basically measures how easy it would be to correctly assign the true species label using a standard lipid species library. Such is not the case for a per-point error measure such as sum squared error (SSE) or an intrinsic cluster metric like normalized mutual information (NMI) or purity.

Δ NUMBER OF RIDGES. In downstream algorithms, each estimated ridge will be treated as an actual isotope. It is clear that any identification or quantitation algorithms will be highly sensitive to the number of predicted ridges versus the number of actual ridges.

PURITY. Purity measures the averaged homogeneity of each estimated ridge over all data points. It is defined as:

$$\operatorname{purity}(\hat{\mathbf{R}}, \mathbf{R}) = \frac{1}{|\mathbf{D}|} \sum_{\hat{r}\in\hat{\mathbf{R}}} \max_{r\in\mathbf{R}} |\hat{r}\cap r| \tag{10.5}$$

A purity of 1 is perfect, and zero is the lowest possible score. One way to achieve high purity is to reduce the size of the predicted ridges. In fact, a naïve algorithm that simply assigns each data point into its own ridge will achieve a perfect score for purity.

NORMALIZED MUTUAL INFORMATION (NMI). NMI allows the quantitation of the trade off between number of predicted ridges and the quality of predicted ridges.

$$\operatorname{NMI}(\hat{\mathbf{R}}, \mathbf{R}) = \frac{\mathrm{I}(\hat{\mathbf{R}}, \mathbf{R})}{[\mathrm{H}(\hat{\mathbf{R}}) + \mathrm{H}(\mathbf{R})]/2} \tag{10.6}$$

where I is mutual information, given by

$$\mathrm{I}(\hat{\mathbf{R}}, \mathbf{R}) = \sum_{\hat{r}\in\hat{\mathbf{R}}} \sum_{r\in\mathbf{R}} \frac{|\hat{r}\cap r|}{|\mathbf{D}|} \log \frac{|\mathbf{D}||\hat{r}\cap r|}{|\hat{r}||r|} \tag{10.7}$$

and H is entropy, given by

$$\mathrm{H}(\hat{\mathbf{R}}) = -\sum_{\hat{r}\in\hat{\mathbf{R}}} \frac{|\hat{r}|}{|\mathbf{D}|} \log \frac{|\hat{r}|}{|\mathbf{D}|} \tag{10.8}$$

NMI indicates the dependence of sets $\hat{\mathbf{R}}$ and $\mathbf{R}$. If they are completely independent the ridge predictions provide no information about the observed ridge assignments (indicated by an NMI of 0). A perfect score of 1 indicates that the observed ridge assignments provide no additional information beyond that provided by the predicted ridge assignments.

SUM SQUARED ERROR (SSE). SSE is a common measurement of error. It is computed by summing the squared error of each assignment.

For the SSE of the m/z dimension, we use:

$$\mathrm{SSE}(\hat{\mathbf{R}}, \mathbf{R}) = \sum_{d \in \mathbf{D}} (\hat{r}^d_{m/z} - r^d_{m/z})^2 \tag{10.9}$$

where $\hat{r}^d_{m/z}$ indicates the m/z of the predicted ridge containing point d and $r^d_{m/z}$ indicates the m/z of the observed ridge containing point d.

Intensity SSE is calculated in the same fashion, with intensity replacing m/z in Equation 10.9.

TREATMENT OF NOISE POINTS. The clustering metrics used here were modified to address the problem of noise points (whether true noise or falsely assigned as such), which are not a typical occurrence in standard clustering problems. Let $\mathbf{R_0}$ denote the set of observed noise points and $\mathbf{N}$ denote the set of points assigned as noise points.

Purity was modified to consider only the points $\mathbf{D} \setminus (\mathbf{R_0} \cup \mathbf{N})$. In other words, purity ignores both assigned and observed noise points. Thus, for $\hat{r} \in \hat{\mathbf{R}}$, $\hat{r}$ is replaced by $\hat{\rho}$ (see Equation 10.10) and $\hat{\mathbf{R}}$ is replaced by $\hat{\mathbf{P}}$ (see Equation 10.11).

$$\hat{\rho} = \{i \mid i \in \hat{r} \wedge i \notin \mathbf{R}_0\} \tag{10.10}$$

$$\hat{\mathbf{P}} = \bigcup_{\hat{r} \in \hat{\mathbf{R}}} \hat{\rho} \setminus \mathbf{N} \tag{10.11}$$

With this change, the equation for purity becomes:

$$\mathrm{purity}(\hat{\mathbf{R}}, \mathbf{R}) = \frac{1}{|\mathbf{D}|} \sum_{\hat{\rho} \in \hat{\mathbf{P}}} \max_{r \in \mathbf{R}} |\hat{\rho} \cap r| \tag{10.12}$$

We posit that purity should not be increased in the case where a prospective ridge is exclusively composed of points labeled by the summarization method as noise, since these points do not belong to an observed ridge and should not therefore contribute to the purity measurement of the observed ridge assignments. Likewise, our modifications prevent the case where a noise point misassigned into a ridge incorrectly covers for an observed ridge member that has been incorrectly assigned to noise or another ridge.

NMI was modified as follows: To provide an equitable comparison for the entropy and mutual information terms, we make each true noise point its own true ridge. Formally,

$$\mathbf{R}_{NMI} = \mathbf{R} \bigcup_{d\in\mathbf{R}_0} \{d\} \tag{10.13}$$

Additionally, we remove observed noise points assigned to ridges (and therefore not assigned as noise) from $\mathbf{R}$, as including them would obviate the penalty due when predicted ridges replace observed ridge members with noise points. This decision provides an entropic penalty to observed noise points incorrectly assigned to ridges since they would otherwise offset the deficient number of points in the predicted ridges they are assigned to. However, assigned noise points are included in $\mathbf{R}$, with each point treated as its own ridge. Treating them as their own ridge provides a more local and reasonable way of measuring error than, say, comparing individual noise ridges to the centroid value of a ridge containing all the noise points.

$$\hat{\mathbf{R}}_{NMI} = \bigcup_{\hat{r}\in\hat{\mathbf{R}}} \hat{r} \setminus \mathbf{N} \bigcup_{d\in\mathbf{N}\wedge d\notin\hat{r}\forall\hat{r}\in\hat{\mathbf{R}}} \{d\} \tag{10.14}$$

Thus the NMI equation becomes:

$$\mathrm{NMI}(\hat{\mathbf{R}}, \mathbf{R}) = \frac{\mathrm{I}(\hat{\mathbf{R}}_{NMI}, \mathbf{R}_{NMI})}{[\mathrm{H}(\hat{\mathbf{R}}_{NMI}) + \mathrm{H}(\mathbf{R}_{NMI})]/2} \tag{10.15}$$

For SSE, the point assignment error calculation is dependent on whether or not the point d is really a noise point and whether or not the point is assigned as error. There are four cases (see Table 10.1):

- If a real point is assigned as noise ($d \notin \mathbf{R}_0$ and $d \in \hat{\mathbf{R}}_0$), the error amount is the distance between the point and the observed ridge the point belongs to.

- If a noise point is assigned as a noise point ($d \in \mathbf{R}_0$ and $d \in \hat{\mathbf{R}}_0$), there is no error.
- If a noise point is assigned as a real point ($d \in \mathbf{R}_0$ and $d \notin \hat{\mathbf{R}}_0$), the error is the distance between that point and the closest observed noise point.
- If a real point is assigned as a real point ($d \notin \mathbf{R}_0$ and $d \notin \hat{\mathbf{R}}_0$), the error is the distance between the predicted ridge the point was assigned to and the observed ridge the point belongs to.

Table 10.1: SSE error calculation for an assigned point δ. When δ is noise and assigned as such, there is no error. If δ is a real point but assigned as noise, the error amount is the distance between δ and the observed ridge δ belongs to. When δ is a noise point and is assigned as real, the error is the distance between δ and the closest observed noise point. When δ is real and assigned to ridges, the error is the distance between the predicted ridge δ was assigned to and the observed ridge δ belongs to. For intensity SSE, replace m/z values with intensity.

		Actual	
		Noise	Real
Predicted	Noise	0	$\lvert\delta_{m/z} - \hat{r}^{\delta}_{m/z}\rvert$
	Real	$\min_{r \in \mathbf{R}_0}(\lvert\delta_{m/z} - d_{m/z}\rvert)$	$\lvert r^{\delta}_{m/z} - \hat{r}^{\delta}_{m/z}\rvert$

10.3 Algorithms

While both methods proposed as well as the fixed width method follow the ridge summarization paradigm by combining multiple scans (see Figure 10.2), each of the three methods diverges in the way the ridges are segmented once combined into one spectra.

10.3.1 Fixed Ridge Width Method

Many practitioners use some variant of this method (e.g., [84]). Defining the ridge width in terms of the mass of the given point models the variation of resolution along the m/z scale [47] (see Figure 10.3). The combined spectra (see Figure 10.2) are sliced into adjacent bins of width $\frac{m/z}{resolution}$, where m/z is the m/z at the current point and *resolution* is the resolution of the machine. Each bin is then treated as a ridge.

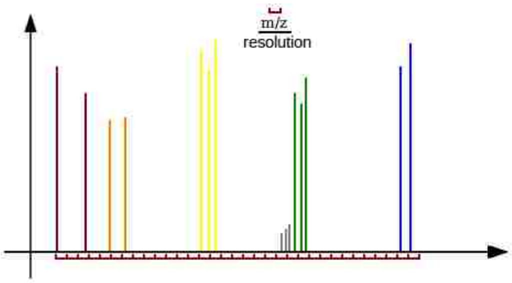


Figure 10.3: Fixed Width Segmentation. The combined spectra (see Figure 10.2) are sliced into bins of width $\frac{m/z}{resolution}$. Note how fixed width has no means of detecting data density, nor comparing the intensity of points. The shadow ridge (gray) is indistinguishable from the geen ridge next to it, despite the intensity difference. Also, note how the hard bin limits segment observed ridges that happen to fall on both sides of a bin interval. The colors delineate observed ridges. The red segments along the x-axis indicate bin boundaries.

10.3.2 Modified Bin Agglomeration

Modified Bin Agglomeration (MBA) uses a series of decisions based on the shape of intensity histogram bins to partition the data into ridges. First, the data is binned according to the Fixed Width algorithm, except with a user-defined bin width whose default is 5ppm for the Orbitrap XL (see Figure 10.4). After this initial binning, the contiguous bins demarcated by empty bins are considered ridges. Note the difference between this and the Fixed Width algorithm, which considers hard contiguous bin intervals as ridges irrespective of the content of each bin. At this point, if the user has selected the `zero` option, the algorithm is complete.

There are two other options available: `share` and `greedy_y`. Both options split all ridges where the sum of the intensities of each bin form a local minima within a series of contigous bins. The difference between the `share` and `greedy_y` options consists of how these local minima are treated (see Figure 10.5).

10.3.3 Statistical Agglomeration

Statistical Agglomeration (SA) bases bin agglomeration decisions on statistical analysis of the data. The approach here is to treat ridges as distributions and bins of data as samples from those distributions. Although there is no guarantee that the samples being tested are

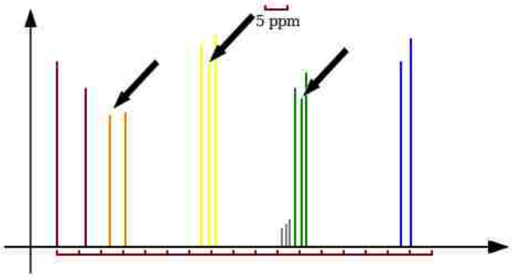
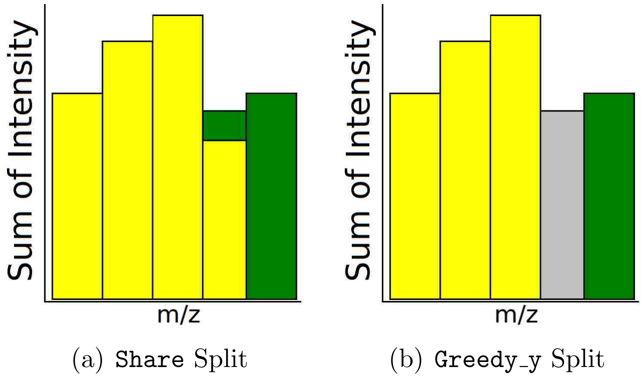


Figure 10.4: Modified Bin Agglomeration Segmentation. The combined spectra (see Figure 10.2) is sliced into bins of user-defined width (default 5ppm). MBA then segments existing bins into disparate ridges at local minima (black arrows). The colors delineate observed ridges. See Figure 10.5 for more detail on MBA bin splitting.



(a) `Share` Split (b) `Greedy_y` Split

Figure 10.5: MBA Bin Splitting. After segmenting all points into fixed interval bins and creating initial ridges of each contiguous segment bounded by empty bins, the MBA algorithm further divides ridges by considering local minima. With the `share` method (a), the local minimum is split among adjoining ridges proportional to the neighboring ridges' intensities. The `greedy_y` method (b) awards the entire disputed bin to the adjoining ridge of greatest total intensity. Note that the bars in this figure represent histograms of the intensity of the points in the assigned bins, not the component points themselves.

normally distributed, we make this assumption in order to use the t-test. ridges (distributions) whose means are not statistically different according to this test are combined iteratively until all remaining ridges are statistically different with high confidence.

As with the previous methods, the data is first sorted by ascending m/z and split into bins of size m/z_{window} (see Figure 10.6):

$$m/z_{window} = resolution \times 10^{-7} \tag{10.16}$$

This formula was empirically derived from observation of several lipid samples to yield a good balance between minimal window size and sufficient size to estimate ridge statistics, and it should be applicable across many mass spectrometers.

After the initial bin assignment, starting at the lowest m/z value, adjacent bins are subjected to a Welch t-test [116] (we use the Welch t-test because the samples (bins) have potentially different sizes and variances) to test the hypothesis that the two sample distributions have the same mean:

$$t = \frac{\bar{X_1} - \bar{X_2}}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \tag{10.17}$$

where $\bar{X_i}$, s_i^2 and N_i are the i^{th} sample mean, sample variance and sample size, respectively. The degrees of freedom are approximated using the Welch-Satterthwaite equation [85]:

$$v = \frac{(\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2})^2}{\frac{s_1^4}{N_1^2 \cdot (N_1 - 1)} + \frac{s_2^4}{N_2^2 \cdot (N_2 - 1)}} \tag{10.18}$$

For each potential bin agglomeration, the p value is obtained from a t-distribution for a two-tailed test for the computed t and v values (see Eq. 10.17, 10.18) to validate the null hypothesis that the ridge means are equal. If the p value is greater than 0.01, meaning the confidence that they are different is less than 99%, we accept the null hypothesis and combine the bins being tested. Note that, in order to accommodate a test of both the m/z

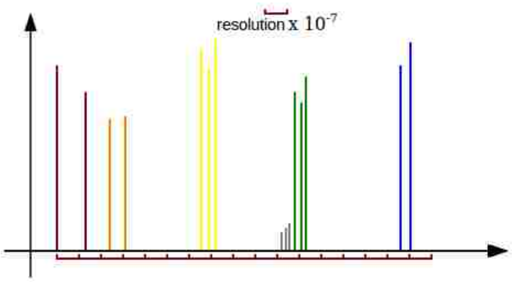


Figure 10.6: Statistcal Agglomeration Segmentation. The combined spectra (see Figure 10.2) is sliced into bins of width $resolution_{\times}10^{-7}$. The colors delineate observed ridges. The red segments along the x-axis indicate bin boundaries.

and intensity differences of the considered bins, each tested bin pair is subjected to two *t*-tests, one using the m/z data and one using the intensity data. As an overall measure of confidence, we use the maximum *p* value for the two *t*-tests. The approach here is to be no more confident than our least confident *t*-test dimension (intensity or m/z). This design decision provides an implicit awareness of situations which would be deceptive if the minimum *p* value were used as an overall measure of confidence, such as when two bins have a very similar m/z values but very different intensities. This situation, which we call *shadow ridges*, occurs surprisingly often when a low intensity ridge appears directly adjacent to a very high intensity ridge. This approach also helps discriminate in cases when two bins that should not be combined are similar in average intensity. This is a common occurrence at low intensities. In this case, the lack of confidence in the m/z dimension will prevent combination of the two ridges.

In the event that the two bins under consideration are combined, the resulting agglomerated bin is considered as a single bin in the next iteration's comparison to the next bin in ascending m/z order. If they are not combined, the first bin in m/z order remains unchanged, and with the next iteration the second bin is compared with the next subsequent

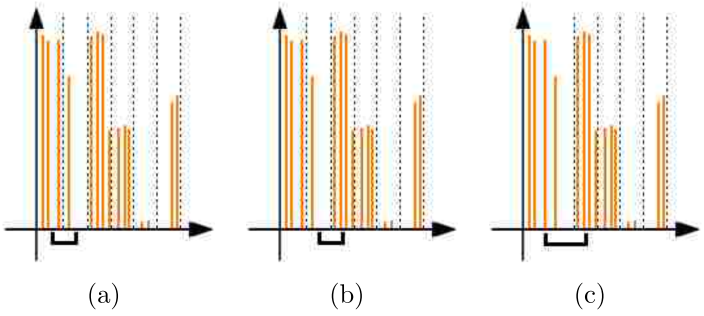


Figure 10.7: SA Bin Agglomeration. After sorting the data by m/z value, and assigning data points to bins of fixed width, a *t*-test is conducted on the intensity and m/z means of the first two bins (a). If either of the *t*-tests fail to show a high confidence that the means are different, the bins are not combined and the algorithm considers the next two bins for agglomeration (b). Otherwise, the two bins are agglomerated, and the algorithm considers the agglomerated bin and the next bin for agglomeration (c). Dotted lines indicate ridge boundaries.

bin in ascending m/z order (see Figure 10.7). The entire algorithm runs in just one pass, resulting in O(n) performance, where n is the number of bins.

For post-processing noise removal, we use an established noise filtering method where all points with intensities below the estimated noise level (signal to noise ratio (s/n) = 1) are labeled as noise and removed. This method is borrowed from Samuelsson, *et al.*, but we modify the quantitation of noise from an intensity level to a frequency count, which is more robust to lower intensity signals [84]. This approach rests on the assumption that noise points are distributed uniformly, and thus should be equally distributed across the initial bins. The expected noise level is one noise point per bin.

10.3.4 Xcalibur

Xcalibur is a propriety mass spectrometry software platform from Thermo Scientific. Since Xcalibur will not accept data in the community standard mzML format, we were unable to use it on the Noyce synthetic data set [24]. However, the raw data of the Sample_3 data sets were analyzed using Xcalibur 2.1.

10.4 Results

SA generally outperforms the other methods under consideration across all data sets on both the qualitative and quantitative measures considered in this study.

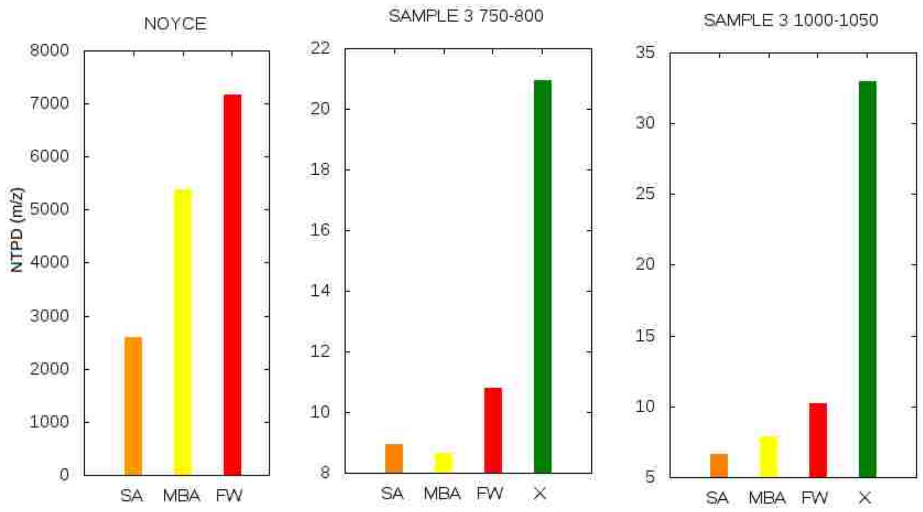
SA and MBA outperform all other methods on NTPD m/z (see Figure 10.8). MBA had a slightly lower NTPD rate on Sample_3_750-800, while SA outperformed all other methods on the other two data sets. The relative performance was identical for NTPD intensity, with the exception being more disparity between the SA and MBA scores and Fixed Width on the Noyce data set (see Figure 10.9(a)). Note that Xcalbur's NTPD is dramatically higher for both NTPD intensity and NTPD m/z than all other methods on the two data sets that were comparable given Xcalibur's proprietary data limitations.

SA predicted the number of ridges far more accurately than any other method tested, including Xcalibur, which was furthest from the actual number of ridges (see Figure 10.10). MBA was second-best on average at predicting the correct number of ridges.

On average, each of the three methods performs rather similarly on purity. The scores averaged across all three data sets are 0.73, 0.7, and 0.74 for SA, MBA, and Fixed Width respectively (see Figure 10.11). Because we are ignoring all noise points (real or assigned), and because Fixed Width produces the narrowest ridges, it is not surprising that Fixed Width performed so well on purity.

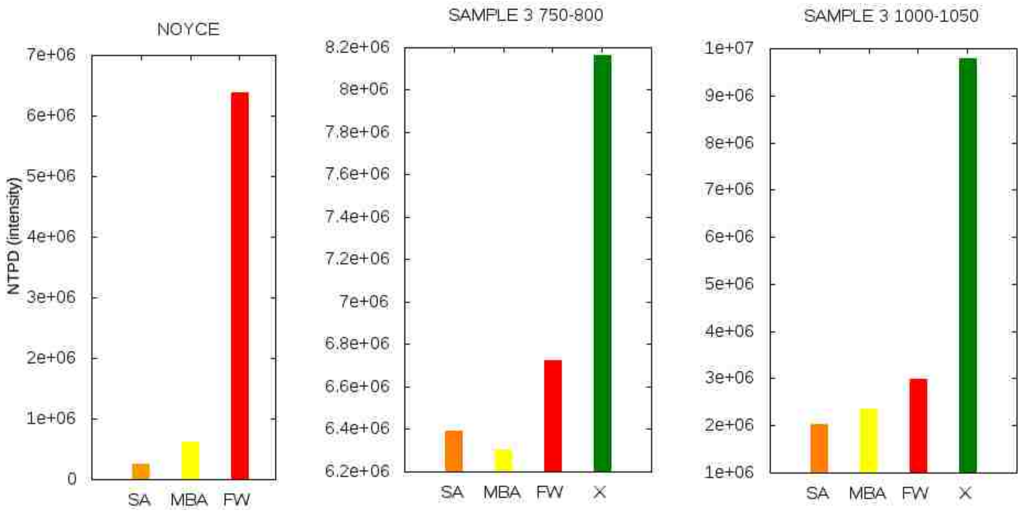
The NMI scores averaged across all three data sets are 0.95, 0.96, and 0.93 for SA, MBA, and Fixed Width respectively (see Figure 10.12). It is surprising that they are so close, but this is likely a result of the modifications to this metric to handle noise.

Each of the three methods performs inconsistently on SSE. SA outperforms the other methods on both Sample 3 data sets for m/z SSE, but MBA has a dramatically lower SSE



(a) (b) (c)

Figure 10.8: Normalized True Peak Distance (NTPD) - m/z. NTPD is a difference metric that compares the predicted ridge to the nearest observed ridge. Here we compare the ridges' m/z values resulting from each of the four methods (Statistical Agglomeration, Modified Bin Agglomeration, and Xcalibur) using the (a) Noyce, (b) Sample_3_750-800, and (c) Sample_3_1000-1050 data sets. On average, SA provides a 38% reduction in error from Fixed Width and provides a 68% improvement over Xcalibur for the two data sets for which Xcalibur's propriety data restrictions precluded measurement. Note the different scales.



(a) (b) (c)

Figure 10.9: Normalized True Peak Distance (NTPD) - Intensity. Here we compare predicted ridge intensities to the nearest observed ridge for each of the four methods (Statistical Agglomeration, Modified Bin Agglomeration, and Xcalibur) using the (a) Noyce, (b) Sample_3_750-800, and (c) Sample_3_1000-1050 data sets. SA outperforms the other methods on average, providing a 51% error reduction from Xcalibur for the two measurable data sets given Xcalibur's proprietary data restrictions. SA provides a 44% reduction on average over Fixed Width. Note the different scales.

for the Noyce data set then either of the other methods (see Figure 10.14). Fixed Width has a dramatically lower intensity SSE than either of the other methods on the Noyce data set, but only slightly less SSE than SA on the Sample_3_750-800 data set (see Figure 10.13). MBA noticably outperforms other methods on the Sample_3_1000-1050 data set.

While the above reported metrics should give an overall quantitative measure of the performance of each method, the segments of the spectra in Figures 10.15 and 10.16 provide a qualitative assessment of each method on the Sample 3 data sets (see Figure 10.17) . The pattern that emerges across data sets is that, at least on these random segments, SA consistly summarizes ridges exactly or very close to the hand annotation. MBA also performs well. Fixed Width is not consistent in performance but usually adds extra ridges and/or shifts m/z values of ridges substantially. Across both Sample 3 data sets, Xcalibur drastically increases the number of ridges in the segment. Xcalibur's predicted ridges are also notably less intense than the hand annotated data set.

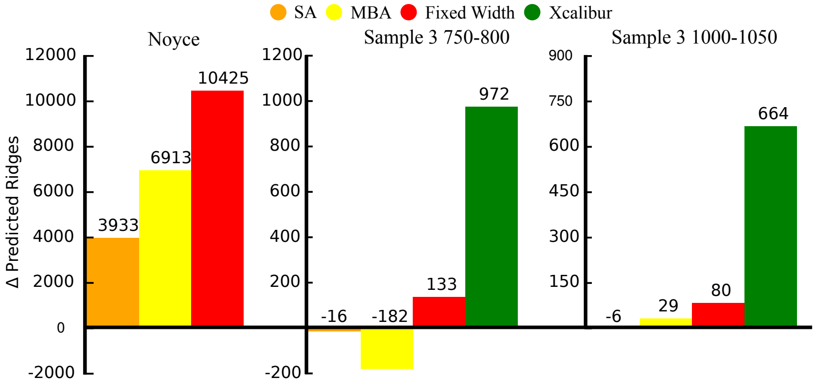


Figure 10.10: Δ Number of Ridges Predicted by Method. Each bar represents the difference from the actual number of ridges for each of the four methods (Statistical Agglomeration, Modified Bin Agglomeration, and Xcalibur) summed across all data sets. SA's number of predicted ridges is much closer to the observed number than any other method. Xcalibur predicted far more ridges than any other method. Because Xcalibur only accepts data in its proprietary format, the results are not available for the Noyce data set. Note the different scales.

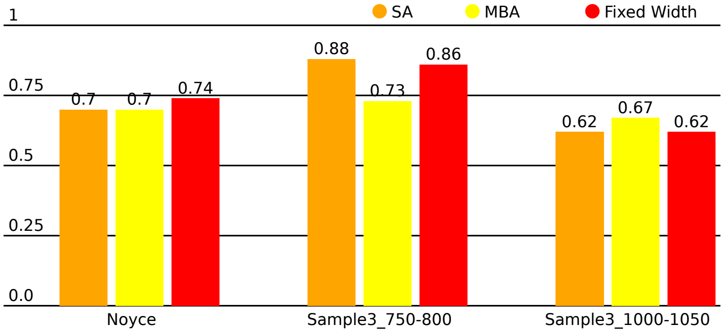


Figure 10.11: Purity. Purity measures the averaged homogeneity of each estimated ridge over all data points. There is no notable difference between methods on average. Since an inflated number of ridges increases purity, predicted vs. actual number of ridges suggests that MBA and Fixed Width purity values are due in part to overestimating the true number of ridges. Xcalibur does not provide sufficient detail about ridges' constituent points to measure purity.

10.5 Discussion

Fixed Width, to our knowledge the only extant algorithmic solution to this problem, is simple to code, yet has some obvious limitations. In mass spectrometry, the intra-sample resolution is inherently variable [89]. At least for the Orbitrap, low intensity signal groups are more dispersed while high intensity signal groups have less m/z variance. Any fixed width solution will either chop low intensity ridges into incorrect component ridges, incorrectly

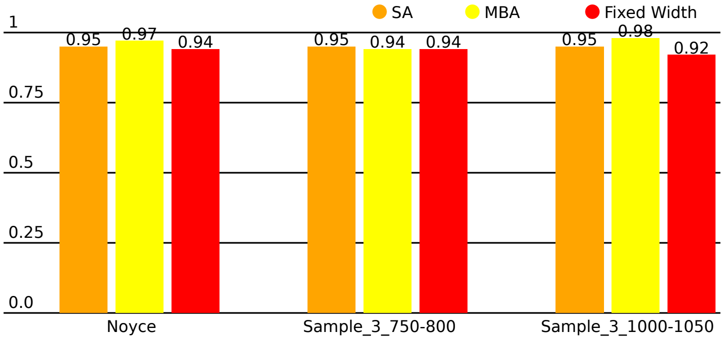


Figure 10.12: Normalized Mutual Information (NMI). NMI measures the information shared between the real ridge assignments and the predicted ridge assignments. Xcalibur does not provide sufficient detail about ridges' constituent points to measure NMI.

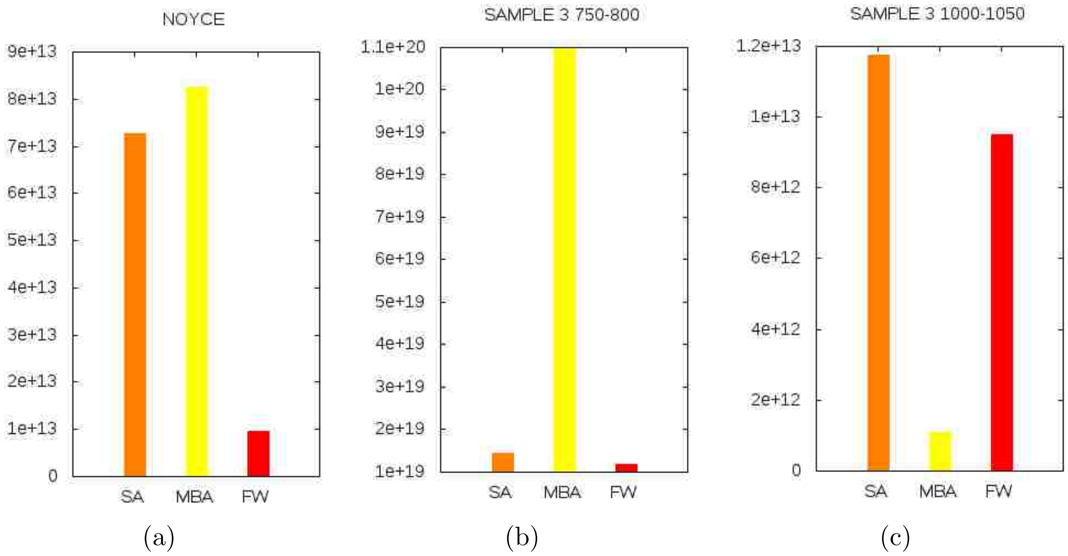
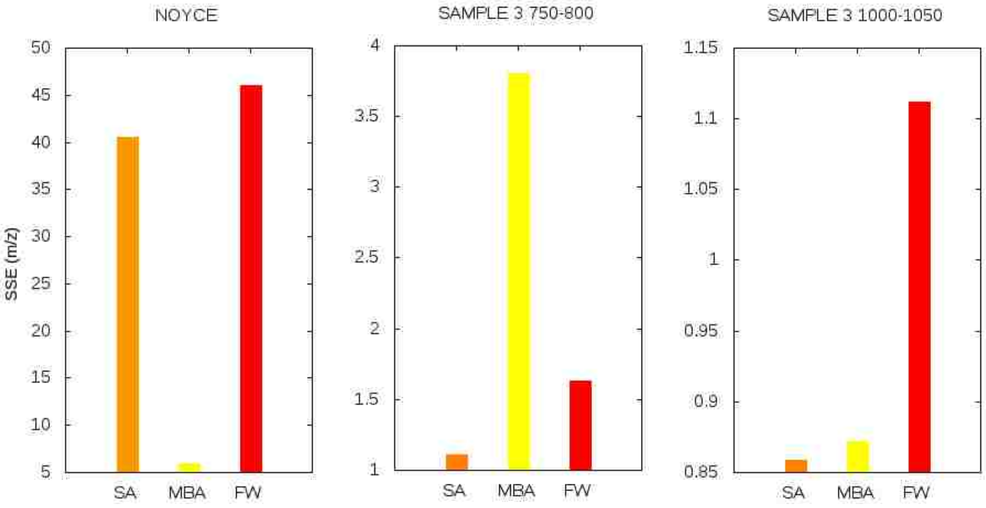


Figure 10.13: Sum Squared Error (SSE) - Intensity. Fixed Width has the lowest error on two of three data sets. This metric could not be measured for Xcalibur's ridge assignments. Note the different scales.

agglomerate high intensity ridges, or both. As shown in the results, fixed width methods significantly overestimate the number of ridges, cascading error downstream into identification and quantitation.

MBA attempts to provide robust means for dealing with ridges that overlap, and builds on the idea of Fixed Width binning by agglomerating any adjacent non-empty bins. Although the initial fixed width and the choice of which bin splitting options to use are parameters that must be determined and set by the operator, the information in manufacturer



(a) (b) (c)

Figure 10.14: Sum Squared Error (SSE) - m/z. The SSE of each of the four methods (Statistical Agglomeration, and Modified Bin Agglomeration) is measured for each of the 3 data sets (a) Noyce, (b) Sample_3_750-800, and (c) Sample_3_1000-1050 data sets. SA outperforms the other methods on Sample_3_750-800 and Sample_3_1000-1050, but MBA outperforms the other methods on the Noyce data set. SA's average error is 23% lower than Fixed Width. This metric could not be measured for Xcalibur's ridge assignments. Note the different scales.

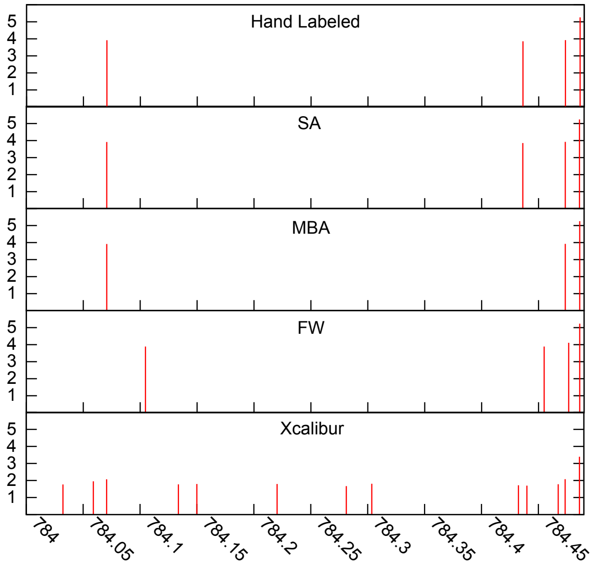


Figure 10.15: Peak Summarization of Sample 3: 784-785. Note: all intensities have been log-transformed for fit.

specifications, such as resolution, should assist in deciding the MBA parameters. In practice, the machine calibration to which the specifications are tied is not always the setup desired for

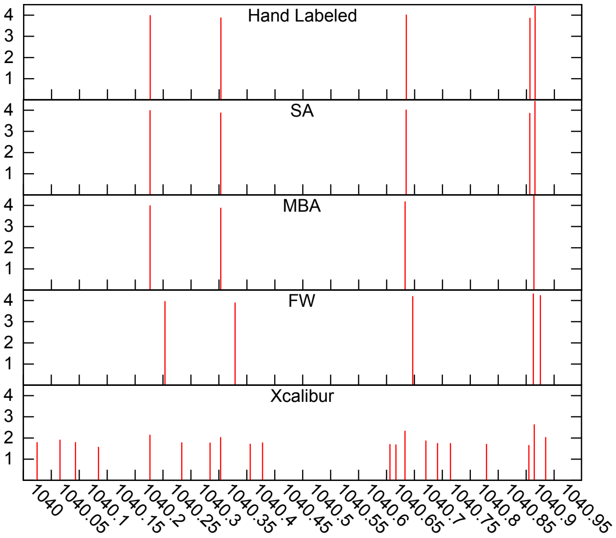


Figure 10.16: Peak Summarization of Sample 3: 1040-1041. Note: all intensities have been log-transformed for fit.

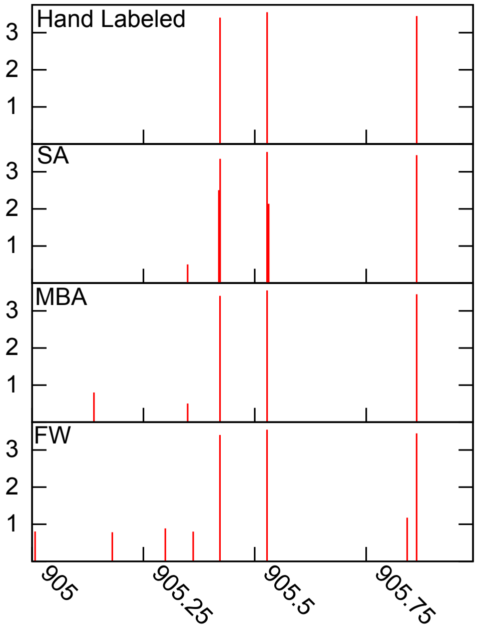


Figure 10.17: Peak Summarization of the Noyce Data Set. Note: all intensities have been log-transformed for fit. Xcalibur could not be compared due to proprietary data restrictions.

the practitioner due to time requirements, desire to use MS/MS, etc. Also, the true machine resolution can vary widely outside of the m/z value the specification is provided for. However, practical experience may assist in knowing when the manufacturer specs are sufficient and what changes need to be made when they are not.

Since each ridge can be a different width, SA addresses the problem of bin size in a flexible, data-driven manner. The ridge agglomeration procedure is statistically driven using the data itself, handling problems like overlapping ridges and avoiding the need for users to set parameters or for *apriori* knowledge about the data set. Noise filtering allows for the avoidance of boundary conditions found in fixed width methods such as ridges with just one data point. We consider s/n=1 to be a useful *apriori* setting, as it was the ideal setting across all three of our data sets. SA's ability to predict a far more accurate number of ridges than the other methods suggests it will increase accuracy in downstream processes over the current methods used, including Xcalibur (see Figure 10.10).

One troubling observation from this study is the difficulty in accurately assessing intensity of discovered ridges. Both species identification and quantitation require an accurate intensity measurement. Yet, even SA's performance is simply the best of several inaccurate methods. Given the amount of lipid quantitation performed currently, and also the state of the art, better methods of estimating intensity are needed.

We have described the need for accurate ridge summarization in direct injection lipidomics samples. Interestingly, despite the importance of accuracy in this first step of the analysis pipeline, there has been no study of solutions to this version of the ridge summarization problem to our knowledge. We present our estimate of what is currently done in the community, and also propose two novel algorithms, MBA and SA, for resolving ridges in shotgun lipidomics samples. We show that SA outperforms open source and proprietary methods on average in a measure of ridgewise error, NTPD, on three data sets. We also show that SA significantly outperforms the proprietary program Xcalibur on the two data sets for which we could use Xcalibur.

Incorporation of SA into existing analysis pipelines could drastically improve downstream quantitation and identification results in a variety of lipidomics experiments. Future work should continue improving our capacity to produce summarized ridges that more accurately estimate intensity. In light of the recent calls for greater reproducibility in mass spectrometry [123], and to foster development of improved algorithms, these data sets and the SA algorithm (with ample documentation) are available freely for non-commercial use at `http://github.com/optimusmoose/statistical_agglomeration`.

Chapter 11

Conclusion

This work attempts to improve the fundamental processes of mass spectrometery (MS) identification and quantification. We have focused our efforts on isotope trace detection, one module in the MS data processing pipeline, simulation, a general purpose tool for created MS labeled data, and increasing general scientific rigor in the field with meta-analysis of common practices.

In order to facilitate understanding and outside entry into the field, we provide the first ever tutorial covering the breadth of MS data processing (Chapter 1). Although tutorials exist for specific aspects of MS data processing, they are written for domain experts and not interdisciplinary contributors that are not familiar with the field. The paper presents enough detail to provide entry points for novel contributions while simultaneously providing enough background to facilitate a quick entry-level comprehension.

We attempt to catalyze a paradigm shift in the field by suggesting a modular approach to the quantitation and identification problem while stressing the importance of thorough evaluation (Chapters 3 and 4). The importance of this contribution is notable, as these two problems have contributed to the glut of papers that do not necessarily make a novel contribution in the field. This makes is very difficult for practitioners to establish a state-of-the-art for algorithm selection, and also perpetuates the lack of evaluation by making it difficult to find and test individual algorithms that may exist for a problem the theorist is trying to improve upon. Our evaluation paper was so well received that it was selected by the Faculty of 1000, an interdisciplinary post-peer-review indexing group that recommends manuscripts they consider to be important for the broader scientific audience.

In Chapter 5 we propose a novel unambiguous controlled vocabulary for MS data processing. Our nomenclature is unambiguous and provides coverage for concepts not described in the existing two controlled vocabularies or the colloquial vocabulary. Our nomenclature facilitates precise algorithm description, and is also very important for a mathematical specification of MS data.

In Chapter 6 we provide a formal characterization for the behavior of MS data. This characterization describes the data and suggests approaches to various problems, such as the segmentation of isotope traces from the whole set of signals in an MS output file. A formalization of the problem has never before been published, and has proven essential to our other contributions.

In order to facilitate evaluations, which are quite difficult because of the lack of labeled data intrinsic to MS, propose two MS simulators. The mspire-simulator was designed to improve upon the two existing MS simulators (Chapter 7), both of which failed to implement the known behavior we characterized with our mathematical description of LC-MS data (Chapter 6). As we showed in Figure 7.1F, our models for simulating the variance in real isotope traces are qualitatively superior to ideal isotope traces without variance (Figure 7.1B), such as those in previous simulators. This variance model is essential to achieving fidelity with real data. The simulator achieves quantitative fidelity with real data, as shown in Figure 7.3.

Although mspire-simulator creates more accurate simulations than existing simulators, it has serious performance limitations due to the language of implementation. JAMSS implements the models in mspire-simulator, but improves upon the programmatic and user interface aspects (see Chapter 8). Simulating MS data is computationally intensive. It is also a complicated process that precludes normal multi-threading. JAMSS features an innovative workflow consisting of several phases that are separately processed, as well as several custom optimizations to provide speed-up without omitting 50% of the proteins in the sample as does MSSimulator, a previously published simulator. It also features a GUI with simplified user parameters to make it easier to use. It is packaged as a JAR file and designed to omit any external libraries, allowing a one click install, compared to many external package dependencies of the three other published simulators.

Using labeled data and applying our mathematical characterization, we create an advanced isotope trace extraction algorithm, Massifquant, that outperforms popular existing algorithms (Chapter 9). Massifquant dramatically outperforms the state of the art algorithms we compared it against. Additionally, the same process of mathematical characterization and innovation has yielded a novel isotope trace extraction algorithm for a different MS experiment type that outperforms state-of-the-art algorithms (Chapter 10).

11.1 Future Work

We have created a body of work that suggests a paradigm shift towards greater modular focus in the MS data processing pipeline. However, there are still significant future contributions that can and should be made.

Though our evaluative comparison of Massifquant included the most widely used existing algorithms, a benchmark evaluation with a broader range of datasets and existing algorithms is needed to define which algorithms practitioners should consider and which perform poorly enough that they ought to be abandoned.

Isotope trace feature detection is the first step towards an MS1 pipeline for identification and quantification. Our algorithm is the prerequisite step for an isotopic envelope extraction algorithm. The process of creating a benchmark, testing existing algorithms, studying their drawbacks, and developing a new algorithm must be repeated to do for isotopic envelope extraction what we've done for isotope trace detection.

This process will be slightly more straightforward due to our contribution of the JAMSS and mspire simulators, as well as our mathematical characterization of both isotope traces and isotopic envelopes. The JAMSS simulator is the most realistic MS simulator available. However, due to the lack of labeled data, its models are limited in fidelity compared to real data. In the future, algorithms that facilitate more observations on real data, such as Massifquant, will inform the models in JAMSS and allow a bootstrap process to create more realistic simulations. These, in turn, can produce more realistic evaluations on existing algorithms. Thus, our work has not only produced an advanced isotope trace detection algorithm, as well as a mathematical characterization of both isotope traces and isotopic envelopes, but also developed a framework for attacking these problems that will be just as useful for other modules in the MS identification and quantitation pipeline.

References

[1] Aberg, K. M., Torgrip, R. J., Kolmert, J., Schuppe-Koistinen, I. and Lindberg, J. (2008) Feature detection and alignment of hyphenated chromatographicmass spectrometric data: Extraction of pure ion chromatograms using kalman tracking. *Journal of Chromatography A*, **1192**, 139 – 146. URL `http://www.sciencedirect.com/science/article/pii/S0021967308005025`.

[2] Adam, B.-L., Qu, Y., Davis, J. W., Ward, M. D., Clements, M. A., Cazares, L. H., Semmes, O. J., Schellhammer, P. F., Yasui, Y., Feng, Z. *et al.* (2002) Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men. *Cancer Research*, **62**, 3609–3614.

[3] Annesley, T. M. (2003) Ion suppression in mass spectrometry. *Clinical Chemistry*, **49**, 1041–1044.

[4] Arnold, R. J., Jayasankar, N., Aggarwal, D., Tang, H. and Radivojac, P. (2006) A machine learning approach to predicting peptide fragmentation spectra. In *Pacific Symposium on Biocomputing*, volume 11, pp. 219–230.

[5] Babushok, V. I. and Zenkevich, I. G. (2010) Retention characteristics of peptides in RP-LC: Peptide retention prediction. *Chromatographia*, **72**, 781–797.

[6] Ballardini, R., Benevento, M., Arrigoni, G., Pattini, L. and Roda, A. (2011) MassUntangler: A novel alignment tool for label-free liquid chromatography–mass spectrometry proteomic data. *Journal of Chromatography A*, **1218**, 8859–8868.

[7] Bellew, M., Coram, M., Fitzgibbon, M., Igra, M., Randolph, T., Wang, P., May, D., Eng, J., Fang, R., Lin, C. *et al.* (2006) A suite of algorithms for the comprehensive analysis of complex protein mixtures using high-resolution LC-MS. *Bioinformatics*, **22**, 1902–1909.

[8] Bickel, P. J., Hammel, E. A. and O'Connell, J. W. (1975) Sex bias in graduate admissions: Data from berkeley. *Science*, **187**, 398–404. URL `http://www.sciencemag.org/content/187/4175/398.abstract`.

[9] Bielow, C., Aiche, S., Andreotti, S. and Reinert, K. (2011) MSSimulator: Simulation of mass spectrometry data. *Journal of Proteome Research*, **10**, 2922–2929.

[10] Biemann, K. (1992) Mass spectrometry of peptides and proteins. *Annual Review of Biochemistry*, **61**, 977–1010.

[11] Böcker, S. and Kaltenbach, H.-M. (2007) Mass spectra alignments and their significance. *Journal of Discrete Algorithms*, **5**, 714–728. URL `http://www.sciencedirect.com/science/article/pii/S1570866706001043`.

[12] Braisted, J. C., Kuntumalla, S., Vogel, C., Marcotte, E. M., Rodrigues, A. R., Wang, R., Huang, S.-T., Ferlanti, E. S., Saeed, A. I., Fleischmann, R. D. *et al.* (2008) The APEX quantitative proteomics tool: generating protein quantitation estimates from LC-MS/MS proteomics results. *BMC bioinformatics*, **9**, 529.

[13] Brusniak, M.-Y., Bodenmiller, B., Campbell, D., Cooke, K., Eddes, J., Garbutt, A., Lau, H., Letarte, S., Mueller, L., Sharma, V. *et al.* (2008) Corra: Computational framework and tools for LC-MS discovery and targeted mass spectrometry-based proteomics. *BMC bioinformatics*, **9**, 542.

[14] Cappadona, S., Baker, P., Cutillas, P., Heck, A. and van Breukelen, B. (2012) Current challenges in software solutions for mass spectrometry-based quantitative proteomics. *Amino Acids*, **43**, 1–22.

[15] Cappadona, S., Muñoz, J., Spee, W. P., Low, T. Y., Mohammed, S., van Breukelen, B. and Heck, A. J. (2011) Deconvolution of overlapping isotopic clusters improves quantification of stable isotope–labeled peptides. *Journal of Proteomics*, **74**, 2204–2209.

[16] Choi, H., Fermin, D. and Nesvizhskii, A. I. (2008) Significance analysis of spectral count data in label-free shotgun proteomics. *Mol Cell Proteomics*, **7**, 2373–2385. URL `http://www.ncbi.nlm.nih.gov/pubmed/18644780`.

[17] Chong, K. F. and Leong, H. W. (2012) Tutorial on *de novo* peptide sequencing using MS/MS mass spectrometry. *Journal of Bioinformatics and Computational Biology*, **10**.

[18] Christin, C., Hoefsloot, H., Smilde, A., Suits, F., Bischoff, R. and Horvatovich, P. (2010) Time alignment algorithms based on selected mass traces for complex LC-MS data. *Journal of proteome research*, **9**, 1483–1495.

[19] Christin, C., Smilde, A., Hoefsloot, H., Suits, F., Bischoff, R. and Horvatovich, P. (2008) Optimized Time Alignment Algorithm for LC- MS Data: Correlation Optimized Warping Using Component Detection Algorithm-Selected Mass Chromatograms. *Analytical Chemistry*, **80**, 7012–7021.

[20] Cole, R. B. (ed.) (1997) *Electrospray Ionization Mass Spectrometry: Fundamentals, Instrumentation, and Applications.* Wiley-Interscience, New York.

[21] Conley, C., Smith, R., Torgrip, R. J. O., Taylor, R. M., Tautenhahn, R. and Prince, J. T. (in review) Massifquant: Open-source Kalman filter based XC-MS feature detection. *Bioinformatics.*

[22] Cox, J. and Mann, M. (2008) MaxQuant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification. *Nature Biotechnology*, **26**, 1367–1372.

[23] Dakna, M., He, Z., Yu, W. C., Mischak, H., Kolch, W. *et al.* (2009) Technical, bioinformatical and statistical aspects of liquid chromatography/mass spectrometry (LC-MS) and capillary electrophoresis-mass spectrometry (CE-MS) based clinical proteomics: A critical assessment. *Journal of Chromatography B*, **877**, 1250–1258.

[24] Deutsch, E. (2008) mzML: A single, unifying data format for mass spectrometer output. *PROTEOMICS*, **8**, 2776–2777. URL `http://dx.doi.org/10.1002/pmic.200890049`.

[25] Dixon, S. J., Brereton, R. G., Soini, H. A., Novotny, M. V. and Penn, D. J. (2006) An automated method for peak detection and matching in large gas chromatography-mass spectrometry data sets. *Journal of Chemometrics*, **20**, 325–340.

[26] Domon, B. and Aebersold, R. (2006) Mass spectrometry and protein analysis. *Science Signalling*, **312**, 212.

[27] Du, P. and Angeletti, R. H. (2006) Automatic deconvolution of isotope-resolved mass spectra using variable selection and quantized peptide mass distribution. *Analytical Chemistry*, **78**, 3385–3392.

[28] Egertson, J., Eng, J., Bereman, M., Hsieh, E., Merrihew, G. and MacCoss, M. (2012) De Novo Correction of Mass Measurement Error in Low Resolution Tandem MS Spectra for Shotgun Proteomics. *Journal of The American Society for Mass Spectrometry*, pp. 1–8.

[29] Ejsing, C. S., Duchoslav, E., Sampaio, J., Simons, K., Bonner, R., Thiele, C., Ekroos, K. and Shevchenko, A. (2006) Automated Identification and Quantification of Glycerophospholipid Molecular Species by Multiple Precursor Ion Scanning. *Analytical Chemistry*, **78**, 6202–6214. URL `http://pubs.acs.org/doi/abs/10.1021/ac060545x`.

[30] Ekroos, K., Chernushevich, I. V., Simons, K. and Shevchenko, A. (2002) Quantitative Profiling of Phospholipids by Multiple Precursor Ion Scanning on a Hybrid Quadrupole Time-of-Flight Mass Spectrometer. *Analytical Chemistry*, **74**, 941–949. URL `http://pubs.acs.org/doi/abs/10.1021/ac015655c`.

[31] Elias, J. E., Gibbons, F. D., King, O. D., Roth, F. P. and Gygi, S. P. (2004) Intensity-based protein identification by machine learning from a library of tandem mass spectra. *Nature Biotechnology*, **22**, 214–219.

[32] Eng, J. K., McCormack, A. L. and Yates Iii, J. R. (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, **5**, 976–989.

[33] Fahy, E., Subramaniam, S., Murphy, R. C., Nishijima, M., Raetz, C. R., Shimizu, T., Spener, F., van Meer, G., Wakelam, M. J. and Dennis, E. A. (2009) Update of the LIPID MAPS comprehensive classification system for lipids. *Journal of Lipid Research*, **50**, S9–S14.

[34] Feng, L. and Prestwich, G. (2005) *Functional Lipidomics*. Taylor & Francis. URL `http://books.google.com/books?id=x_2uWPRWl0cC`.

[35] Fiehn, O. (2002) Metabolomics–the link between genotypes and phenotypes. *Plant Molecular Biology*, **48**, 155–171.

[36] Fischer, B., Grossmann, J., Roth, V., Gruissem, W., Baginsky, S. and Buhmann, J. (2006) Semi-supervised LC/MS alignment for differential proteomics. *Bioinformatics*, **22**, e132–e140.

[37] Frank, A. and Pevzner, P. (2005) PepNovo: de novo peptide sequencing via probabilistic network modeling. *Analytical Chemistry*, **77**, 964–973.

[38] Frank, A. M., Bandeira, N., Shen, Z., Tanner, S., Briggs, S. P., Smith, R. D. and Pevzner, P. A. (2008) Clustering Millions of Tandem Mass Spectra. *Journal of Proteome Research*, **7**, 113–122. URL `http://pubs.acs.org/doi/abs/10.1021/pr070361e`.

[39] Frank, E., Hall, M., Trigg, L., Holmes, G. and Witten, I. H. (2004) Data mining in bioinformatics using Weka. *Bioinformatics*, **20**, 2479–2481.

[40] Gentleman, R. C., Carey, V. J., Bates, D. M. and others (2004) Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, **5**, R80. URL `http://genomebiology.com/2004/5/10/R80`.

[41] German, J. B., Gillies, L. A., Smilowitz, J. T., Zivkovic, A. M. and Watkins, S. M. (2007) Lipidomics and lipid profiling in metabolomics. *Current opinion in lipidology*, **18**, 66–71.

[42] Good, B. M. and Su, A. I. (2013) Crowdsourcing for bioinformatics. *Bioinformatics*, **29**, 1925–1933. URL `http://bioinformatics.oxfordjournals.org/content/early/2013/07/03/bioinformatics.btt333.abstract`.

[43] Griffiths, W. J. and Wang, Y. (2009) Mass spectrometry: from proteomics to metabolomics and lipidomics. *Chem. Soc. Rev.*, **38**, 1882–1896. URL `http://dx.doi.org/10.1039/B618553N`.

[44] Han, X. and Gross, R. W. (1994) Electrospray Ionization Mass Spectroscopic Analysis of Human Erythrocyte Plasma Membrane Phospholipids. *Proceedings of the National Academy of Sciences of the United States of America*, **91**, pp. 10635–10639. URL `http://www.jstor.org/stable/2366084`.

[45] Han, X. and Gross, R. W. (2005) Shotgun lipidomics: Electrospray ionization mass spectrometric analysis and quantitation of cellular lipidomes directly from crude extracts of biological samples. *Mass Spectrometry Reviews*, **24**, 367–412. URL `http://dx.doi.org/10.1002/mas.20023`.

[46] Hemminger, B. M., Losi, T. and Bauers, A. (2005) Survey of bioinformatics programs in the United States. *Journal of the American Society for Information Science and Technology*, **56**, 529–537.

[47] Herzog, R., Schwudke, D., Schuhmann, K., Sampaio, J., Bornstein, S., Schroeder, M. and Shevchenko, A. (2011) A novel informatics concept for high-throughput shotgun lipidomics based on the molecular fragmentation query language. *Genome Biology*, **12**, 1–25. URL `http://dx.doi.org/10.1186/gb-2011-12-1-r8`. 10.1186/gb-2011-12-1-r8.

[48] Holmes, G., Hall, M. and Prank, E. (1999) *Generating rule sets from model trees.* Springer.

[49] Jeffries, N. (2005) Algorithms for alignment of mass spectrometry proteomic data. *Bioinformatics*, **21**, 3066–3073. URL http://bioinformatics.oxfordjournals.org/content/21/14/3066.abstract.

[50] Kessner, D., Chambers, M., Burke, R., Agus, D. and Mallick, P. (2008) ProteoWizard: open source software for rapid proteomics tools development. *Bioinformatics*, **24**, 2534–2536. URL http://bioinformatics.oxfordjournals.org/content/24/21/2534.abstract.

[51] Köfeler, H. C., Fauland, A., Rechberger, G. N. and Trötzmüller, M. (2012) Mass Spectrometry Based Lipidomics: An Overview of Technological Platforms. *Metabolites*, **2**, 19–38. URL http://www.mdpi.com/2218-1989/2/1/19/.

[52] Kohlbacher, O., Reinert, K., Gröpl, C., Lange, E., Pfeifer, N., Schulz-Trieglaff, O. and Sturm, M. (2007) TOPP—the OpenMS proteomics pipeline. *Bioinformatics*, **23**, e191–e197.

[53] Kraegen, E., Cooney, G., Ye, J., Thompson, A. and Furler, S. (2001) The Role of Lipids in the Pathogenesis of Muscle Insulin Resistance and Beta Cell Faiture in Type II Diabetes and Obesity. *Experimental and Clinical Endocrinology & Diabetes*, **109**, S189–S201.

[54] Lange, E. (2006) High-Accuracy Peak Picking of Proteomics Data Using Wavelet Techniques Eva Lange, Clemens Gropl, Knut Reinert, Oliver Kohlbacher, and Andreas Hildebrandt Pacific Symposium on Biocomputing 11: 243-254 (2006). In *Pacific Symposium on Biocomputing*, volume 11, pp. 243–254.

[55] Lange, E., Tautenhahn, R., Neumann, S. and Gröpl, C. (2008) Critical assessment of alignment procedures for LC-MS proteomics and metabolomics measurements. *BMC Bioinformatics*, **9**, 375.

[56] Li, X.-j., Eugene, C. Y., Kemp, C. J., Zhang, H. and Aebersold, R. (2005) A software suite for the generation and comparison of peptide arrays from sets of data collected by liquid chromatography-mass spectrometry. *Molecular & Cellular Proteomics*, **4**, 1328–1340.

[57] Li, X.-J., Zhang, H., Ranish, J. A. and Aebersold, R. (2003) Automated statistical analysis of protein abundance ratios from data generated by stable-isotope dilution and tandem mass spectrometry. *Analytical chemistry*, **75**, 6648–6657.

[58] Listgarten, J., Neal, R., Roweis, S., Wong, P. and Emili, A. (2007) Difference detection in LC-MS data for protein biomarker discovery. *Bioinformatics*, **23**, e198–e204.

[59] Liu, H., Sadygov, R. and Yates III, J. (2004) A model for random sampling and estimation of relative protein abundance in shotgun proteomics. *Analytical chemistry*, **76**, 4193–4201.

[60] Liu, J., Bell, A., Bergeron, J., Yanofsky, C., Carrillo, B., Beaudrie, C. and Kearney, R. (2007) Methods for peptide identification by spectral comparison. *Proteome Science*, **5**, 1–12. URL `http://dx.doi.org/10.1186/1477-5956-5-3`. 10.1186/1477-5956-5-3.

[61] Lundgren, D. H., Hwang, S.-I., Wu, L. and Han, D. K. (2010) Role of spectral counting in quantitative proteomics. *Expert review of proteomics*, **7**, 39–53.

[62] MacLean, B., Tomazela, D. M., Shulman, N., Chambers, M., Finney, G. L., Frewen, B., Kern, R., Tabb, D. L., Liebler, D. C. and MacCoss, M. J. (2010) Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics*, **26**, 966–968.

[63] Mallick, P., Schirle, M., Chen, S., Flory, M., Lee, H., Martin, D., Ranish, J., Raught, B., Schmitt, R., Werner, T., Kuster, B. and Aebersold, R. (2007) Computational prediction of proteotypic peptides for quantitative proteomics. *Nature Biotechnology*, **25**, 125–131.

[64] Michalski, A., Cox, J. and Mann, M. (2011) More than 100,000 Detectable Peptide Species Elute in Single Shotgun Proteomics Runs but the Marjority is Inaccessible to Data-Dependent LC-MS/MS. *Journal of Proteome Research*, **10**, 1785–1793.

[65] Mischak, H., Coon, J. J., Novak, J., Weissinger, E. M., Schanstra, J. P. and Dominiczak, A. F. (2009) Capillary electrophoresis–mass spectrometry as a powerful tool in biomarker discovery and clinical diagnosis: an update of recent developments. *Mass Spectrometry Reviews*, **28**, 703–724.

[66] Morris, M. and Watkins, S. M. (2005) Focused metabolomic profiling in the drug development process: advances from lipid profiling. *Current Opinion in Chemical Biology*, **9**, 407–412. URL `http://www.sciencedirect.com/science/article/pii/S1367593105000797`.

[67] Mueller, L., Rinner, O., Schmidt, A., Letarte, S., Bodenmiller, B., Brusniak, M., Vitek, O., Aebersold, R. and Müller, M. (2007) SuperHirn–a novel tool for high resolution LC-MS-based peptide/protein profiling. *Proteomics*, **7**, 3470–3480.

[68] Mueller, L. N., Brusniak, M.-Y., Mani, D. R. and Aebersold, R. (2008) An Assessment of Software Solutions for the Analysis of Mass Spectrometry Based Quantitative Proteomics Data. *Journal of Proteome Research*, **7**, 51–61. URL `http://pubs.acs.org/doi/abs/10.1021/pr700758r`.

[69] Murray, K. K., Boyd, R. K., Eberlin, M. N., Langley, G. J., Li, L., Naito, Y. *et al.* (2013) Definitions of terms relating to mass spectrometry (IUPAC recommendations 2013). *Pure and Applied Chemistry*, pp. None–None.

[70] Noyce, A. B., Smith, R., Dalgliesh, J., Taylor, R. M., Erb, K., Okuda, N. and Prince, J. T. (2013) Mspire-Simulator: LC-MS Shotgun Proteomic Simulator for Creating Realistic Gold Standard Data. *Journal of Proteome Research*.

[71] Orešič, M. (2009) Bioinformatics and computational approaches applicable to lipidomics. *European Journal of Lipid Science and Technology*, **111**, 99–106. URL `http://dx.doi.org/10.1002/ejlt.200800144`.

[72] Pedrioli, P. G., Eng, J. K., Hubley, R., Vogelzang, M., Deutsch, E. W., Raught, B., Pratt, B., Nilsson, E., Angeletti, R. H., Apweiler, R., Cheung, K., Costello, C. E., Hermjakob, H., Huang, S., Julian, R. K., Kapp, E., McComb, M. E., Oliver, S. G., Omenn, G., Paton, N. W., Simpson, R., Smith, R., Taylor, C. F., Zhu, W. and Aebersold, R. (2004) A common open representation of mass spectrometry data and its application to proteomics research. *Nat Biotechnol*, **22**, 1459–1466. URL `http://www.ncbi.nlm.nih.gov/pubmed/15529173`.

[73] Pluskal, T., Castillo, S., Villar-Briones, A. and Oresic, M. (2010) MZmine 2: Modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics*, **11**, 395. URL `http://www.biomedcentral.com/1471-2105/11/395`.

[74] Podwojski, K., Fritsch, A., Chamrad, D., Paul, W., Sitek, B., Stühler, K., Mutzel, P., Stephan, C., Meyer, H., Urfer, W. *et al.* (2009) Retention time alignment algorithms for LC/MS data must consider non-linear shifts. *Bioinformatics*, **25**, 758–764.

[75] Prakash, A., Tomazela, D. M., Frewen, B., MacLean, B., Merrihew, G., Peterman, S. and MacCoss, M. J. (2009) Expediting the development of targeted srm assays: using data from shotgun proteomics to automate method development. *Journal of Proteome Research*, **8**, 2733–2739.

[76] Prince, J. T. and Marcotte, E. M. (2008) mspire: mass spectrometry proteomics in Ruby. *Bioinformatics*, **24**, 2796–2797.

[77] Qu, Y., Adam, B.-L., Yasui, Y., Ward, M. D., Cazares, L. H., Schellhammer, P. F., Feng, Z., Semmes, O. J. and Wright, G. L. (2002) Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from noncancer patients. *Clinical Chemistry*, **48**, 1835–1843.

[78] Rockwood, A. L. and Van Orden, S. L. (1996) Ultrahigh-speed calculation of isotope distributions. *Analytical chemistry*, **68**, 2027–2030.

[79] Rodriguez, J., Gupta, N., Smith, R. D. and Pevzner, P. A. (2007) Does trypsin cut before proline? *Journal of proteome research*, **7**, 300–305.

[80] Rohn, H., Junker, A., Hartmann, A., Grafahrend-Belau, E., Treutler, H., Klapperstück, M., Czauderna, T., Klukas, C. and Schreiber, F. (2012) VANTED v2: a framework for systems biology applications. *BMC systems biology*, **6**, 139.

[81] Ross, P. L., Huang, Y. N., Marchese, J. N., Williamson, B., Parker, K., Hattan, S., Khainovski, N., Pillai, S., Dey, S., Daniels, S. *et al.* (2004) Multiplexed protein quantitation in saccharomyces cerevisiae using amine-reactive isobaric tagging reagents. *Molecular & cellular proteomics*, **3**, 1154–1169.

[82] Saghatelian, A., Trauger, S. A., Want, E. J., Hawkins, E. G., Siuzdak, G. and Cravatt, B. F. (2004) Assignment of endogenous substrates to enzymes by global metabolite profiling. *Biochemistry*, **43**, 14332–14339. URL http://pubs.acs.org/doi/abs/10.1021/bi0480335. PMID: 15533037.

[83] Saito, K., Koizumi, E. and Koizumi, H. (2012) Application of parallel hybrid algorithm in massively parallel gpgputhe improved effective and efficient method for calculating coulombic interactions in simulations of many ions with simion. *Journal of The American Society for Mass Spectrometry*, **23**, 1609–1615.

[84] Samuelsson, J., Dalevi, D., Levander, F. and Rögnvaldsson, T. (2004) Modular, scriptable and automated analysis tools for high-throughput peptide mass fingerprinting. *Bioinformatics*, **20**, 3628–3635. URL http://bioinformatics.oxfordjournals.org/content/20/18/3628.abstract.

[85] Satterthwaite, F. E. (1946) An Approximate Distribution of Estimates of Variance Components. *Biometrics Bulletin*, **2**, pp. 110–114. URL http://www.jstor.org/stable/3002019.

[86] Schmelzer, K., Fahy, E., Subramaniam, S. and Dennis, E. A. (2007) The Lipid Maps Initiative in Lipidomics. In Brown, H. A. (ed.), *Lipidomics and Bioactive Lipids: MassSpectrometry–Based Lipid Analysis*, volume 432 of *Methods in Enzymology*, pp. 171–183. Academic Press. URL `http://www.sciencedirect.com/science/article/pii/S0076687907320077`.

[87] Schulz-Trieglaff, O., Pfeifer, N., Gröpl, C., Kohlbacher, O. and Reinert, K. (2008) LC-MSsim–a simulation software for liquid chromatography mass spectrometry data. *BMC Bioinformatics*, **9**, 423.

[88] Schwudke, D., Oegema, J., Burton, L., Entchev, E., Hannich, J. T., Ejsing, C. S., Kurzchalia, T. and Shevchenko, A. (2006) Lipid Profiling by Multiple Precursor and Neutral Loss Scanning Driven by the Data-Dependent Acquisition. *Analytical Chemistry*, **78**, 585–595. URL `http://pubs.acs.org/doi/abs/10.1021/ac051605m`. PMID: 16408944.

[89] Schwudke, D., Schuhmann, K., Herzog, R., Bornstein, S. R. and Shevchenko, A. (2011) Shotgun Lipidomics on High Resolution Mass Spectrometers. *Cold Spring Harbor Perspectives in Biology*, **3**.

[90] Smith, C., Elizabeth, J., O'Maille, G., Abagyan, R. and Siuzdak, G. (2006) XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Analytical chemistry*, **78**, 779–787.

[91] Smith, C. A., O'Maille, G., Want, E. J., Qin, C., Trauger, S. A., Brandon, T. R., Custodio, D. E., Abagyan, R. and Siuzdak, G. (2005) METLIN: a metabolite mass spectral database. *Therapeutic Drug Monitoring*, **27**, 747–751.

[92] Smith, R., Anthonymuthu, T. S., Ventura, D. and Prince, J. T. (2012) Statistical Agglomeration: Peak Summarization for Direct Infusion Lipidomics. *Bioinformatics*, **29**, 2445–2451.

[93] Smith, R., Mathis, A. D., Ventura, D. and Prince, J. T. (????) Proteomics, Lipidomics, Metabolomics: A Mass Spectrometry Tutorial from a Computer Scientist's Point of View. *BMC Bioinformatics*, **in review**.

[94] Smith, R. and Prince, J. T. (in review) JAMSS: Proteomics mass spectrometry simulation in java. *Bioinformatics*.

[95] Smith, R., Taylor, R. M. and Prince, J. T. (2014) Clarity in Concepts: A Novel, Unambiguous Nomenclature for MS-omics Data Structures. *Proteomics*, **(in review)**.

[96] Smith, R., Ventura, D. and Prince, J. T. (2013) Controlling for Confounding Variables in MS-omics Protocol: Why Modularity Matters. *Briefings in Bioinformatics*.

[97] Smith, R., Ventura, D. and Prince, J. T. (2013) LC-MS Alignment in Theory and Practice: A Comprehensive Algorithmic Review. *Briefings in Bioinformatics*.

[98] Smith, R., Ventura, D. and Prince, J. T. (2013) Novel algorithms and the benefits of comparative validation. *Bioinformatics*, **29**, 1583–1585.

[99] Song, H., Hsu, F.-F., Ladenson, J. and Turk, J. (2007) Algorithm for Processing Raw Mass Spectrometric Data to Identify and Quantitate Complex Lipid Molecular Species in Mixtures by Data-Dependent Scanning and Fragment Ion Database Searching. *Journal of the American Society for Mass Spectrometry*, **18**, 1848–1858. URL `http://www.sciencedirect.com/science/article/pii/S1044030507006319`.

[100] Sturm, M., Bertsch, A., Gropl, C., Hildebrandt, A., Hussong, R., Lange, E., Pfeifer, N., Schulz-Trieglaff, O., Zerck, A., Reinert, K. and Kohlbacher, O. (2008) OpenMS - An open-source software framework for mass spectrometry. *BMC Bioinformatics*, **9**, 163. URL `http://www.biomedcentral.com/1471-2105/9/163`.

[101] Sturm, M. and Kohlbacher, O. (2009) TOPPView: An Open-Source Viewer for Mass Spectrometry Data. *Journal of Proteome Research*, **8**, 3760–3763.

[102] Sugimoto, M., Kawakami, M., Robert, M., Soga, T. and Tomita, M. (2012) Bioinformatics Tools for Mass Spectroscopy-Based Metabolomic Data Processing and Analysis. *Current Bioinformatics*, **7**, 96.

[103] Tabb, D. L., Shah, M. B., Strader, M. B., Connelly, H. M., Hettich, R. L. and Hurst, G. B. (2006) Determination of peptide and protein ion charge states by Fourier transformation of isotope-resolved mass spectra. *Journal of the American Society for Mass Spectrometry*, **17**, 903–915.

[104] Tautenhahn, R., Bottcher, C. and Neumann, S. (2008) Highly sensitive feature detection for high resolution LC/MS. *BMC Bioinformatics*, **9**, 504. URL `http://www.biomedcentral.com/1471-2105/9/504`.

[105] Taylor, C. F., Hermjakob, H., Julian Jr, R. K., Garavelli, J. S., Aebersold, R. and Apweiler, R. (2006) The work of the human proteome organisation's proteomics standards initiative (HUPO PSI). *Omics: a Journal of Integrative Biology*, **10**, 145–151.

[106] Tengstrand, E., Lindberg, J. and Aberg, K. M. (2014) Tracmass 2—a modular suite of tools for processing chromatography-full scan mass spectrometry data. *Analytical Chemistry*, **86**, 3435–3442. URL `http://pubs.acs.org/doi/abs/10.1021/ac403905h`.

[107] Tsai, T.-H., Tadesse, M. G., Di Poto, C., Pannell, L. K., Mechref, Y., Wang, Y. and Ressom, H. W. (2013) Multi-profile bayesian alignment model for lc-ms data analysis with integration of internal standards. *Bioinformatics*, **29**, 2774–2780.

[108] Tseng, Y.-H., Uetrecht, C., Yang, S.-C., Barendregt, A., Heck, A. J. and Peng, W.-P. (2013) A game theory-based search engine to automate the mass assignment in complex native electrospray mass spectra. *Analytical Chemistry*.

[109] Van Nederkassel, A., Daszykowski, M., Eilers, P. and Heyden, Y. (2006) A comparison of three algorithms for chromatograms alignment. *Journal of Chromatography A*, **1118**, 199–210.

[110] Wang, P., Tang, H., Fitzgibbon, M., Mcintosh, M., Coram, M., Zhang, H., Yi, E. and Aebersold, R. (2007) A statistical method for chromatographic alignment of LC-MS data. *Biostatistics*, **8**, 357–367.

[111] Wang, W., Zhou, H., Lin, H., Roy, S., Shaler, T. A., Hill, L. R., Norton, S., Kumar, P., Anderle, M. and Becker, C. H. (2003) Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards. *Analytical Chemistry*, **75**, 4818–4826.

[112] Wang, Y. and Witten, I. H. (1997) Inducing model trees for continuous classes. In *Proceedings of the Ninth European Conference on Machine Learning*, pp. 128–137.

[113] Watson, A. D. (2006) Thematic review series: Systems Biology Approaches to Metabolic and Cardiovascular Disorders. Lipidomics: a global approach to lipid analysis in biological systems. *Journal of Lipid Research*, **47**, 2101–2111.

[114] Wehofsky, M. and Hoffmann, R. (2002) Automated deconvolution and deisotoping of electrospray mass spectra. *Journal of Mass Spectrometry*, **37**, 223–229.

[115] Weisser, H., Nahnsen, S., Grossmann, J., Nilse, L., Quandt, A., Brauer, H., Sturm, M., Kenar, E., Kohlbacher, O., Aebersold, R. *et al.* (2013) An automated pipeline for high-throughput label-free quantitative proteomics. *Journal of Proteome Research*.

[116] Welch, B. L. (1947) The generalization of 'Student's' problem when several different population variances are involved. *Biometrika*, **34**, 28–35. URL `http://biomet.oxfordjournals.org/content/34/1-2/28.short`.

[117] Welch, G. and Bishop, G. (2006) An introduction to the Kalman filter. *UNC-Chapel Hill, TR 95-041, http://www.cs.unc.edu/.* URL `http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf`.

[118] Wenk, M. R. (2005) The Emerging Field of Lipidomics. *Nature Reviews Drug Discovery*, **4**, 594–601.

[119] Whetzel, P. L., Parkinson, H., Causton, H. C., Fan, L., Fostel, J., Fragoso, G., Game, L., Heiskanen, M., Morrison, N., Rocca-Serra, P. *et al.* (2006) The MGED Ontology: a resource for semantics-based description of microarray experiments. *Bioinformatics*, **22**, 866–873.

[120] Wickham, H. (2007) Reshaping data with the reshape package. *Journal of Statistical Software*, **21**, 1–20. URL `http://www.jstatsoft.org/v21/i12/`.

[121] Wickham, H. (2009) *ggplot2: elegant graphics for data analysis.* Springer New York. URL `http://had.co.nz/ggplot2/book`.

[122] Wiese, S., Reidegeld, K. A., Meyer, H. E. and Warscheid, B. (2007) Protein labeling by itraq: a new tool for quantitative mass spectrometry in proteome research. *Proteomics*, **7**, 340–350. URL `http://www.ncbi.nlm.nih.gov/pubmed/17177251`.

[123] Wilkins, M. R., Appel, R. D., Van Eyk, J. E., Chung, M. C. M., Görg, A., Hecker, M., Huber, L. A., Langen, H., Link, A. J., Paik, Y.-K., Patterson, S. D., Pennington, S. R., Rabilloud, T., Simpson, R. J., Weiss, W. and Dunn, M. J. (2006) Guidelines for the next 10 years of proteomics. *PROTEOMICS*, **6**, 4–8. URL `http://dx.doi.org/10.1002/pmic.200500856`.

[124] Wisniewski, J. R., Zougman, A., Nagaraj, N. and Mann, M. (2009) Universal sample preparation method for proteome analysis. *Nat Methods*, **6**, 359–362. URL `http://www.ncbi.nlm.nih.gov/pubmed/19377485`. PMID:19377485.

[125] Wolski, W. E., Farrow, M., Emde, A.-K., Lehrach, H., Lalowski, M. and Reinert, K. (2006) Analytical model of peptide mass cluster centres with applications. *Proteome science*, **4**, 18.

[126] Wong, J. W., Sullivan, M. J. and Cagney, G. (2008) Computational methods for the comparative quantification of proteins in label-free LCn-MS experiments. *Briefings in Bioinformatics*, **9**, 156–165.

[127] Yu, T., Park, Y., Johnson, J. M. and Jones, D. P. (2009) apLCMS—adaptive processing of high-resolution LC/MS data. *Bioinformatics*, **25**, 1930–1936.

[128] Zhang, J., Gonzalez, E., Hestilow, T., Haskins, W. and Huang, Y. (2009) Review of peak detection algorithms in liquid-chromatography-mass spectrometry. *Current Genomics*, **10**, 388.

[129] Zhang, X., Asara, J. M., Adamec, J., Ouzzani, M. and Elmagarmid, A. K. (2005) Data pre-processing in liquid chromatography–mass spectrometry-based proteomics. *Bioinformatics*, **21**, 4054–4059.

[130] Zhang, Z. and Marshall, A. G. (1998) A universal algorithm for fast and automated charge state deconvolution of electrospray mass-to-charge ratio spectra. *Journal of the American Society for Mass Spectrometry*, **9**, 225–233.

[131] Zhou, B., Xiao, J. F., Tuli, L. and Ressom, H. W. (2011) LC-MS-based metabolomics. *Mol. BioSyst.*

[132] Zhu, Z.-J., Schultz, A. W., Wang, J., Johnson, C. H., Yannone, S. M., Patti, G. J. and Siuzdak, G. (2013) Liquid chromatography quadrupole time-of-flight mass spectrometry characterization of metabolites guided by the metlin database. *Nature protocols*, **8**, 451–460.