Regression models to Detect and Quantify Peptides from Mass Spectra

Alex Hu

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:

William Stafford Noble, Chair

Michael MacCoss

Jeff Bilmes

Program Authorized to Offer Degree:
Genome Sciences

University of Washington

Abstract

Regression models to Detect and Quantify Peptides
from Mass Spectra

Alex Hu

Chair of the Supervisory Committee:
Professor William Stafford Noble
Genome Sciences

Data-independent acquisition (DIA) mass spectrometry-based proteomics aims to quantify every peptide and its derivatives in a sample by systematically sampling every ion. However, much of the signal in the resulting spectra is difficult to interpret because they represent complex mixtures of ions, preventing the accurate quantification of every peptide.

I propose regularized linear regression approaches to jointly account for mixtures of multiple peptides and their relationships in DIA spectra to deconvolve spectra precursor and fragment spectra, remove the problem of interference, and improve the sensitivity and precision of peptide detection and quantification. The deconvolution extracts information invisible to current methods and provides a framework to detect and quantify more peptides.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

DDA: Data-dependent acquisition.

DIA: Data-independent acquisition.

MS1: Spectra containing precursor ions.

MS2: Spectra containing ions resulting from the fragmentation of intact precursor ions.

LASSO: Least absolute shrinkage and selection operator: the use of L1 regularization on the parameters of a regression problem.

ACKNOWLEDGMENTS

I acknowledge as instrumental to this work and my development as a student: my advisors Bill Noble and Alejandro Wolf-Yadlin, my committee, Jeff Howbert, Sonia Ting, Jarrett Egertson, Lindsay Pino, Nathan Camp, and Zoi Sychev.

Chapter 1

INTRODUCTION

Liquid chromatography-tandem mass spectrometry (LC-MS/MS)-based proteomics has provided broad detection and relative quantification of thousands of proteins across a variety of biological samples using a data-dependent acquisition (DDA) strategy. In DDA, the mass spectrometer sequentially samples ion species from narrow precursor m/z windows as they elute from chromatography and fragments each sample into its own MS2 spectrum. The goal of DDA's use of narrow precursor m/z windows is for each MS2 spectrum to contain molecules of a single peptide sequence so that they can be identified but not necessarily quanitified. In recent years, alternative LC-MS/MS targeted acquisition strategies, such as multiple-reaction monitoring[25] and parallel reaction monitoring (PRM)[18], have provided precise and reproducible absolute quantification of up to hundreds of proteins. A next goal of proteomics is the development of acquisition strategies that have both the breadth of DDA and the precision of MRM/PRM to provide reproducible identification and quantification of every protein in any biological sample. Data-independent acquisition (DIA) may provide a viable path to this goal because it is designed to fragment every ion species, but current pipelines to analyze DIA are yet unable to detect and quantify every ion species because they do not account for key aspects of of the complex structure of DIA spectra. Below is a description of the problem of peptide detection and quantification from DIA spectra and the current approaches to solve it.

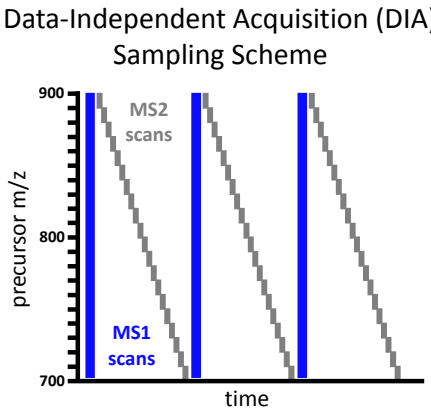

Figure 1.1: **Schematic of DIA's sampling scheme.** Each rectangle represents a spectrum. A rectangle's height on the y-axis denotes the precursor m/z's of the peptides it contains, and its position on the x-axis denotes the elution time at which it was produced. Blue rectangles are MS1 spectra and gray rectangles are MS2 spectra.

1.1 Shotgun Proteomics by Mass Spectrometry using Data-independent Acquisition

A typical DIA experiment generates spectra in a pre-determined, repeating sequence that begins with an MS1 scan covering a pre-determined m/z range followed by a sequence of MS2 scans with wide, non-overlapping precursor isolation windows that span a given m/z range (Figure 1.1). The MS1 and MS2 sequence is repeated until the chromatography finishes. The period of the repeated sequence is designed such that any given peptide species is sampled multiple times across its elution profile in MS1 and MS2 spectra to facilitate more accurate detection and quantification. In typical chromatography protocols that last 90 minutes, the period 2-3 seconds. The DIA data therefore has three dimensions- time, precursor m/z (MS1), and fragment m/z (MS2).

1.2 Popular approaches to detect peptides from DIA spectra

Proper interpretation of DIA data is currently problematic because the complex MS2 scans with wide precursor windows contain mixtures of peptides and therefore are more difficult to analyze. Fortunately, recent developments in bioinformatics software have adequately overcome this DIA issue, so that DIA now closely matches DDA in the number of peptide identifications while still allowing precise quantification of most of them. Although traditional algorithms for identifying peptides from DDA spectra can be applied to DIA spectra analysis, these algorithms are not appropriate for DIA for two reasons: they incorrectly assume that each MS2 scan contains fragments from just one peptide, and they ignore the dynamic pattern of elution profiles in DIA spectra. Consequently, three main classes of computational algorithms have emerged to specifically analyze DIA data that accommodate the complexity and time variation in DIA spectra. The first two classes are for untargeted, discovery-based identification, and the third is for precise quantification of previously identified peptides from spectral libraries.

1.2.1 Deconvolution of fragment spectra over elution time

These methods use a pre-processing step that deconvolves DIA MS2 scans into multiple pseudo-spectra, each ideally containing the fragments of only a single peptide species in the mixture. The intensities of different fragments of the same peptide species should correlate over elution time, and the pre-processing step uses this correlation to assign fragment ions from MS2 scans to their intact peptide species in MS1 scans. These pseudo-spectra then can be searched by using a traditional DDA database search method. DIA-Umpire[29] and DeMux[2] are two strategies that take this approach. They differ in the specific algorithms used to group ions and compile them into deconvolved spectra. In principle, DIA-Umpire tends to work better because it considers isotope peak distributions in MS1 scans to narrow down candidate peptides. DIA-Umpire can detect up to 89% of the peptides detected by analogous DDA experiments. DIA-Umpire also includes additional methods that generate

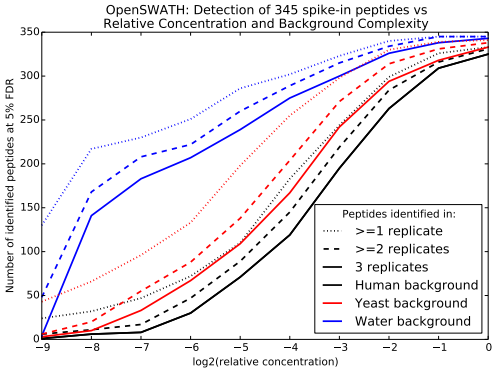

Figure 1.2: **Numbers of reproducible synthetic peptide detections** across 3 replicates at 5% FDR by OpenSWATH as a function of peptide concentration and sample backgrounds that vary by their complexity. The water background is the least complex, the human background is the most complex, and the yeast background is in between.

new reference/ library spectra, incorporates prior library spectra, and uses them for further steps in protein detection and quantification, achieving on average a 0.931 R2 correlation in the quantifications of peptides between replicates. This strategy works in a fashion analogous to DDA experiments in that it detects many peptides, but also the strategy allows for precise DIA quantification of the detected peptides[29].

1.2.2 Chromatographic peak finding

Strategies in this category are adapted from methods used to analyze PRM/MRM spectra, because both DIA and MRM/PRM repeatedly sample the same peptides to obtain sequences of fragment ion intensities over elution time (fragment ion chromatograms; Figure 2D). These methods take library spectra as input (compiled from prior DDA peptide identifications or prior DIA-Umpire identifications) and extract fragment ion chromatograms at the peaks in the library spectra. Potential elution peaks of each peptide from these fragment ion chromatograms are evaluated on the basis of many criteria, including how well the fragment ions correlate over elution time and how well their relative intensities match their corresponding

library spectrum. Elution peaks are evaluated by using a discriminative model that combines these criteria to distinguish real peptide signals from decoy peptide signals. These methods also quantify proteins by quantifying the fragment ion chromatograms of their peptides.

OpenSWATH[22], SWATHProphet[10], Spectronaut[3], and a module in DIA-Umpire[29] all implement this strategy. Skyline[14] provides elution peak quantification but not statistical validation. OpenSWATH[22] has been shown to achieve coefficients of variation of between 0% and 20% on 345 spike-in peptides detected across 256-fold concentration differences in a yeast lysate, although the ability to detect these peptides decrease as concentration decreases(Figure 1.2). In a series of human lysate runs, Spectronaut[3] achieved a 98% peptide identification reproducibility rate from run to run on DIA data compared with 49% on DDA data on 26,738 peptides covering 3,690 proteins.

1.2.3 Dot-product scoring

These strategies are inspired by traditional database searching without library spectra and do not focus on using dynamic patterns to detect peptides. Instead, they include other heuristics to adapt to a DIA context. They score each peptide against each observed spectrum by computing the dot-product of the peptides theoretical spectrum against each observed spectrum, much like traditional DDA search algorithms. However, these methods introduce additional heuristic filtering steps, such as considering only observed spectra that match a threshold number of peaks in a theoretical spectrum. The first example of this strategy was FT-ARM[32]. However, a more recent method, Pecan[28], contains additional heuristics as well as a discriminative model to combine them, so it performs better than FT-ARM. It aggregates the dot-product scores and auxiliary heurstics over multiple scans to collect the evidence for the presence of each peptide and computes statistical signifiance for the evidence at the peptide level.

1.3 Regression Approaches

The above analytical strategies identify and quantify many peptides from DIA spectra, but none of them explicitly account for the contributions of multiple peptides to the same peaks simultaneously. Because they do not jointly account for peptides, the analytical strategies detect fewer peptides when samples become become complex (contain more peptides), and the spectra become mixed (Figure 1.2) However, doing so improves spectral interpretation. The methods Nitpick[20] and Specter[17] for MS1 and MS2 analysis, respectively, use explicit models of spectra as linear combinations of signals from multiple ion species. From these models, they use LASSO regression to infer the relative abundances of the ion species that effectively disentangle the combined signals for better peptide detection [17].

This thesis proposes three additional regression approaches that expand on the successes of Nitpick and Specter. The method Siren scales and extends Nitpick's approach, designed for DDA data, to apply to the larger scale and dimensionality of DIA data. The method Bichir extends Specter's approach to include peptides for which library spectra do not exist and considers elution profile shape in the optimization of its model. The method Defraggle is a reformulation of DIA-Umpire's approach into a regression problem that also jointly incorporates known fragment ion relationships. The new regression approaches and the precedents that inspired them are described further in the following chapters.

Chapter 2

SIREN: REGRESSION TO DETECT PRECURSORS FROM MS1 SPECTRA

2.1 Introduction

The detection of peptides from MS2 spectra is facilitated by precise knowledge of precursor mass and charge from MS1 spectra[9]. Database search algorithms for DDA identify each MS2 spectrum from a set candidate peptides that could have produced the peak or peaks contributing to that spectrum. The more knowledge a method has of the peaks, the fewer candidate peptides are tested, increasing the statistical power to find true peptide-spectrum matches. The minimum amount of knowledge about a precursor peak that produced an MS2 spectrum is that its m/z value lies somewhere within an error tolerance of the isolation m/z of the spectrum. Three additional pieces of knowledge about the peak can help narrow down the candidate peptide list: its charge state, precise m/z, and the number of heavy elemental isotopes it contains. All of this information may be inferred from MS1 peaks proximal to the MS2 spectrum in question. These pieces of information are also useful in DIA analysis for similar reasons, and we call the determination of this knowledge p̈recursor annotation.̈

2.1.1 Existing methods

There are a few methods that annotate precursors from MS1 scans: Bullseye[9], Nitpick[20], and DIA-Umpire[29]. Each of these are designed to consider different pieces of relevant information to improve precursor inference, but no individual method considers all of them to their fullest extent (Table 2.1).

Bullseye analysis has two steps[9]; the first examines MS1 peaks in and around the MS2 isolation window and determines which possible isotope distributions are consistent with

Table 2.1: Comparison of Precursor Annotation Methods from MS1 Spectra

Method	**Joint peptide model**	**Considers multiple scans**
Nitpick	✓	✗
Bullseye	✗	✓
DIA-Umpire	✗	✓
Siren	✓	✓

the m/z values, number, and m/z spacing of the peaks. An isotope distribution is a set of peaks that represent a molecular ion species with that share a common elemental formula and charge but differ in m/z because they consist of different combinations of elemental isotopes. The second step checks which of the possible isotope distributions persist in a sequence of MS1 scans around the MS2 scan because they are more likely to be true than those that do not persist. The subsequent database search algorithm then considers these persistent isotope distributions. Bullseye does not model interference because it does not jointly model the contributions of multiple precursors to individual peaks, which can prevent it from correctly identifying peaks whose relative intensities are inconsistent with models of individual isotope distributions. Its consideration of precursors individually also may erroneously recognize the isotope distribution of a non-existant precursor from combinations of peaks from other peptides. Bullseye ignores the informative variation of intensity of the same peak over multiple scans; isotope peaks of the same distribution co-elute such that their intensities over time should correlate absent interference, which is behavior that other methods like DIA-Umpire consider.

Nitpick[20] is a regression method that jointly matches multiple possible isotope distributions to observed peaks. Nitpick finds the linear combination of isotope distributions that best matches the observed peaks, making it less susceptible to interference, and identifies isotope distributions that substantially contribute to each MS2 spectrum. Nitpick does not take into account persistence and variation of the isotope distributions over elution time to

help narrow down the candidates. Nitpick uses a narrow bin widths of $8 \cdot 10_5$ Th to provide precise estimates of precursor m/z.

Software that analyzes DIA data uses and infers precursor information in more diverse ways. DIA-Umpire includes precursor identification as a first step within a larger pipeline to deconvolve MS2 spectra. It identifies precursor ions by extracting precursor peak chromatograms and identifying them as a precursor's isotope distribution if they correlate well over elution time and have appropriate m/z values. DIA-Umpire does not jointly consider multiple precursors, which makes it vulnerable to interference that disrupts the correlation between isotope peaks over time. Pecan[28] is similar to DDA database search but also uses precursor information in its function to evaluate peptide candidates. Pecan computes a theoretical isotope distribution for the candidate and uses its dot-product against observed peaks. Because the dot-products consider only one peptide at a time and are agnostic to the presence of other peptides, they will be high and uninformative for non-existent peptides whose isotope distribution happen to overlap those of existing peptides. Accordingly, Pecan quantifies background signal and revises its interpretation of the dot-product based on the quantification of the background.

We propose a precursor annotation method called Siren (Sparse Isotope RegressioN) that builds from Nitpick by using regression to jointly model multiple precursor and from DIA-Umpire by sparsely representing peaks and considering their variation over time. The joint modeling enables proper deconvolution of interference, and its sparse representation allows it to efficiently scale to wide DIA isolation windows. It also provides estimates of peptide abundance over multiple scans to denote precursor elution profiles.

2.2 Linear model of MS1 spectra in DIA data

Siren represents the observed MS1 spectra as a linear combination of theoretical spectra. Siren approximates a single MS1 spectra as the sum of weighted theoretical precursor isotope distributions (Figure 2.1a). Repeating this relationship for a sequence of successive scans yields the full model (Figure 2.1b). Accordingly, the model consists of three matrices.

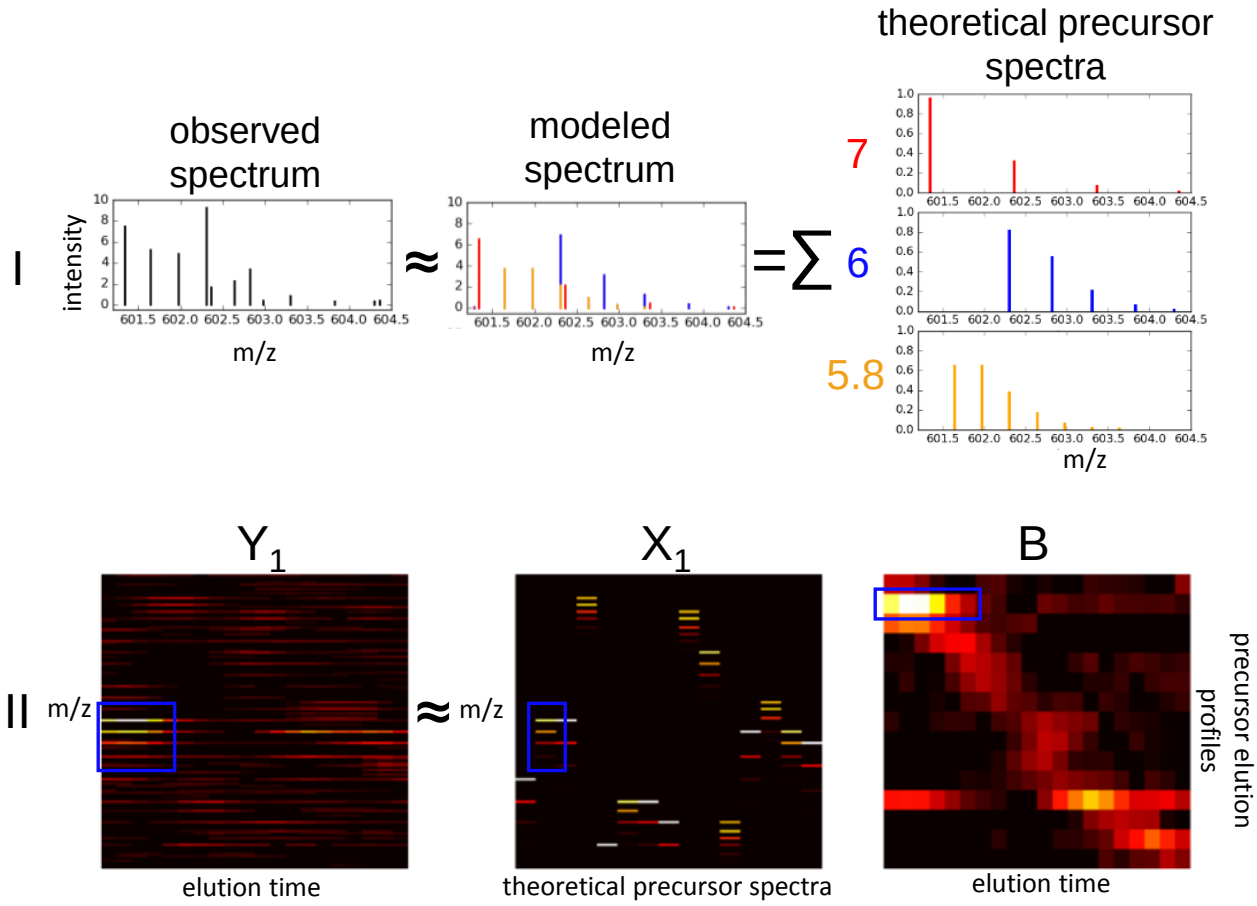

Figure 2.1: **I. Linear model of single MS1 spectrum.** II. Linear model of a sequence of MS1 spectra. The boxed isotope distribution and elution peak in the X_1 and B matrices yield the boxed pattern visible in matrix Y_1.

- $Y_1 \in \mathbb{R}^{MxT}$ consists of the observed MS1 data, corresponding to a series of MS1 spectra from a single mass spectrometry run. Each of T columns in Y_1 is a vector representing a single MS1 spectrum, and each of M rows corresponds to a single m/z bin across the spectra.

- $X_1 \in \mathbb{R}^{MxN}$ contains theoretical isotope peaks for each precursor ion hypothesized to exist in the data. Accordingly, each of N columns in X_1 represents a normalized theoretical MS1 spectrum produced by a single precursor ion defined by its mass and charge, with the same discretized m/z bins as Y.

- $B \in \mathbb{R}^{NxT}$ contains the inferred abundances of the precursor ions of X_1 in Y. Each of N rows in B represents the abundance over time of a single precursor. Conversely, each column in B represents the abundances of all N theoretical precursors in one observed spectrum in Y.

Siren assumes that

$$Y = X_1 B + \epsilon \tag{2.1}$$

where ϵ represents machine noise, contaminants, and other signals that are not represented by theoretical peaks in X. Siren proceeds in three steps: constructing the matrix Y, constructing the matrix X_1, and then inferring the matrix B. However, Siren exploits the observation that the inference of each column in B (i.e., one vector of abundance values across all precursors) can be carried out independently for each column (scan) in Y. Thus, the three steps are carried out separately for each scan.

2.2.1 Construction of the observed matrix Y and theoretical matrix X_1

Each observed spectrum consists of a set of peaks, with real-valued m/z and intensity values. However, Siren requires that these peaks be arranged in a matrix Y. The columns of Y correspond to discrete scans, but the rows must be created by some form of discretization of the m/z axis.

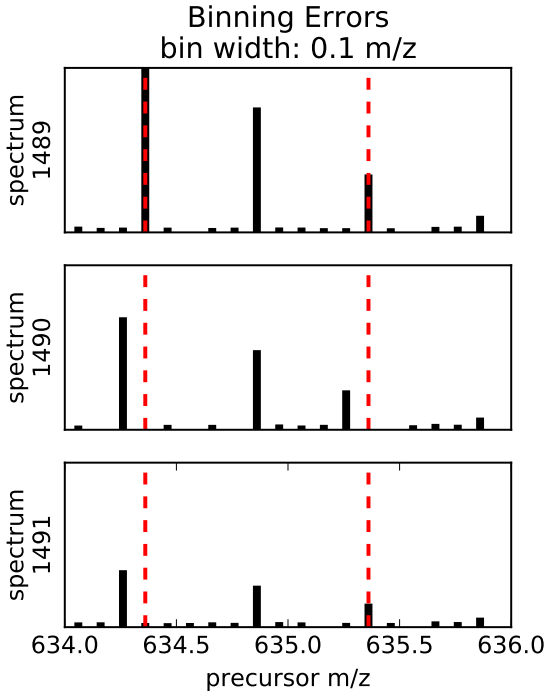

Figure 2.2: **Binning errors introduced by small m/z bin size.** The figure shows excerpts of a sequence of three MS1 spectra over time, depicting the elution of a charge +2 isotope distribution whose peaks are binned at a 0.1 m/z bin width. The monoisotopic peak and the third isotopic peak each shift one bin to the left from scan 1489 to scan 1490.

Siren does not adopt the conventional strategy, in which the m/z span is divided into equal-width bins, for several reasons. Say that we are working with high-resolution data that exhibits a native resolution of 10 parts-per-million (ppm) on the m/z axis. If we make our bins much larger than 10 ppm, then we end up erroneously placing into the same column peaks that should be assigned distinct m/z values. On the other hand, if we use a fine-grained bin size, then edge effects will introduce arbitrary (and incorrect) distinctions between peaks that truly correspond to the same underlying type of ion (Figure 2.2). Furthermore, a very fine binning scheme leads to a very large matrix, making calculations very expensive. For example, Nitpick, which was designed for targeted analysis of individual precursors, uses a very fine width of $8 \cdot 10^5$. Using this binning scheme across the commonly used m/z span of 400–1200 leads to 10 million bins. Nitpick mitigates off-by-one errors by describing each isotope peak as a Gaussian distribution of many peaks across many m/z bins rather than as a single peak in a single bin, but this smoothing increases Nitpick's computational expense.

Rather than arbitrarily binning the m/z axis, Siren creates entries in Y that correspond to observed monoisotopic peaks and their corresponding isotope peaks. These entries sparsely describe the important portions of the m/z range, while ignoring m/z values unoccupied by informative peaks. The specific procedure to create a vector Y for a single observed scan is as follows:

1. **Noise reduction** Peaks that occur in a single scan, with no corresponding peaks in neighboring scans, likely represent noise. Accordingly, for a scan at time t and a peak with m/z value m, we consider the scans at times $t+1$ and $t-1$. The current peak is eliminated if neither of the adjacent scans contains a peak with m/z value in the range $[m-\tau, m+\tau]$, where τ is an m/z tolerance value reflecting the precision of the data.

2. **Observed peak clustering** Peaks are clustered together if they fall within a noise m/z tolerance with the assumption that small variations in m/z value represent imprecise measurements of the same ion species that should be represented in the same row in Y. The peaks are represented as a graph, in which vertices are peaks and an

edge connects pairs of peaks within a small m/z tolerance of each other. Peaks are then clustered, such that each cluster corresponds to a connected component within the graph. Each cluster c_i has an associated range of m/z values, $m_i^{\min}$ to $m_i^{\max}$.

3. **Monoisotope identification** For each peak cluster c_i, we look for a sibling peak cluster that is separated from c_i by an m/z difference associated with a pair of isotopic peaks. For a pair of clusters c_i and c_j, a sibling relationship is identified if and only if the m/z difference between the clusters is consistent with the distance between the first two isotopic peaks of a precursor of charge +1, +2, +3 or +4. Formally, the latter condition corresponds to

$$m_i^{\min} + 1.0/C < m_j^{\max} + \tau \tag{2.2}$$

and

$$m_i^{\max} + 1.0/C > m_j^{\min} - \tau \tag{2.3}$$

for charge values of $C \in \{1, 2, 3, 4\}$. Each time a sibling cluster is identified, the initial cluster is marked as a monoisotopic peak with the corresponding charge. Note that, in this procedure, a single cluster can occasionally be marked with multiple charge states. This step ensures that only monoisotopic precursors whose first two isotope peaks exist in the data are included in the model.

4. **Theoretical peak generation** In principle, the m/z and relative intensities of peaks in an isotope distribution can be accurately predicted from the charge of the ion and its elemental composition [21]. However, in our setting, the elemental compositions of each ion in the sample is not known a priori. Accordingly, Siren employs the "averagine" model to approximate a theoretical isotope distribution for each observed monoisotopic peak with m/z $m_i = (m_i^{\min} + m_i^{\max})/2$ and charge C_i [23]. The calculation retains the +1 through +6 isotope peaks, each with an m/z value and a relative intensity.

5. **Construction of Y.** The vector Y is constructed with one entry corresponding

to each observed monoisotopic peak, plus additional entries for the +1 through +6 peaks in each marked charge state. Intensity values for the monoisotopic peaks are directly observed. For the remaining isotope peaks, if an observed peak exists in the data within an m/z tolerance of τ, then the observed intensity is used; otherwise, the intensity is set to zero.

6. **Construction of X_1.** The matrix X_1 of theoretical isotope distributions is constructed such that rows in X_1 correspond to entries in Y. The averagine isotope distributions are placed into X_1, scaled so that the L2 norm of each column in X_1 is equal to 1. If the theoretical isotope peaks across X_1 do not exist within τ m/z of an observed peak, they are clustered with each other in the same way as described in Step 2. Each of these clusters forms a row in Y_1 and X_1.

We also implemented a variation of the above method to construct X and Y, in which multiple scans are considered jointly. The intution is that, given that the same ion species will elute over multiple scans, the peak clustering might improve if the peaks spanned multiple scans. The construction works in essentially the same way, except that the peaks over contiguous and disjoint sequences of spectra (blocks of spectra) of a given length are clustered together, and Y becomes a matrix where each column is a different scan that share the same rows. Step 2 from above is changed such that edges may be drawn between peaks of adjacent scans within the same block to create peak clusters that span multiple spectra. The differences in performance of these variants compared to the original are discussed in Results.

2.2.2 LASSO regression to identify precursors

Finally, given matrices Y and X, Siren solves the following optimization problem:

$$\underset{B \geq 0}{\operatorname{argmin}} \quad ||Y_1 - X_1 B||_2^2 + \lambda_1 ||B||_1 \tag{2.4}$$

This is known as LASSO regression with a non-negative constraint [26]. The regularization parameter λ_1 controls overfitting and increases the sparsity of the learned B. We cannot know a priori what level of sparsity will yield optimal performance, so a grid search of λ_1 were tried across some of the Siren models on one dataset H1 and used that λ_1 on other Siren models. In the future, there should be a procedure for fitting λ_1 automatically using cross-validation. The grid search and cross-validation procedures and their performance measures may vary depending on what Siren is used for. Here, the performance measure used was whether the precursor inferences could be identified by database search of the MS2 spectra.

2.2.3 Construction of the abundance matrix B

Because Siren is run separately on each scan, the LASSO regression procedure produces a sequence of vectors $\{b_1, ..., b_T\}$, each of which contributes one column in B. However, the rows from vector to vector do not necessarily correspond to the same hypothesized precursor and are therefore aligned to form B using the following procedure.

1. The vector b_1 is designated the first the column of B.

2. Subsequent vectors are added one by one to B. To add b_t to B that already contains $\{b_1, ..., b_{t-1}\}$, the set of N_t rows of b_t that correspond to the set of precursors $\{p_{t,1}, ..., p_{t,N_t}\}$ are incorporated one at a time to B:

   (a) If the monoisotopic peak of $p_{t,n}$ overlaps with that of one precursor of the same charge already in B that is non-zero at time $t-1$, $b_{t,n}$ is appended to that row in B.

   (b) If the monoisotopic peak of $p_{t,n}$ overlaps with more than one non-zero precursor of the same charge already in B, the rows for the overlapping precursors are summed together and merged into a single row in B, and $b_{t,n}$ is appended to that row.

   (c) If the monoisotopic peak of $p_{t,n}$ does not overlap any precursor of the same charge with a non-zero value at time $t-1$, then a new row of zeros is created in B and

assigned to $p_{t,n}$, and $b_{t,n}$ is appended to that row.

After the entirety of $\{p_{t,1}, ..., p_{t,N_t}\}$ has been added to B, a zero is appended to all rows that did not get assigned a new precursor from scan t.

2.2.4 Data sets

The spectra analyzed came from experiments measuring HEK-293 lysates using an Orbitrap Fusion mass spectrometer[30] over a 135-minute gradient, downloaded as mzXML files from `ftp://ftp.pride.ebi.ac.uk/pride/data/archive/2016/06/PXD003179/` and were converted to ms1 and ms2 file formats using msconvert. The two runs used are referred to as H1, with 2205 MS1 spectra, and H2, with 2219 MS1 spectra. The isolation widths vary from 24 to 222 m/z across the range 400-1250 m/z.

2.2.5 Validation procedure

We use the associated MS2 data to validate Siren's results because Siren considers only the MS1 data. We extract elution peaks for each inferred precursor using the following procedure. For each precursor (row) in B, we apply Savitzky-Golay smoothing (using a third degree polynomial over five points at a time) [19]. Local maxima are then found in the smoothed row as points that are greater than both of its adjacent values. We only consider maxima whose adjacent values are both greater than 0. For each local maximum, we extract the corresponding MS2 scan and search it using a traditional database search engine, using the precursor mass identified by Siren. This approach is imperfect because many peptides in the sample may fail to be identified if their peaks are obscured by other co-eluting peptides of higher abundance. Nonetheless, the procedure provides an unbiased way to compare the relative rates at which precursors identified by different MS1 analysis methods are successfully identified from MS2 data.

We employ the Tide search engine with exact p-values using a database of human tryptic peptides derived from Uniprot (11/02/2014) using various precursor isolation widths from

0.005 to 0.1 m/z [8]. Shuffled decoys were created by Tide, and the top two peptide matches were retained for each precursor, allowing target-decoy competition. Peptide-level FDR was estimated using the "weed-out then estimate" procedure [7]. Spectrum-level FDR was also reported final performance measures that we employ are the number of identified MS2 spectra and the number of distinct peptides detected at an FDR threshold of 1%.

To evaluate how accurately Siren models observed peaks, we use two metrics that compare the observed Y_1 to the modeled peaks X_1B_1. The first is average R^2 was defined as $1 - \frac{\sum_{t=1}^{T} ||Y_t - \hat{Y}_t||_2^2}{\sum_{t=1}^{T} ||Y_t - X_t B_t||_2^2}$ where $\hat{Y}_t$ is a matrix filled with the mean value in Y_t of the same dimension as Y_t. The second is the average absolute error in peak intensity in the models, defined as $\frac{\sum_{t=1}^{T} \sum |Y_t - X_t B_t|}{\sum_{t=1}^{T} Y_t}$. As a control, additional models were similarly constructed and evaluated using the above metrics on randomized Y_t matrices in which the peak intensities within each column of Y_t were randomly shuffled while preserving the m/z positions of the peaks.

2.2.6 Competing methods

For comparison, the human lysate data set was analyzed using DIA-Umpire. DIA-Umpire performed its deconvolution pipeline, thereby identifying precursor elution profiles and deconvolving the MS2 spectra into one pseudospectrum for each elution profile. The pseudospectra were categorized into three sets: Q1 if its first three precursor isotope peaks appear in the MS1 spectra, Q2 if the first two appear, and Q3 if only its precursor peak appears in the MS2 spectra. Because we only want to compare its ability to identify precursors relative to Siren, we ignore the deconvolved MS2 information within the pseudospectra and extract only the precursor masses, charges, and peak elution times. We then employ the Tide search engine to search the raw MS2 spectra using the precursor annotations as described in the previous section. Note that the Q3 precursors are derived from the MS2 data, which Siren does not consider.

2.3 Results

Siren's performance was tested at two levels; how accurately its models can describe observed peaks and how many of its annotated precursors can be validated in the MS2 data via database search. Siren has the tunable parameter $lambda_1$ that affects its performance, so its effect on performance is also explored.

2.3.1 Ability of peptide isotope distributions to describe data

To determine whether Siren accurately models MS1 spectra, we build a Siren model for dataset H1 and infer B, setting $\lambda_1 = 0$ to maximize the model's accuracy in modeling peak intensities; non-zero values of λ_1 result in the underestimation of the values in B and the modeled peak intensities in X_1B.

The learned models produced an average R^2 of 0.982, meaning that 98.2% of the variance in peak intensities was attributable to precursor isotope models. The average absolute error in peak intensity in the models was 0.18. The average R^2 from the shuffled control model was 0.602 with an average absolute error of 0.694. This means that the data in particular are accurately described as linear combinations of precursor isotope distributions.

2.3.2 Peptide identification vs sparsity of Siren model

The sparsity of the Siren model can controlled by varying λ_1, so models derived from multiple values of λ_1 were constructed and evaluated. The sparsity of the models increase with greater λ_1, and the number of elution peaks annotated decreases with sparsity (Figure 2.4). The Siren model with no regularization ($\lambda_1 = 0$) assigned non-zero abundances to 69% of the hypothesized precursors, leading to 129,673 elution peaks and 2,104 identified peptides at 1% FDR from database search on the raw MS2 spectra. The Siren model from which the most peptides could be confidently identified from MS2 database search assigned non-zero abundances to 5.78% of hypothesized precursors, leading to 20,047 elution peaks and 2,505 identified peptides. The greater number of peptide detections is partly due to LASSO's

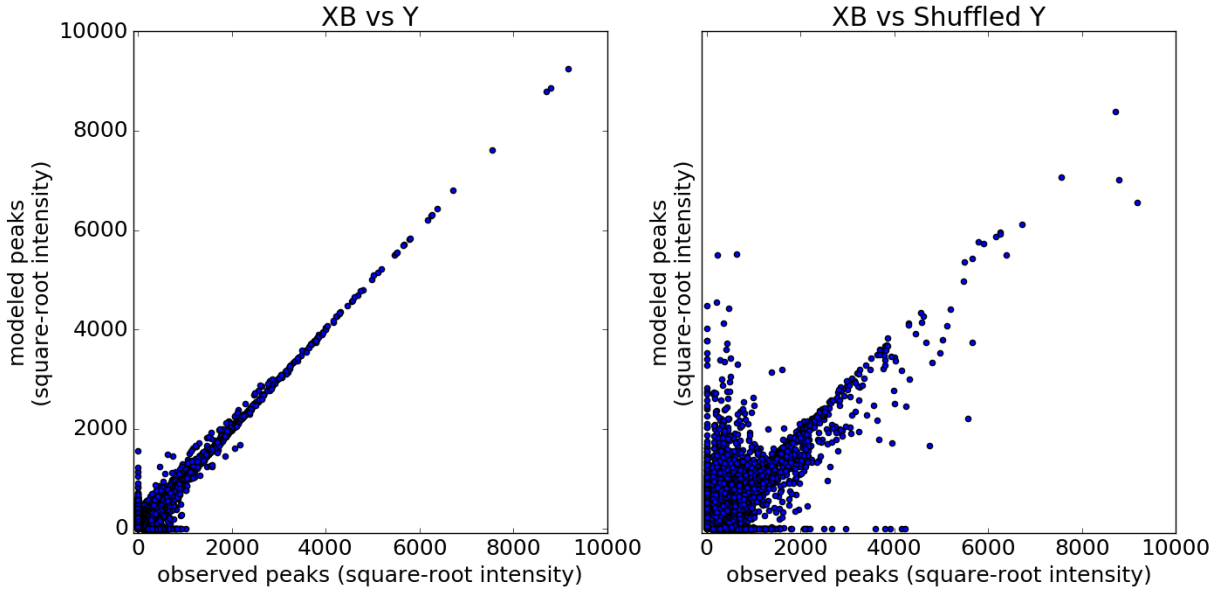

Figure 2.3: **Observed peak intensities (Y) vs modeled peak intensities (XB) from Siren.** The left plot shows the observed and modeled peaks. The right plot shows the peaks whose intensities have been shuffled within each scan and the peaks modeled from those shuffled peaks.

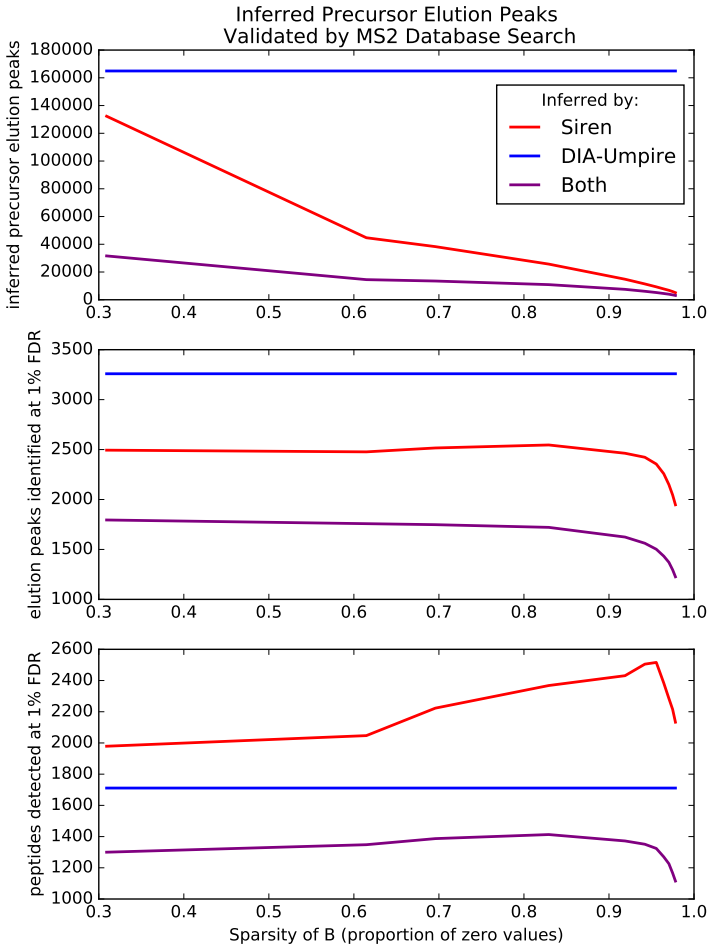

Figure 2.4: **Number of precursors annotated, precursors identified, and unique peptides detected** by DIA-Umpire without fragment deconvolution, Siren, and both methods. (0.01 m/z tolerance).

removal of false positives. However, LASSO also removed 416 peptides confidently identified by the Siren model with no regularization, showing that LASSO also removes true signal from peptides of low abundance.

2.3.3 Peptide identification from inferred precursors

Next, we aimed to compare the performance of Siren, DIA-Umpire, Bullseye, and Nitpick by using them to select spectra and precursor m/z values from a DIA data set derived from a human cell lysate. Unfortunately, we found that Nitpick, which was designed for targeted analysis of individual precursors, cannot scale to analysis of a full MS1 run. Also, Bullseye, which is no longer actively maintained, yielded very few validated identifications. Therefore, our analysis focused on a comparison to DIA-Umpire, which is a recently developed and actively maintained tool.

Siren and DIA-Umpire identify similar numbers of precursor elution peaks, but they only agree on a small subset (Figure 2.4). More peptide sequences are identified from elution profiles inferred by Siren than DIA-Umpire, and a majority of DIA-Umpire's identified peptides are also found by Siren. However, DIA-Umpire can only identify about half of Siren's at its optimal level of sparsity. Bullseye did not appear to work properly on this data given its low yield of peptide identifications.

Because Siren jointly considers the contributions of multiple precursors to observed peaks, it is able to precisely pinpoint the elution peaks of precursors even if their peaks exhibit interference.

Figure 2.5 shows an example of a precursor ion II, found by database search whose observed peaks exhibit interference from precursor I. Both Siren and DIA-Umpire identify precursor I, but only Siren identifies precursor II. Although the precise reason why DIA-Umpire could not identify precursor B is unknown, the Figures 2.5 A and B suggests that it is because precursor II's second isotopic peak did not correlate well over time with the first and third due to interference from precursor II. Figure 2.5.B shows that Siren was able to deconvolve the interference and recognize the elution profile of I.

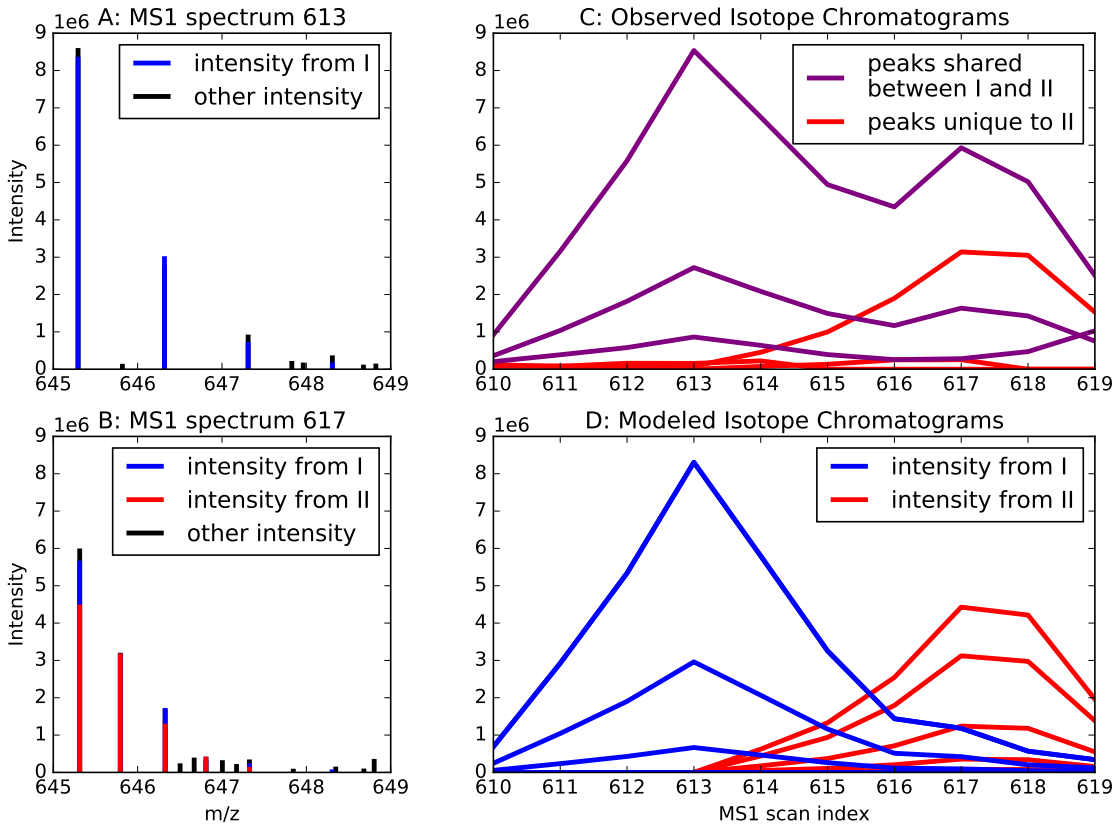

Figure 2.5: **A: Window of MS1 spectrum** 613 depicting peaks identified by DIA-Umpire and Siren as the isotope distribution of precursor ion I whose monoisotopic m/z is 645.3 and charge is 1. The colors denote the precursor ion(s) contributing to peak intensity according to Siren. B: Window of MS1 spectrum 617 depicting peaks identified by and Siren as the combined isotope distributions of precursor ion I and precursor ion II, whose monoisotopic m/z is 645.3 and charge is 2. The colors denote the precursor ion(s) contributing to peak intensity according to Siren. Siren suggests that the monoisotopic peaks of I and II interfere with each other and that the second isotopic peak of I interferes with the third isotopic peak of II. Precursor II is not recognized by DIA-Umpire but is matched by database search to the sequence DISTNYYASQK. C: Chromatograms of the ions of precursors I and II. Each chromatogram is labeled purple if its m/z value matches to both precursors I and II or red if it matches only to precursor II. D: Ion chromatograms modeled by Siren for precursors I and II whose colors correspond their precursors.

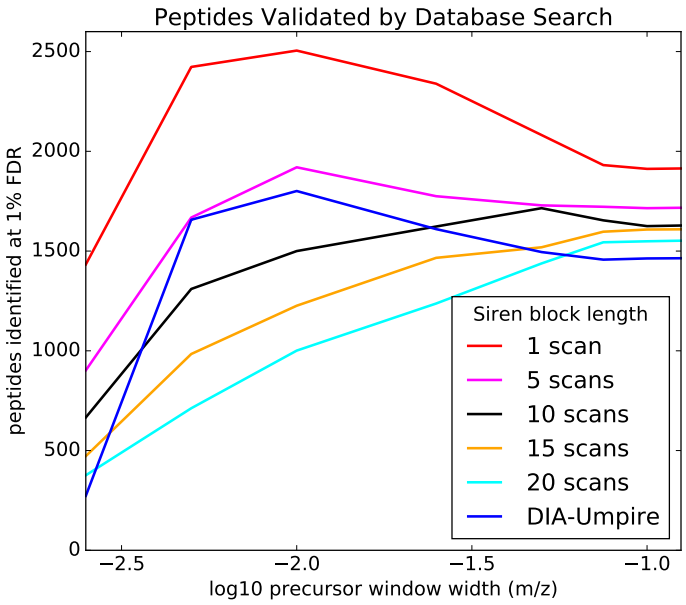

Figure 2.6: **Number of unique peptides identified at 1% FDR** from Siren models of different scan sequence lengths and DIA-Umpire at different precursor m/z windows for database search.

2.3.4 Effect of block length on optimal binning resolution

The theoretically ideal precursor tolerance in a database search spans the range of all m/z values occupied by the ions isolated and fragmented into an MS2 spectrum and no more. The span depends on the mass spectrometer's precision in measuring the m/z values of the ions as well as its precision in isolating ions from its target m/z range. This tolerance also varies in the post-processing algorithm's determination of precursor m/z's. The peak clustering algorithms of Siren and DIA interpret clusters of noisy peaks as a singular ion species with the same m/z value. However, these algorithms vary in their implementation and result in different optimal precursor tolerances that maximize the number of confidently identified precursor sequences. Siren in particular can cluster peaks within blocks that vary in the number of spectra they contain.

Figure 2.6 shows the performance of different Siren models that vary by the number of spectra in the blocks at different precursor window tolerances. The Siren models with a block length of 1 were tested using varying values of λ_1 as seen in Figure **??**, while the λ_1 values for the other Siren models were chosen using a heuristic related to an upper bound in non-zero values. The optimal precursor tolerance was 0.01 m/z for DIA-Umpire and Siren models of block lengths 1 and 5 and grows progressively larger for increasing block lengths. This growth can be explained by the fact that as the block lengths get longer, more peaks get added to each cluster, increasing the m/z span of each cluster and the number of possible precursors that contribute to each cluster. The fact that the Siren model with a block length of 1, in which each MS1 scan is modeled individually, suggests that clustering peaks over time introduces unnecessary imprecision that confounds peaks from different precursors but have similar m/z's. The λ_1 value also affects the optimal precursor tolerance even between Siren models that have the same block length; the optimal precursor tolerance for Siren models of block length 1 where $\lambda_1 = 0$ was 0.005 m/z, while that value for the optimal Siren model was 0.01 m/z.

2.4 Discussion

Siren combines analytical ideas from Nitpick, Bullseye, and DIA-Umpire to be able to jointly infer precursor information at a scale suitable for DIA analysis and has advantages over its predecessors that allow it to perform better at precursor identification. Like Nitpick, it uses regression to jointly infer the abundances of multiple precursors per spectrum, but represents the precursors in a sparse way similar to DIA-Umpire to allow efficient scaling to the wide isolation windows of DIA.

2.4.1 Methods to optimize models

LASSO regression has been implemented using multiple methods, some of which are variations on coordinate descent and gradient descent methods[24]. A custom solver using gradient descent was implemented in C++ to perform the LASSO regression. This solver was

used rather than a pre-existing package for LASSO to allow future additions to the basic LASSO objective function to account for other kinds of structure within the data, which are described in chapters 3, 4, and 5. The structures we encoded into the model via additional regularization terms in the objective function were not all adaptable to pre-existing packages.

The solver uses different variations of gradient descent. Gradient descent in general iteratively computes the gradients of the objective function on the data with respect to the parameters and adjust the parameters by an amount related to the gradients until the gradients become sufficiently small, suggesting that the model has converged to an optimum. There exist many variations on these methods that modulate those basic steps in the interest of improving convergence time, accuracy, and computational efficiency [24]. "Batch" methods compute and sum gradients on all the data (all of the peaks) at once before parameter adjustment, whereas "stochastic" methods randomly choose a subset of the data, containing as few as one peak, before each adjustment. Method variants also prescribe different ways to compute parameter adjustments using functions of the current gradient as well as previous gradients. Some of these methods include Fast iteratve shrinkage-thresholding algorithm (FISTA) [1] and Adam [11], which have been implemented in Sirens LASSO solver.

The optimization we apply uses stochastic descent and batch descent in sequence; stochastic iterations quickly find an approximate but noisy solution, while batch iterations refine the parameters to be as close to the optimum as possible. The batch iterations are especially useful when we expect sparse data from L1 regularization, as stochastic gradient descent solutions tend to provide excess small, non-zero parameters when they should be zero [24]; an incorrectly-inferred non-zero abundance estimate of a precursor may result from a stochastic gradient based on a single theoretical peak that happened to match an observed peak while the rest of its theoretical peaks matched no observed peaks.

2.4.2 Other uses B inferred from model.

The inferred elution profiles in B can be used in many ways in both DDA and DIA analysis. The benchmarking scheme above shows how it can be used to improve the precursor isolation

window for better candidate peptide discrimination in DDA database search. B may also be used to improve DIA analysis pipelines. For instance, in DIA-Umpire, the elution profiles in B can be used instead of the precursor peak chromatograms, which are likely to have better correlations with fragment ion chromatograms due to the removal of interference. In Pecan, the inferred abundance of a precursor in a given point in the elution profile B can replace for the dot-product score between observed and theoretical isotope distributions, which is again more robust to peptide co-elution. Chapter 5 proposes a regression approach inspired by DIA-Umpire where B plays the same role as it does in DIA-Umpire.

Chapter 4 proposes a regression approach inspired by DIA-Umpire where B plays the same role as it does in Siren.

Chapter 3

BICHIR: REGULARIZED REGRESSION ANNOTATE ELUTION PROFILES AND DETECT PEPTIDE SEQUENCES FROM MS2 SPECTRA

3.1 Introduction

Peptide identification from DIA MS2 spectra is difficult because each spectrum is likely to contain fragments from multiple precursor ions from a wide precursor range. Traditional identification methods, described in Chapter 1, do not explicitly consider that fact and generally attempt to identify peptides individually. A newer method that uses LASSO to identify peptides from library spectra, called Specter[17], does jointly consider multiple peptides at once, allowing Specter to better detect and quantify peptides compared to those traditional methods. However, because Specter relies on library spectra, it only detects peptides that have been previously characterized by mass spectrometry.

We propose Bichir (short for bitonic chromatogram inference using regression), a peptide detection method that uses a LASSO approach similar to Specter to identify peptides but uses computationally predicted theoretical spectra rather than library spectra so that a wider set of peptides may be identified. Bichir includes a new form of regularization related to nearly-isotonic regression[27] to add to the LASSO formulation to account for the fact that peptides should only elute once during a mass spectrometry experiment. We provide empirical evidence that the bitonic LASSO can detect peptides missed by just the LASSO approach, but that it does not perform as well as the LASSO approach in general. Its slightly lower performance may be a result of optimization difficulties.

3.2 Methods

Bichir's linear model of MS2 spectra has the same structure as Siren in Chapter 2, but Siren's observed and theoretical MS1 spectra (Y_1 and X_1) are replaced by observed and theoretical MS2 fragment spectra in matrices Y_2 and X_2. Because the MS2 spectra in a DIA experiment cover multiple precursor windows that each samples different peptides, we can separate the spectra into multiple sets that each contains spectra from the same precursor window, and each set becomes a separate Y_2. We create a separate model for each precursor window because the windows are mostly independent, barring relationships caused by isotopic and charge variation. We propose two objective functions to infer peptide abundances B_2 from the observed spectra and theoretical spectra. The first is a formulation of LASSO[26] with the same structure as Siren or Specter[17]. The second is based off of nearly-isotonic regression[27], adding a new "bitonic" regularizer that encourages each elution profile to monotonically increase over time until it reaches an elution peak and monotonically decrease after. Elution profiles with this property are called "bitonic", and this objective function with the bitonic regularizer is called "bitonic LASSO".

$$\underset{B_2 \geq 0}{\operatorname{argmin}} \quad \|Y_2 - X_2 B_2\|_2^2 + \lambda_2 \|B_2\|_1 \tag{3.1}$$

In the first objective function (3.1), λ_2 fulfills the same role as λ_1 in the previous chapter. The balances B_2 the OLS penalty $\|Y_2 - X_2 B_2\|_2^2$, to accurately describe the data, against the L1 regularization penalty $\lambda_2 \|B_2\|_1$ to avoid overfitting.

The second objective function adds the bitonic regularizer to equation 3.1 and is written below:

$$\underset{P \geq 0,\, B_2 \geq 0}{\operatorname{argmin}} \quad \|Y_2 - X_2 B_2\|_2^2 + \lambda_2 \|B_2\|_1 + \sum_{n=1}^{N} \Big(\sum_{t=1}^{T} P_{n,t} \operatorname{bitonic}(B_{2\{n,:\}}, t) + (\|P_{n,:}\|_1 - \omega)^2 \Big) \tag{3.2}$$

$$\operatorname{bitonic}(B_{2\{n,:\}}, t) = \sum_{\tau=1}^{t-1} (B_{2\{n,\tau\}} - B_{2\{n,\tau+1\}})_+ + \sum_{\tau=t}^{T-1} (B_{2\{n,\tau\}} - B_{2\{n,\tau+1\}})_- \tag{3.3}$$

It requires an additional inferred matrix $P \in \mathbb{R}^P NxT$ that specifies the possible locations of the elution peak for each peptide ion, where N is the number of target peptides and T is the number of spectra. $P_{n,t} > 0$ if peptide n has an elution peak at time t. The intent is for $P_{n,t} > 0$ for at most one t for each n. Equation 3.3 is a generalization of nearly-isotonic regression[27]. This bitonic approach penalizes all peptide abundances that deviate from a bitonic profile that peaks at time t. The penalty assumes the existence of two nearly-isotonic sequences of parameters converging at an intermediate point, the elution peak, rather than the single nearly-isotonic sequence that Tibshirani proposes. The parameter ω controls the relative value of the penalty against the accurate description of the data, and represents the strength of the belief that peptides should elute bitonically. We choose a sufficiently large ω to ensure that $P_{n,t} >> 0$ for at least one t for all peptides n. A nonzero $P_{n,t}$ encourages peptide n to elute bitonically by with peak time t by giving the penalty bitonic $B_{n,}, t$ a nonzero contribution to the bitonic objective function. Bichir's bitonic LASSO objective function reduces to LASSO when ω is zero.

3.2.1 Construction of X_2 and Y_2

To build a model for a subset of MS2 spectra within the same precursor window, we first construct a Y_2 for those spectra and the theoretical matrix X_2 corresponding to it. Y_2's columns are the T spectra ordered by elution time. Usually, T is also the the number of MS1 spectra. X_2 contains the target database of theoretical spectra associated with the peptide sequences whose m/z's fall within the precursor range of the associated observed MS2 spectra. These theoretical spectra may be library spectra (used by Specter[17]), or provided by other means. Here we explore the use of theoretical spectra predicted by PeptideArt[13]. The OpenSWATH dataset used consists of 345 synthetic labeled peptides spiked into a yeast lysate, so the database consists of these 345 peptides combined with a yeast tryptic peptide database. We enumerate all peptide ions of charges 1–3 from the database and assign each to the X_2 associated with the set of MS2 spectra whose precursor isolation window contains the ion's monoisotopic peak. For the ions assigned to a particular MS2

sequence, PeptideArt predicts the theoretical MS2 spectra that form the columns of X_2. The peaks within PeptideArt spectra associated with labeled peptides are shifted the appropriate amount. X_2 also contains decoy spectra to provide a statistical null for downstream analyses, generated in the same way from shuffled target peptide sequences.

3.2.2 Objective and Optimization

The bitonic LASSO objective function (3.2)is not convex, so typical convex optimization methods like gradient descent do not provide the globally optimal solution. A proof of its non-convexity is below.

1. The real function f is convex if and only if $f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$
2. Let $f(B, P)$ be the objective function 3.2 on matrices B and P where $Y = [0, 1, 0]$, $X = [1]$, and $\omega = 1$
3. Let $\theta = 0.5$
4. Let $B_1 = [0, 0, 0]$, $B_2 = [0, 0, 1]$, and $B_3 = \theta B_1 + (1-\theta)B_2 = [0, 0, 0.5]$.
5. Let $P_1 = [0, 2, 0]$, $P_2 = [0, 0, 2]$, and $P_3 = \theta P_1 + (1-\theta)P_2 = [0, 1, 1]$.
6. Then $f(B_1, P_1) = 2.0$, $f(B_2, P_2) = 3.0$, and $f(B_3, P_3) = 2.75$
7. Then $f(B_3, P_3) = 2.75 \leq 2.0\theta + 3.0(1-\theta) = 2.5$, which is False
8. Therefore $f(B, P)$ is not convex.

However, the bitonic objective is biconvex, which means that the function 3.2 of B given a fixed P is convex, and function 3.2 of P given a fixed B is also convex[6]. The biconvexity allows the use of a variant of alternate convex search (ACS) [6] to efficiently find a local optimimum for function 3.2. A typical ACS algorithm would iteratively alternate

learning B and P given fixed parameters λ_2 and ω[6]. We modify the algorithm such that ω begins at 0 for the first iteration, which would cause the model to assign no elution peaks (set all values in P to zero) and infer elution profiles of any shape according to the OLS penalty. We incrementally increase ω at each iteration such that in early iterations, the model assigns elution peaks only to peptides for whom a single peak is obvious, such that those assignments might help the model in future iterations to assign peaks to peptides whose peaks are ambiguous. The algorithm terminates after a set number of iterations, where the incremental changes in ω are chosen such that all peptides have been assigned a peak by the end of the iterations. The algorithm is as follows:

1. Input: Y, X, λ_1, $P = 0$, $k \in \mathbb{Z}^+$

2. $E_k \leftarrow \{\frac{100}{k}, \frac{200}{k}, ..., \frac{100(k-1)}{k}, 100\}$

3. For e in E_k.

   (a) Fixing P, compute $B = \underset{B}{\operatorname{argmin}} \|Y - XB\|_2^2 + \lambda_1 \|B\|_1 + \sum_n^N \sum_t^T P_{n,t} \cdot \text{bitonic}(B_{n,}, t)$ using stochastic gradient descent. The terms in 3.2 that include ω are ignored because they do not affect B.

   (b) Fixing B compute $P = \underset{P}{\operatorname{argmin}} \sum_n^N \left(||P_{n,:}||_1 - \omega\right)^2 + \sum_n^N \sum_t^T P_{n,t} \cdot \text{bitonic}(B_{n,}, t)$, where ω is chosen such that for e percent of peptides n, $P_{n,:}$ contains at least one non-zero value.

The first iteration of (a) learns B given only the L1 regularized least-squares regression penalty because it starts with $P = 0$. The subsequent iterations alternate between learning P given B and learning B given P. In step (b), the process of computing the ω to assign peaks for e percent of peptides is described in section 3.2.3, where e becomes progressively larger over the iterations. We choose k to be 4 such that it takes 5 iterations to assign peaks to every peptide, and each iteration assigns peaks to 25% more of the peptides. The algorithms to perform steps (a) and (b) are below.

Learning B given P

The first step is to learn B given P by stochastic gradient descent, with a minor variation in the parameter update procedure intended to increase the speed of convergence. The optimal B subject to the bitonic penalty likely contains many examples of $B_{n,t}$ equal to an adjacent parameter $B_{n,t+1}$ because unequal values may induce a non-zero bitonic penalty. However, because the penalty is non-differentiable, gradient descent is unlikely to reach or adequately approximate that solution. Therefore, we revise the parameter update procedure so that if an update for a parameter $B_{n,t}$ would change its relationship with an adjacent value ($B_{n,t-1}$ or $B_{n,t+1}$) from greater-than or less-than (or vice versa), rather executing the update, the algorithm sets $B_{n,t}$ equal to that adjacent value. The procedure allows the algorithm to find solutions that do not violate the bitonic property, where adjacent parameters are equal to each other.

3.2.3 Learning P given B

The algorithm to learn P given B is based on the fact that for any peptide n, when the objective function is minimized, either no values of $P_{n,:}$ are non-zero, or $P_{n,t} \geq 0$ for the value(s) of t for which bitonic $B_{n,:}, t$ is minimized. The proof is below:

1. Assume we have the vector $P_{n,:}$ that minimizes 3.2.

2. Let $\hat{t}$ be the value of t such that bitonic$(B_{n,}, t)$ is minimized.

3. Assume there is another t such that bitonic$(B_{n,}, t) >$ bitonic$(B_{n,}, \hat{t})$ and $P_{n,t} > 0$.

4. If we set $P_{n,\hat{t}} \leftarrow P_{n,\hat{t}} + P_{n,t}$ and $P_{n,t} \leftarrow 0$, the value of the term $||P_{n,:}||_1 - \omega$ in 3.2 is preserved.

5. However, because bitonic$(B_{n,}, \hat{t}) <$ bitonic$(B_{n,}, t)$, the third term in 3.2 decreases, resulting in a smaller penalty. This is a contradiction to the premise that the original

$P_{n,}$ minimizes 3.2. This proves that any t for which bitonic$(B_{n,:}, t)$ is not minimized, $P_{n,t}$ must be zero.

Therefore, to minimize P, we set $P_{n,t} \leftarrow 0$ for all t that does not minimize bitonic$(B_{n,:}, t)$ given n. We then compute the values of $P_{n,t}$ for all other t and n by setting to 0 the derivatives of equation 3.2 with respect to those values:

$$\frac{\partial}{\partial P_{n,t}} \left\{ (P_{n,t} - \omega)^2 + P_{n,t} \cdot \text{bitonic}(B_{n,}, t) \right\} = 0 \tag{3.4}$$

$$P_{n,t} = \omega - \frac{1}{2} \text{bitonic}(B_{n,}, t) \tag{3.5}$$

If $P_{n,t}$ is negative, then we set it to 0, because of the non-negative constraint for P. If there are multiple values of t for which bitonic$(B_{n,}, t)$ is minimized, then we set each $P_{n,t} = \omega - \frac{1}{2d} \text{bitonic}(B_{n,}, t)$. We may compute all of the values in P individually because they are all independent of each other given a fixed B_2. To choose the correct ω for a given iteration of the optimiziation algorithm, we set ω to the value where $\omega \geq \frac{1}{2} \text{bitonic}(B_{n,}, t_n)$ for e percent of peptides n where t_n minimizes bitonic B_n, t_n.

3.2.4 Evaluating False Discovery for Peptides

To validated the elution profiles inferred by Bichir, it assigns each target and decoy peptide a score based on their learned abundances in B_2 and other features extracted from the model. Because peptide abundances vary by many orders of magnitude within a biological sample, we do not score the peptides based directly on inferred abundances in B_2 but instead use them to identify elution peaks and then score the peaks based on other features. Many existing DIA analysis algorithms such as Pecan, OpenSWATH, and DIA-Umpire score peptide elution peaks based on multiple features, and one common feature between them is the correlation of the peptide's fragment ion chromatograms around its elution peak. Accordingly, we first choose elution peaks from B_2 and score them based on fragment chromatogram correlation using the following procedure.

For each peptide n, candidate elution profiles are defined as segments within $B_{2\{n,:\}}$ that fall between local minima. A local minimum is defined as as value that is either smaller than both values adjacent to it, or is zero and is adjacent to one non-zero positive value. The elution profile with the greatest sum is considered to be the true elution profile for that peptide, and t_1 and t_2 are the elution times associated with the local minima around that elution profile. Fragment ion chromatograms are then extracted for the 10 most intense fragments within $X_{2\{:,n\}}$ from $Y_{2\{:,t_1:t_2\}}$. The score for the peptide n is the z-score from the hypothesis test that the average pairwise pearson correlation coefficient over every pair of the 10 chromatograms is greater than 0, computed using the Fisher transformation[5]. This z-score is $\frac{\sqrt{n+3}}{2} \ln \frac{1+r}{1-r}$.

To test the success of these z-scores to discriminate targets from decoys, we compute the number of target SGS peptides above a thresholds score versus the number of SGS decoys above that threshold, for every possible threshold. For peptides in the database with multiple charge states, we consider only the charge state with the highest z-score and remove the rest. The number of decoys are presented rather than traditional false discovery rates because the accuracy of these rates depend on the assumption that each peptide score is independent. However, because the elution profiles are learned jointly and affect each other, the assumption may not be true.

3.2.5 Data sets

We use publicly available DIA datasets collectively called the SWATH-MS Gold Standard (SGS) collected on an AB SCIEX TripleTOF 5600 mass spectrometer[22]. These datasets consist of three replicates each from samples containing 345 synthetic peptides spiked into water, yeast lysate, or human lysate backgrounds in concentrations varying from 30 to 0.058 fmol/μL and then sampled via relatively wide precursor isolation windows of 25 m/z. The analyses below involve single replicates of the water and yeast lysate backgrounds containing 30 fmol/μL of synthetic peptides.

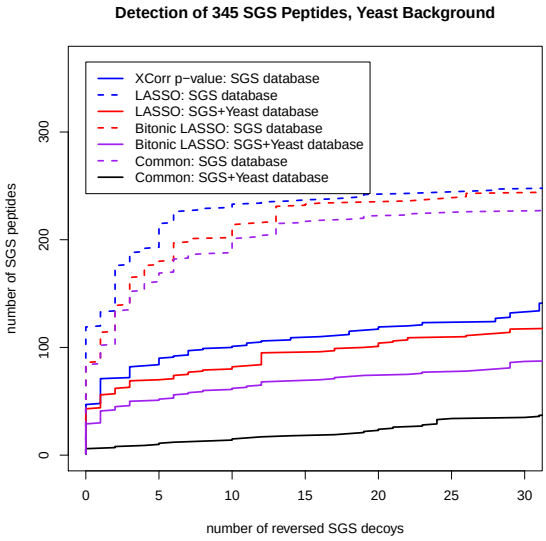

Figure 3.1: **Sensitivity of LASSO, bitonic LASSO, and XCorr p-value to SGS peptides** are plotted as a function of the number of accepted SGS decoys. Purple lines denote the SGS peptides detected by both LASSO and bitonic LASSO.

3.2.6 Peptide Identification by XCorr p-value as baseline

As a baseline against which the peptide identification performance Bichir may be compared, we perform database search on the DIA spectra using XCorr p-value[8]. We use two sequence databases: either the sequences of SGS peptides at charge 2 or those sequences combined with yeast tryptic peptides. The search's precursor window tolerance of 12.5 corresponds to the wide isolation windows of 25 m/z. We computed the performance of the database search using the same metric described for the Bichir detections.

3.3 Results

3.3.1 Identification of SGS peptides

We want to test whether the regression approaches can detect SGS peptides better than a baseline database search using XCorr p-value. In theory, because the regression jointly

considers multiple peptides to model the DIA spectra, we hypothesized that including in the database more target peptides that actually exist in the sample should improve the regression's ability to assign peptides to signal and disentangle interference. We therefore perform the search on a database of just the SGS peptides and another combining both the synthetic peptides and yeast tryptic peptides.

Both bitonic LASSO and LASSO perform better than the baseline of XCorr p-value, which is expected because XCorr p-value does not consider the time dimension. XCorr p-value has the false assumption that the spectra should contain only a single peptide because it was intended to analyze DDA spectra.

For LASSO and bitonic LASSO, a higher proportion of peptides could be confidently identified using the combined database compared to just the SGS database, meaning that the additional yeast target peptides in the combined database correctly explain much of the data that the regression models would interpret as decoy signal. 3.1.

Bitonic LASSO performs slightly worse than LASSO; at 10 false positives, bitonic LASSO identified 214/345 peptides, while LASSO identified 233/345, with 201 in common. The fact that the two methods disagree on so these peptides shows that both LASSO and Bitonic LASSO are missing true signal. The OpenSWATH algorithm (not shown) identifies more peptides (335/345) partly because of its use of library spectra that the regression algorithms and XCorr p-value do not use. The result shows that LASSO and bitonic LASSO do not necessarily perform better when more target features are included if too many false-positives are introduced.

3.3.2 Bitonic shape of elution profiles inferred by bitonic LASSO

We expect that Bichir's inferred elution profiles would be more bitonic than those inferred by LASSO. We visualize the shapes of the two sets of elution profiles learned by LASSO on a subset of the 345 SGS peptides identified at with a threshold resulting in 10 or fewer decoys in either the LASSO or bitonic LASSO-derived scores. (3.2). The LASSO inferred elution profiles appear to have many elution peaks of high intensity, where some profiles have many

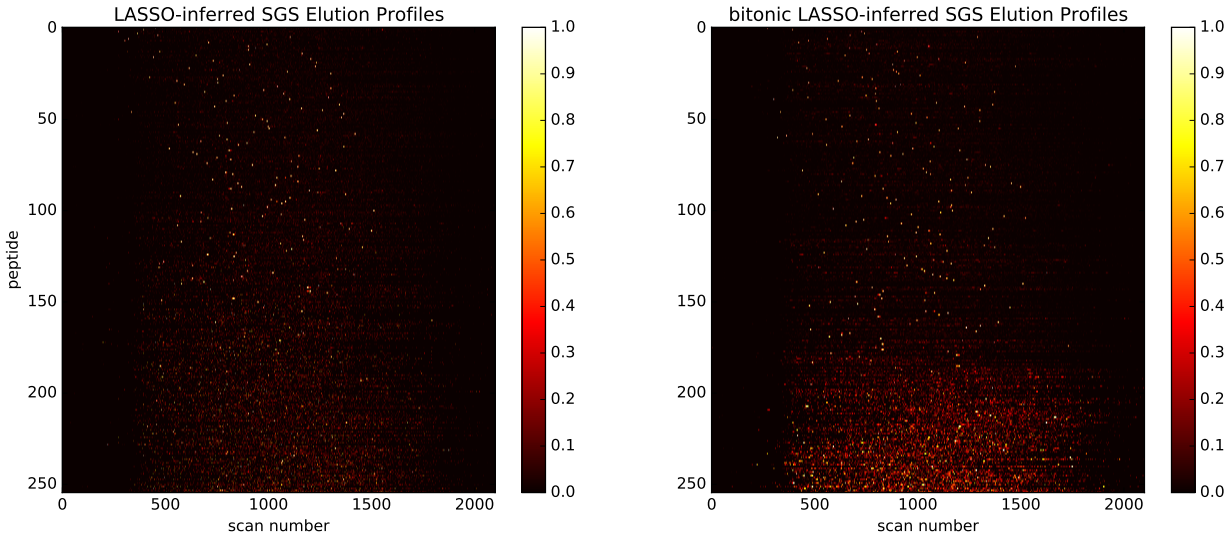

Figure 3.2: **LASSO and bitonic LASSO-inferred elution profiles** (rows in B) of 255 out of 345 SGS peptides that passed a threshold with 10 decoys in either the LASSO solutions or the bitonic LASSO solutions. These are shown as heatmaps where each row is divided by the maximum value in the row, and the square-root of the intensities are shown to improve visibility. The rows in each are sorted in ascending order according to the sum across the row after normalization.

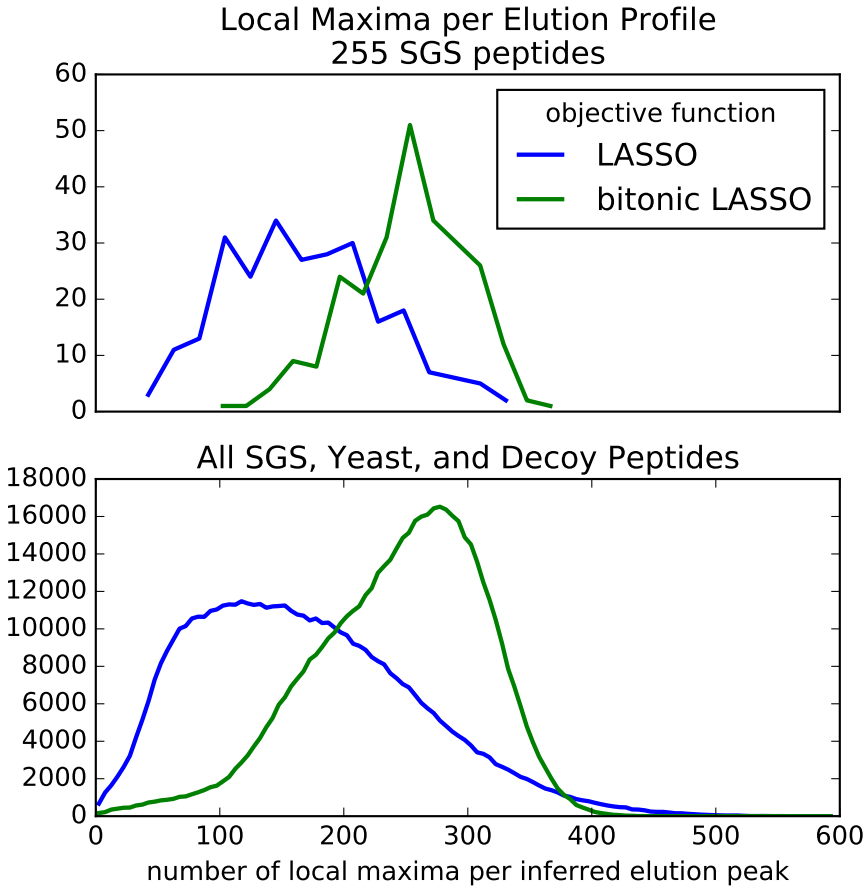

Figure 3.3: **Distributions of the number of local maxima** in each elution profile inferred by LASSO and bitonic LASSO.

more than others. The bitonic LASSO solutions appear to have a larger subset of elution profiles that each has a single, prominent elution peak, but about a quarter of them appear to have very many more. It may be that the bitonic LASSO must compensate for the lack of inferred abundance assigned to peptides with a single elution peak by assigning more peaks to other peptides.

To examine whether the visual observation that bitonic LASSO infers more peptides that exhibit single, prominent elution peaks and forces a small subset to have many small elution peaks, we compare the distributions of local maxima in the elution profiles. Ideal bitonic elution peaks should contain only one or zero local maxima in ideal conditions, and unconstrained elution peaks may contain more.

The distributions conflict with the visual observation that many peptides feature a single, prominent peak. The conflict is apparent for the subset of 255 peptides and also all of the peptides (3.3). There is a general increase in the number of inferred maxima in the bitonic LASSO elution profiles. The conflict may be explained the fact that the bitonic LASSO profiles contain many small fluctuations over time that contribute to the number of maxima but are not necessarily visually prominent. These tiny fluctations may be artifacts of the stochastic gradient descent optimization, created by the noisy parameter updates. Indeed, the bitonic LASSO-inferred B matrices have 176.5% more non-zero values than the LASSO-inferred B.

If noisy optimization is the culprit in creating non-bitonic elution profiles, then an optimization method other than stochastic gradient descent should be used. The fact that the bitonic LASSO optimization does not yet succeed in inferring bitonic profiles may explain why LASSO and bitonic LASSO detect a similar number of peptides.

3.4 Discussion

Although Bichir's bitonic LASSO does not successfully infer peptides to elute in realistic, bitonic profiles, both LASSO and bitonic LASSO identify more peptides than traditional database search. The better performance may be explained by a few differences between

the approaches. First, the regression models use better models for theoretical spectra than database search, where the PeptideArt spectra more accurately represent the relative intensities of fragment ions than XCorr p-value's uniform representations. Second, the regression models jointly model multiple peptides at once, allowing the model to better assign combinations of signals to the peptides that produced them. Third, the regression models account for variation of peak intensities over time in multiple ways, where XCorr p-value does not. Bitonic regression assumes that a peptide should only elute once, allowing it to more correctly determine the provenance of fragment peaks. This difference may also explain the slightly better performance of bitonic regression compared to LASSO. Both regression methods use the correlation of fragment intensities over time to score peptides.

However, the method OpenSWATH is able to out-perform LASSO and bitonic regression on the identification of the synthetic peptides, most likely because it uses library spectra derived from similar experiments rather than computationally predicted spectra. The use of LASSO to identify peptides using library spectra as the theoretical spectra in X_2 has been done by Specter[17] and does indeed improve identification and quantification of peptides over other methods that do not use regression. However, because these methods use library spectra, only peptides that have been previously characterized by fragment MS2 may be identified. The use of computationally predicted spectra allows the possibility for any peptide sequence to be identified.

The fact that including more target yeast spectra in the model improved the ability to detect the SGS peptides has useful implications for Bichir, Specter, and perhaps any regression method used to analyze DIA data. It suggests that the inclusion of any theoretical spectra of peptides that exist in the data, whether or not they are of interested, can help improve a regression model's ability to detect peptides that are of interest.

The use of theoretical fragment peaks makes the Bichir less general than Siren because fragment peaks depend heavily on peptide sequence whereas precursor isotope distribution does not. Bichir requires prior specification of the peptide sequences hypothesized to be present in the data, and only a subset of these are generally known. Therefore, the MS2

model is unlikely to be able to accurately explain many observed peaks. Additionally, the accuracy of the model depends on the accuracy of the theoretical spectra given to the model. Library spectra derived from experimental conditions similar to that of the observed data should produce the highest accuracy, and theoretical spectra created by computational models should produce lower accuracy.

The low accuracy of theoretical models of MS2 spectra for peptides relative to those of MS1 spectra is a result of the inherent complexity and variability of the fragmentation process. The fragmentation of a peptide depends on interactions between amino acid sequence, the type and energy of the dissociation driver, and many unknown variables that makes models unable to generalize across samples. There is no single theoretical or empirical model yet that definitively solves the problem.

Chapter 4

DEFRAGGLE: REGRESSION TO DECONVOLVE MS2 SPECTRA FROM MS1 FEATURES IN TWO STEPS

4.1 Introduction

A method called Defraggle (Deconvolution of Fragments by Group LASSO) is proposed to combine the regression approaches of Specter and the previous chapters with the analytical process used by DIA-Umpire. The combination attempts to ameliorate its predecessors' shortcomings while preserving their advantages. Specter and other regression approaches jointly model multiple peptides at once to better disentangle interference and assign peak intensity to peptides. However, such approaches require a priori knowledge of the peptide sequences and accurate models of their fragmentation that are difficult to obtain. DIA-Umpire avoids these requirements by modelling MS1 and MS2 peaks in a sequence-agnostic way, allowing its model to more fully represent the data, but does not consider all of the ion peptides jointly. Defraggle uses the sequence-agnostic approach of DIA-Umpire but performs the two steps of precursor elution inference and fragment deconvolution using regression to jointly model multiple peptides. Defraggle also improves fragment deconvolution by jointly considering complementary fragment ion and neutral-loss relationships by the addition of latent group LASSO (LGL) regularization.

4.2 Approach

To avoid the pitfalls introduced by the incomplete prior designation of peptide sequences and inaccurate modeling of theoretical spectra in the previous models of DIA MS2 spectra described in Chapter 3, the models of Siren and Chapter 3 may be combined sequentially such that the combination is agnostic to amino acid sequence. The inference of peptide sequences

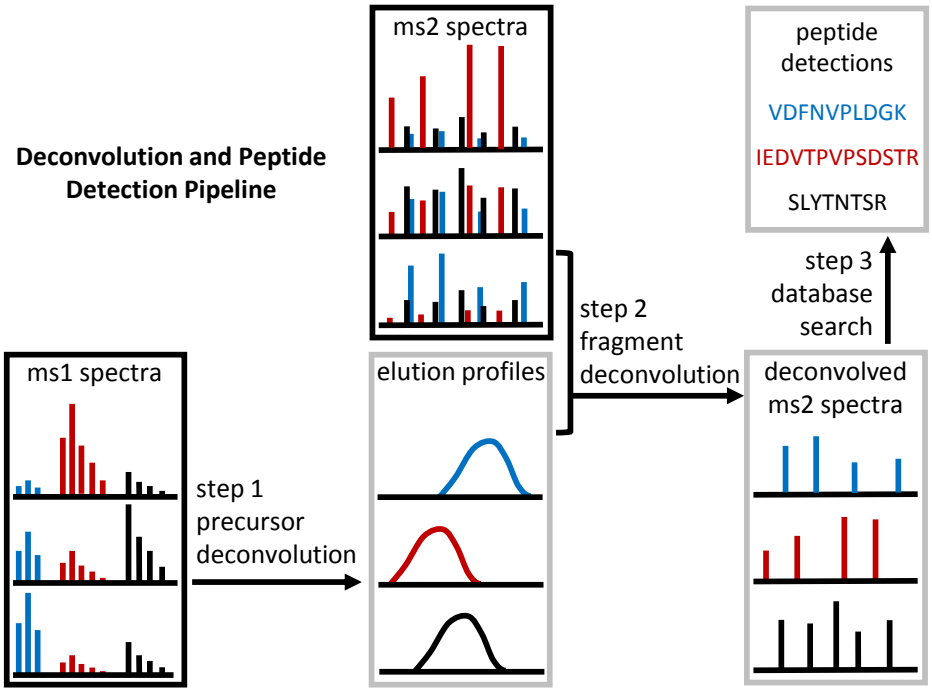

Figure 4.1: **Pipeline to sequentially deconvolve MS1 and MS2 spectra** for peptide identification. Boxes outlined in black are inputs, and boxes outlined in gray are data structures inferred by the model.

is a subsequent step after modelling the MS2 peaks. The whole pipeline (Figure 4.1) is inspired by the methods DIA-Umpire[29]and MSDIAL[31] from the field of metabolomics, which accomplishes the same steps using a similar regression applied to non-peptide metabolites.

4.3 Methods

4.3.1 Step 1. Precursor Deconvolution

Defraggle first estimates the elution profiles of the peptides represented in the sample by first examining the MS1 spectra. This first step is performed by the method described in Chapter 2, in which precursor elution profiles are inferred and reside in the learned matrix B_1. To fit these inferred precursor elution profiles into the Defraggle pipeline, the given B_1 from Siren is processed into multiple matrices, where each matrix contains the elution profiles of the precursor ions that fall within a particular precursor isolation window from which a subset of the MS2 spectra were sampled. We generically refer to the matrix for one of these isolation windows B_2. We construct each row of B_2 so that it contains an elution profile for a single precursor ion species. This property is not necessarily true in Siren's B_1, as a row in B_1 will contain the combined elution profiles of every precursor ion species with the same mass and charge. The processing of B_1 into B_2 excises the likely elution profile of each precursor ion species within a row of B_1 into its own row in B_2, setting the values that do not coincide with that species' elution profile in that row in B_2 to 0. The boundaries for the elution profiles for in B_1 are defined as local minima, where a value in a row of B_1 is a local minumum if it is less than or equal to both adjacent points and is strictly less than at least one of the adjacent points.

4.3.2 Step 2: Fragment Deconvolution

$$\underset{X_2 \geq 0}{\operatorname{argmin}} \quad ||Y_2 - X_2 B||_2^2 + \lambda_2 ||X_2||_1 \tag{4.1}$$

The inferred elution profiles are then considered in the context of the optimization func-

tion 3.1, but used in the inverse way to infer theoretical spectra from elution profiles rather than inferring elution profiles using from theoretical spectra, in the optimization function 4.1. Each row in B_2 is used by the model to infer a theoretical fragment spectrum for that peptide species in its corresponding column in X_2. Prior knowledge about peptide fragmentation can be added via additional regularization of function 4.1 to assist in deconvolution and is described below.

Latent group LASSO to deconvolve MS2 spectra

To overcome the model's inability to distinguish between peptides with similar, overlapping elution profiles, we incorporate knowledge we have about the fragmentation chemistry producing the MS2 spectra; when molecules of a single peptide species fragment at a particular spot in its backbone, both fragments are often ionized and appear together in the MS2 spectrum, and the masses of this pair of fragments add up to the mass of the intact whole. Fragment ions called b and y-ions share this relationship, as well as other pairs of ions depending on the method of fragmentation. Additionally, other fragments called neutral losses are produced when small molecules such as H2O and NH3 break from the pair of fragments. We refer to each pair of fragments and its associated neutral losses as a complementary fragment group, and Defraggle considers the co-occurrence of the fragments within these groups using a regularizer called latent group LASSO (LGL) [16] to aid in MS2 deconvolution.

LGL is a variant of group LASSO[33] (GL) that, like GL, prioritizes the inclusion of features that share membership in the same groups. In Defraggle, features are fragment ion peaks that are grouped by complementary fragment relationships because peptide fragmentation often produces these types of ions together. Each group is the set of complementary ions that result from fragmentation at a particular site in a particular peptide's backbone. LGL differs from GL by allowing a feature to belong to multiple groups and separately accounting for contributions by multiple groups to the same feature. LGL is therefore more appropriate than GL when modelling fragment ion peaks because fragments from multiple fragmentation sites may contribute to the same peak. The groupings are encoded in the

structure of a tensor $W \in \mathbb{R}^{MxFxN}$ whose values represent the inferred contribution of each fragment group to each peptide's theoretical spectrum. W is a factorization of X_2 whose relationship to X_2 is described below.

$$X_{2\{:,n\}} = \sum_{f=1}^{F_n} W_{\{n,:,f\}} \text{ for all } n \in \{1, 2, ..., N\} \tag{4.2}$$

W contains an $M \times F_n$ matrix $W_{:,:,n}$ (Figure 4.2) for each precursor n in the model, where M is the number of fragment m/z bins and F_n is the number of possible fragmentation sites for all peptide sequences that have the same mass as n given an alphabet of amino acids and modifications. We only include the canonical 20 amino acids with the mass modification for cysteine, but other modifications may be added. These fragmentation sites are enumerated using a dynamic programming algorithm described in the section below. The column vector $W_{:,f,n}$ contains either zero or nonzero values corresponding to the intensities of the different fragment types that result from fragmentation at site f, including the b-ion, y-ion, water and ammonia losses for both, and a carbon monoxide loss for the b-ion. Depending on the particular amino acid sequence of a peptide, certain neutral losses may not be appropriate, but Defraggle does not currently capture sequence-specific constraints. Other ion types may be included or excluded in the future depending on the experimental fragmentation type. Elements in $W_{:,f,n}$ that do not correspond to an included fragment type are necessarily zero. Only the charge +1, monoisotopic forms of the fragments are included, and the inclusion of higher charge states or isotopic variants for the fragment ions may be beneficial in the future.

$$\underset{W \geq 0}{\operatorname{argmin}} \quad ||Y_2 - X_2 B||_2^2 + \lambda \sum_{n=1}^{N} ||B_{n,:}||_2^2 \sum_{f=1}^{F_n} \left(\sum_{m=1}^{M} W_{f,m,n}^2 \right)^{\frac{1}{2}} \tag{4.3}$$

The penalty added to the OLS penalty in equation 4.3 penalizes sum of the magnitudes of all the fragment groups, which leads to an inferred W where peak intensity is segregated into fewer fragment groups; having more intensity in fewer fragment groups leads to a lower penalty than having less intensity in more groups. That fact means that optimal solutions for

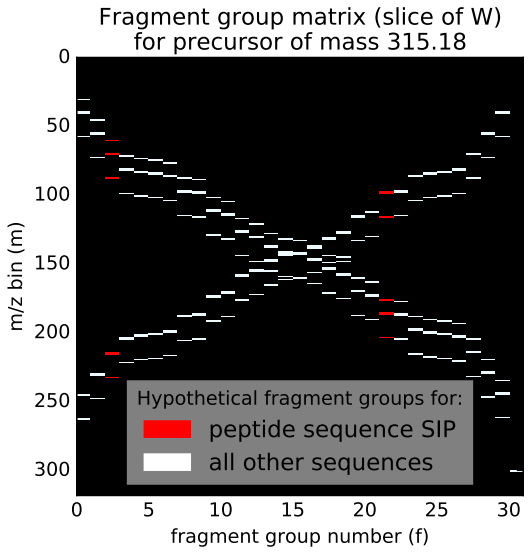

Figure 4.2: **Slice of W** depicting the fragment groups for a precursor of mass 315.18. Each column is a group, and the position of the nonzero values in the column denote the ion m/z's that correspond to a hypothetical fragmentation site for a precursor of the given mass.

LGL will contain more peaks that co-occur in their fragment groups and fewer peaks that are the only members of their group than solutions for LASSO. The λ scales the strength of the LGL penalty and represents the strength of the prior belief that fragment ions must appear in complementary groups. The term $||B_{n,:}||_2^2$ scales the penalty such that the fragment peaks for peptides of low abundance are not underestimated relative to those of more abundant peptides. The underestimation would occur because the OLS penalty for underestimating the fragment peaks in X_2 for a peptide of low abundance is smaller than the OLS penalty for peptides of higher abundance, but the opposing LGL penalty is the same for all peptides unless it is scaled.

Model optimization may optionally consider complementary fragment ion relationships, in which extra terms that decompose theoretical fragment spectra into its constituent ion types are added to the objective function. The pieces of the model Y_2, X_2, and B_2 are the same as those described in Chapter 3, with the addition of the tensor W.

4.3.3 Definition of fragment groups and ions in W

Algorithm 1 Algorithm to enumerate the discretized m/z values of all possible b-ions for all possible peptide ions of a given mass

$\mu_b \leftarrow$ discretized mass of the precursor minus the mass of water
$\alpha \leftarrow$ set of discretized masses of allowed amino acids and any modified forms
$\beta \leftarrow$ boolean matrix of dimension $\mu_b \times \mu_b$ filled with 0s
$\beta_{0,0} \leftarrow 1$
for m_1 in $\{0,...,\mu_b\}$ **do**
    if $\beta_{m_1,0} == 1$ **then**
        for a in α **do**
            $m_2 \leftarrow m_1 + a$
            if $m_2 \leq \mu_b$ **then**
                $\beta_{m_2,m_1} \leftarrow 1$
                for i in $\{0,...,m_1\}$ **do**
                    if $\beta_{m_1,i} == 1$ **then**
                        $\beta_{m_2,i} \leftarrow 1$

To define the fragment groups for a peptide of a given mass within W, Defraggle enumerates the masses of all possible b-ions for all peptide sequences of that mass and defines a fragmentation site for each b-ion. Algorithm 1 uses dynamic programming to enumerate all possible fragmentation sites for all peptides b-ions. Once the algorithm terminates, $\beta_{\mu_b,i} = 1$ for all m/z bins i that contain a b-ion from the set of all possible b-ions for all possible peptides of mass $\mu_b +$ mass of water , and $\beta_{\mu_b,j} = 0$ for all m/z bins j that contain no b-ions from those peptides.

Defraggle then constructs W based on β. For a peptide n with discretized mass $\mu_b +$ mass of water, $W_{:,:,n}$ has a column f for each i where $\beta_{\mu_b,i} = 1$. $W_{k,f,n}$ is allowed to be nonzero where k corresponds to the m/z bin of the b-ion of mass i or one of the b-ion's

complementary fragments. All other values in $W_{:,f,n}$ are must always be zero.

The resolution of the fragment m/z bins must be relatively large (1.0005079 m/z) because of rounding errors. For the algorithm to be correct, the sum of the discretized masses of two b-ions must equal the discretized sum of those two b-ions. If the bin width for that discretization is too small, that rule is violated.

4.3.4 Simulating complex DIA spectra to test Defraggle

Defraggle's deconvolution was tested on simulated sets of spectra, each constructed by adding together two or more discretized theoretical MS2 spectra produced by PeptideArt[13] at various relative ratios. In these simulations, peptide abundances B_2 and theoretical spectra X_2 were simulated and used to create simulated observed spectra $Y_2 \leftarrow X_2 B_2$. In some cases, accompanying theoretical MS1 spectra X_1 were created using the averagine model as described in Chapter 2, and observed MS1 spectra were created as $Y_1 \leftarrow X_1 B_1$, where $B_2 = B_1$. Some sets of simulated spectra contained either peptides with nonzero abundances over multiple scans to simulate DIA spectra, and others contained peptides with non-zero abundance only in a single scan, removing variation over time. Both of these cases were achieved by setting the values with B_2 in the corresponding way, described below.

The sets of spectra where peptides are non-zero only in single scans were created to better characterize Defraggle's ability to deconvolve spectra purely based on precursor mass and fragment group relationships, excluding the variation of intensity over time. In these sets, pairs or triplets of PeptideArt-generated theoretical spectra from randomly chosen peptides from a yeast database were added together at the same (Figure 4.3) or different scales. PeptideArt spectra were created for N peptides randomly selected from a yeast tryptic peptide database of charge 2 (from 728-741 m/z) and placed in matrix X_2. The designation of these peptides into pairs or triplets mixtures was achieved by filling each column $B_{2\{:,t\}}$ with two (pairs) or three (triplets) non-zero values and setting the simulated $Y_2 \leftarrow X_2 B_2$ (Figure 4.3). Each peptide was a part of only one pair or triplet. Defraggle was then used to infer $\hat{X}_2$ from (Y_2, B_2) given the masses of the peptides in X_2 multiple times using various

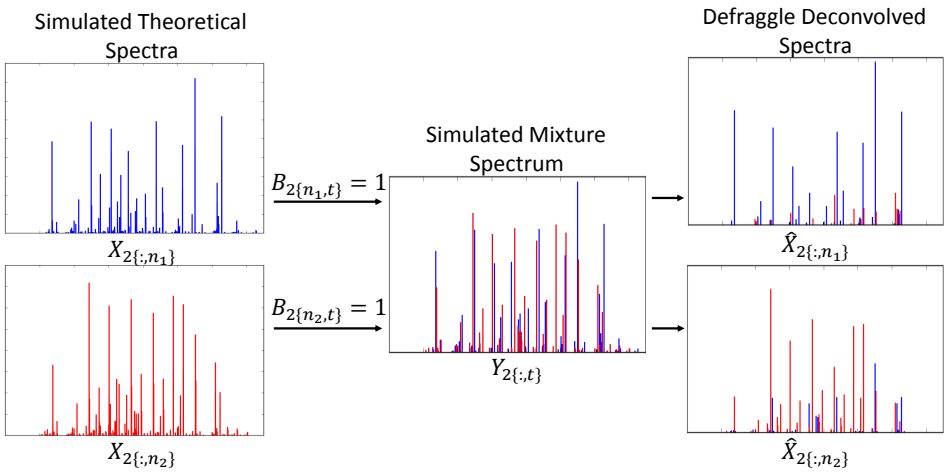

Figure 4.3: **Simulation of mixture spectra and deconvolution by Defraggle.** Columns $n1$ and $n2$ of X_2 are mixed into the simulated mixture spectrum t contained in column t of Y_2 by assigning non-zero values to their abundances in B_2 at corresponding to spectrum t. Defraggle then deconvolves the mixture spectrum into estimates of its constituent parts in columns $n1$ and $n2$ of $\hat{X}_2$. The colors of the peaks correspond to the theoretical spectrum from which the peaks or portions of peaks came.

values of λ_n.

The rest of the spectra sets were created to simulate DIA datasets, where each set contained a sequence of T MS2 spectra in Y_2. X_2 contained PeptideArt spectra derived from N peptides randomly selected from a yeast tryptic peptide database of charges 1-3 that fell within the m/z range 728-741. An elution profile (a row in B_2 and B_1) was generated for each peptide n using a Gaussian distribution with a mean drawn uniformly from 0 to T, a standard deviation drawn uniformly from 0 to 1.5, and scaled by a random value chosen uniformly from 0 to 1. The value of $B_{2\{n,t\}}$ is the value of the Gaussian distribution for peptide n at time t, with values less than 0.001 set to zero, resulting in each peptide having non-zero abundance in 8-16 spectra. To test both steps of Defraggle on this data, Defraggle was used to B_1 as $\hat{B}_1$ given (Y_1, X_1) and then used to estimate X_2 as $\hat{X}_2$ from $\hat{B}_1$ given Y_2.

4.3.5 Evaluating deconvolved spectra

We evaluated the accuracy of Defraggle's deconvolution of Y_2 into the estimated $\hat{X}_2$ by two measures; one measuring the overall accuracy of the deconvolution to recover the theoretical spectra from the mixtures, and another measuring the improvement in identifiability by XCorr p-value of the true peptide sequence on the deconvolved spectra relative to the mixtures. The first measure, for a peptide n for which $B_{2\{n,t\}} > 0$, is the ratio of two sum squared error (SSE) terms: the SSE between the peptide's deconvolved spectrum $\hat{X}_{2\{:,n\}}$ and its corresponding theoretical spectrum $X_{2\{:,n\}}$ and the sum squared-error between the the corresponding mixture spectrum $Y_{2\{:,t\}}$ and the theoretical spectrum. The SSE ratio is $\frac{||\hat{X}_{2\{:,n\}} - X_{2\{:,n\}}||_2^2}{||Y_{2\{:,t\}} - X_{2\{:,n\}}||_2^2}$. Before the SSE ratios are computed, the columns of Y_2, X_2, and $\hat{X}_2$ are scaled such that their 2-norms all equal 1 to correct for the decrease in absolute intensity caused by LGL regularization. The second measure for a peptide n that is non-zero at time t is the log-ratio between the XCorr p-values of the match between the true peptide sequence and the mixture spectrum $Y_{2\{:,t\}}$ and the match between the true peptide sequence and the deconvolved spectrum $\hat{X}_{2\{:,t\}}$. We use XCorr p-value, a representative of a database search score function, because database search is the final step of the pipeline to detect peptides.

The higher this value, the greater the improvement in peptide identifiability provided by the deconvolution. The un-normalized XCorr p-value of the matches between a peptide sequence and the simulated or deconvolved spectra are also examined, as the actual XCorr p-value determines whether a match can be considered statistically significant.

4.4 Results

4.4.1 Deconvolution of individual mixed spectra

Defraggle deconvolves spectra based on their variation over time in addition to knowledge of precursor mass and fragment ion relationships. To understand Defraggle's ability to use precursor mass and fragment ion relationships without considering variation of signal over multiple scans, we first give Defraggle only a single spectrum for each peptide, where each spectrum is a mixture of two theoretical spectra at equal proportions.

Our evaluation of the quality of the Defraggle's deconvolutions differs depending on which of the two performance measures we apply. Performance measured by SSE ratio varies with λ and is optimal at an intermediate value of λ (Figure 4.4). However, its performance on the peptide-spectrum match score only decreases with λ, where the the XCorr p-value increases with greater λ. This suggests that XCorr p-value matching is worsened by the liberal removal of noise/contaminant peaks if that requires the removal of true signal as well. Defraggle's success at improving the SSE ratio does not translate into better peptide detection.

A particularly successful deconvolution with respect to SSE ratio on a pair of peptides (Figure 4.3) using the overall optimal $\lambda = 0.3749$,

achieves SSE ratios of 0.45 for $\hat{X}_{2\{:,n_1\}}$ and 0.56 for $\hat{X}_{2\{:,n_2\}}$. Visual inspection of the peaks show that most of the peaks originating from the wrong spectrum are greatly diminished or removed entirely by Defraggle.

However, because peptides do not all exist at the same abundance, we created and deconvolved pairs of simulated peptides at different relative abundances. Defraggle's deconvolution does not improve peptide detectability in pairs where both peptides are present in signif-

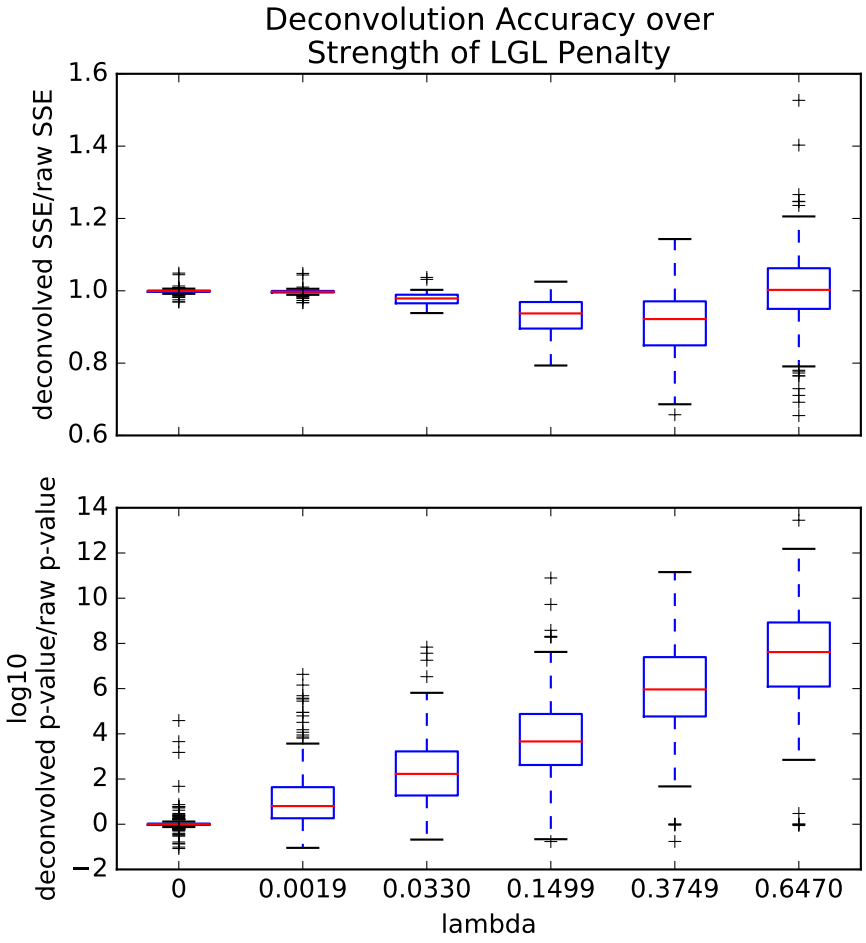

Figure 4.4: **Defraggle's deconvolution accuracy.** 100 pairs of PeptideArt spectra each combined into a mixture spectrum were deconvolved by Defraggle with varying values of λ, and two performance measures described in Methods are shown in boxplots across those values of λ.

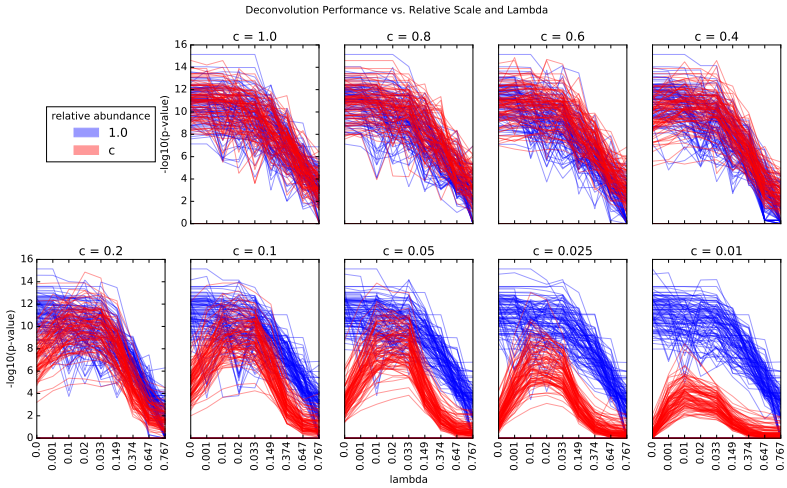

Figure 4.5: **Defraggle's deconvolution accuracy.** 100 pairs of PeptideArt spectra were deconvolved by Defraggle with varying values of λ, and two performance measures described in Methods are shown in boxplots across those values of λ. When $\lambda = 0$, the objective function reduces to OLS, and when $\lambda > 0$, the LGL penalty contributes to the solution.

icant amounts, but for peptides present at 20% abundance or lower, a non-zero value of λ improves XCorr p-values. The same non-zero λ also provides the best improvement for triplets of peptides at various relative intensities (Figure 4.6)

To test Defraggle's ability on even more complex spectra, it was used to deconvolve mixtures of triplets of peptides at three different relative abundances that differ from each other by factors of five (Figure 4.6. Again, we see that XCorr p-values can improve for peptides at 20% and 5% abundance using a non-zero λ, but the λ with the best overall improvement is greater than that in the pair mixtures. This may suggest that the optimal value of λ is dependent on the complexity of the sample.

4.4.2 Deconvolution of mixed spectra over time

To test how Defraggle works when it considers variation over time, the method was applied to a simulated DIA dataset of 500 peptides eluting in 500 scans and with an objective term that

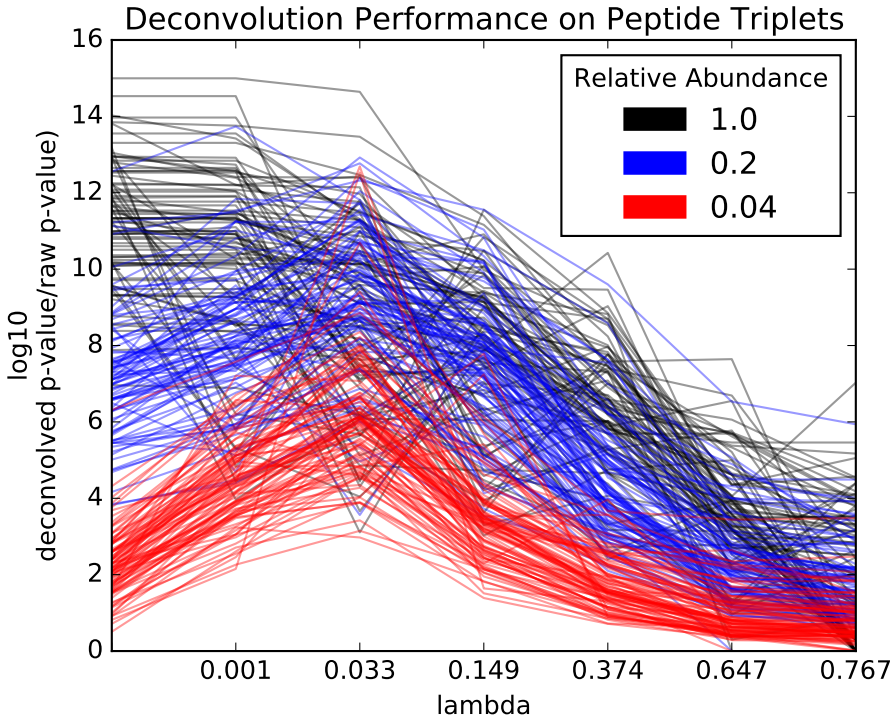

Figure 4.6: **Defraggle's deconvolution accuracy.** 100 pairs of PeptideArt spectra were deconvolved by Defraggle with varying values of λ, and two performance measures described in Methods are shown in boxplots across those values of λ. When $\lambda = 0$, the objective function reduces to OLS, and when $\lambda > 0$, the LGL penalty contributes to the solution.

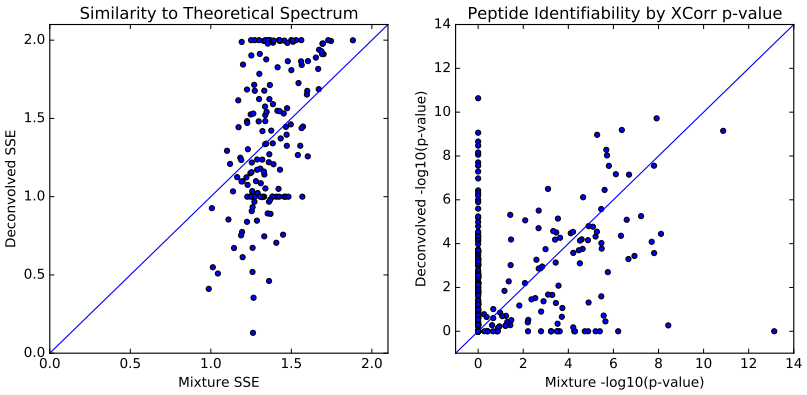

Figure 4.7: **OLS model and Deconvolution Accuracy.** 500 DIA spectra Y_2 were simulated as the elution of 500 simulated peptides as X_2B_2, and Defraggle using just the OLS penalty without L1 regularization or LGL regularization was used to infer $\hat{B}_2$ and $\hat{X}_2$. The first scatterplot shows SSEs of the deconvolved spectra against the SSEs of the mixture spectra, both compared to the true theoretical spectra in cases where the inferred precursor elution peak in B_1 was accurate to within two scans. The second scatterplot shows the Xcorr p-values of $\hat{X}_2$ vs the values in X_2 matched against the true peptide sequence.

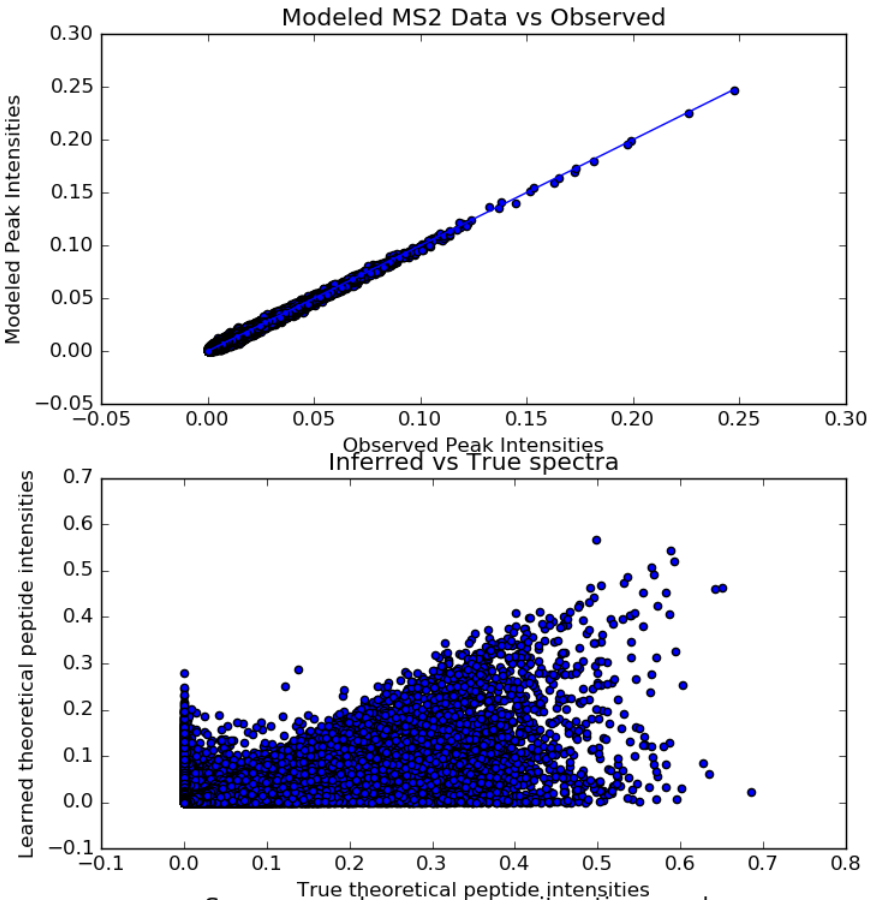

Figure 4.8: **OLS Model and Deconvolution Accuracy.** 500 DIA spectra Y_2 were simulated as the elution of 500 simulated peptides as X_2B_2, and Defraggle using just the OLS penalty without L1 regularization or LGL regularization was used to infer $\hat{X}_2$. The first scatterplot shows the values in $\hat{X}_2B_2$ vs the values Y_2. The second scatterplot shows the values in $\hat{X}_2$ vs the values in X_2.

only uses the OLS penalty. No L1 regularization was used here to understand how the model can reproduce observed peaks without the underestimation bias L1 regularization produces, and no LGL regularization was used to understand how well Defraggle can deconvolve spectra by just considering variation over time.

The modeled peaks $\hat{X}_2 B$ very closely matched the simulated observed peaks Y_2 (Figure 4.8), but the deconvolved spectra $\hat{X}_2$ did not closely resemble X_2. The performance of Defraggle to deconvolve the individual colums of X_2 is mixed, where some deconvolved spectra are more similar to theoretical spectra and some are not, and some XCorr p-values are improved and some are not: 170 of the deconvolved spectra provide better XCorr p-values than the mixture spectra, and 78 of the deconvolved spectra provide better XCorr p-values than the deconvolved spectra. For the rest of the 500 peptides, they did not get matched to their corresponding spectra among the top 5 peptides in the tryptic yeast database on which the spectra were searched (Figure 4.7). Defraggle's ability to recover theoretical is mixed even when it succeeds at its objective of reproducing observed peaks.

The model's ability to reproduce the observed peaks without capturing the true peptide spectra that created it can be explained by numerical reasoning; when the rows of B (the elution profiles) are not linearly independent, there is no unique solution to the OLS optimization problem. The rank of the matrix B was 472, which is less than the number of rows in B (500), which means that the rows are collinear and not linearly independent. In other words, the elution profiles are so correlated that one may be approximated as the linear combination of others.

Because the elution profiles are not different enough from each other to for Defraggle's regression to use them to fully deconvolve the spectra, we tested LGL regularization with a non-zero value of λ so that Defraggle may use fragment group information to help with deconvolution. However, performance did not significantly improv (data not shown). The lack of improvement may be caused by the improper implementation of stochastic gradient descent to optimize Defraggle when peptides elute in more than one scan.

4.5 Discussion

Defraggle builds on regression methods and DIA-Umpire to jointly model and deconvolve peptides in MS1 and MS2 peaks in a sequence-agnostic way while taking into account fragment ion relationships, and its potential is demonstrated in simulations. Defraggle can deconvolve mixture spectra given only precursor information to help identify peptides of low abundance, but its utility in DIA data is not yet demonstrated, likely due to technical issues of the optimization algorithm for LGL. Once these technical issues are solved, it is likely that Defraggle can be useful in DIA experiments to better identify peptides for which library spectra do not exist.

The best amount of latent group LASSO regularization λ was only found empirically by grid search, and it likely depends on particular data; the mixture spectra used in this study consisted of theoretical spectra which were designed so that their fragments roughly did appear in their predictable groupings. Real data may not follow the groupings as well, so the best value of λ may be lower. Future investigations in how to choose the correct λ are necessary, as well as tweaking which fragments should be included in the groups.

Further progress may also be made in how spectra are discretized. The development of Siren from Chapter 2 depended heavily on the details of the flexible binning scheme, and Defraggle's fragment deconvolution currently relies on fixed bin widths. These bin widths do not necessarily describe the peaks at its best resolution, and doing so may prove to improve the quality of the deconvolution.

Chapter 5

CONCLUSION

An ideal of mass spectrometry-based proteomics is the ability to detect and quantify every protein in a sample. Because this goal has not yet been achieved, proteomic experiments are limited to studying biology through the small subset of proteins that can be detected and quantified. Increases in the throughput and precision of mass spectrometers in the past decade have established a key prerequisite of this goal, which is the the ability to fragment and measure every ion species in the sample using the DIA sampling scheme. However, this turning point in hardware and firmware development has yet to realize the goal because many of the spectra produced are still uninterpretable. There are two overarching problems that create the uninterpretability.

This thesis attempts to solve the first problem, which is the mixing of the signals from the different ion species that obscures their identities. The ubiquity of unresolved signal interference in current analyses indicates that there is still much progress to be made in solving the first problem, and there are clear paths for further development of the regression approaches presented in this thesis to make progress. For instance, additional prior knowledge may be more appropriately encoded into the objective functions, like refinement of fragmentation models in LGL regularization and better tuning its parameters. The problem may be made easier by improvements in mass spectrometer hardware or firmware such that the spectrometer attempts to fragment every ion species individually into more DDA-like, unmixed spectra. However, some mixed spectra are unavoidable given that chromatography cannot provide perfect separation, and interpretting those spectra will require software improvements.

The second problem is more fundamental: identifying every peptide requires comprehen-

sive but narrow hypotheses of what they might be. Many current analyses use hypotheses (target databases and library spectra) of what peptide sequences are in the spectra and fragmentation models that predict the peaks they produce. However, the fact that many spectra cannot be identified (the "dark matter" of proteomics), even in DDA experiments where most spectra are unmixed, shows that the solution to the second problem is incomplete. The study that introduces the score function MSFragger, whose main appeal is an improvement in speed rather than quality, implies that this dark matter contains a diverse array of conventionally unrecognized post-translational modifications (PTMs) of peptides, and that better score functions will not help to identify most of them[12]. The same dark matter is also present in DIA data, the methods presented in this thesis to analyze DIA data ultimately suffer from dependence upon an incomplete database; Bichir requires theoretical spectra, and the last step of Defraggle is a database search. The straightforward solution of adding every plausible PTM to the set of hypotheses does not suffice because of false positives and computational complexity. The number of combinations of PTMs exponentially increases with the number of PTMs considered, providing many more chances for incorrect matches to score highly.

The MSFragger study used a variant of database search called open search that avoids hypothesizing specific variants, instead using the conventional hypotheses of unmodified tryptic peptides but allowing for a wide range of precursor mass error[12]. In many fortunate cases, open search's wide errors allow the database search to match non-specified modified versions of peptides to their unmodified versions if enough of their fragments match. The avoidance of specifying particular variants helps to identify more of the dark matter and gives an overview of the diversity of conventionally unrecognised PTMs, but the avoidance prevents open search from properly interpreting many fragment peaks such that it cannot identify all of the dark matter.

An approach called de novo sequencing also avoids hypothesizing specific peptide variants by generating hypotheses at the more basic level of amino acid subsequences[15]. De novo sequencing can interpret many of the subsequences contained in the dark matter, but these

more basic hypotheses typically do not have the same power to identify the full peptides that are included in a database search.

The knowledge that would narrow down hypotheses to only the variants most likely to be true is the identity of the PTM each peptide contains. The knowledge may be provided by future innovations in chemical protocols to label the signals. Alternatively, data analytical innovations may highlight the information already present in the data. Open search may provide the first steps to narrow down the possible peptide variants in a modified database search pipeline. Further developments in another approach called spectral clustering, which searches for pairs of peptides whose masses and fragment masses differ by a mass consistent with a particular chemical variation, may reveal enough about the peptide variants[4].

Progress is being made in solving both the problem of mixed spectra and the problem of insufficient knowledge of peptide variants that leads to uninterpretable dark matter. Continued development in both of these directions will likely bring proteomics much closer to the ideal of detecting and quantifying every peptide in a sample, allowing a more complete understanding of biology.

BIBLIOGRAPHY

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2:183–202, 2009.

[2] M. Bern, G. Finney, M. R. Hoopmann, G. Merrihew, M. J. Toth, and M. J. MacCoss. Deconvolution of mixture spectra from ion-trap data-independent-acquisition tandem mass spectrometry. *Analytical Chemistry*, 82:833–841, 2010.

[3] R. Bruderer, O. M. Bernhardt, T. Gandhi, S. M. Miladinovic, L. Y. Cheng, S. Messner, T. Ehrenberger, V. Zanotelli, Y. Butscheid, C. Escher, O. Vitek, O. Rinner, and L. Reiter. Extending the limits of quantitative proteome profiling with data-independent acquisition and application to acetaminophen-treated three-dimensional liver microtissues. *mcp*, 14:1400–1410, May 2015.

[4] J. A. Falkner, J. W. Falkner, A. K. Yocum, and P. C. Andrews. A spectral clustering approach to MS/MS identification of post-translational modifications. *Journal of Proteome Research*, 7(11):4614–22, 2008.

[5] A. K. Gayen. The frequency distribution of the product-moment correlation coefficient in random samples of any size drawn from non-normal universes. *Biometrika*, 38:219–247, 1951.

[6] J. Gorski, F. Pfeuffer, and K.Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, 66(3):373–407, Lune 2007.

[7] V. Granholm, J. F. Navarro, W. S. Noble, and L. Käll. Determining the calibration of confidence estimation procedures for unique peptides in shotgun proteomics. *Journal of Proteomics*, 80(27):123–131, 2013.

[8] J. J. Howbert and W. S. Noble. Computing exact p-values for a cross-correlation shotgun proteomics score function. *Molecular and Cellular Proteomics*, 13(9):2467–2479, 2014.

[9] E. Hsieh, M. Hoopmann, B. Maclean, and M. J. MacCoss. Comparison of database search strategies for high precursor mass accuracy MS/MS data. *Journal of Proteome Research*, 9(2):1138–1143, 2009.

[10] A. Keller, S. L. Bader, D. Shteynberg, L. Hood, and R. L. Moritz. Automated validation of results and removal of fragment ion interferences in targeted analysis of data-independent acquisition mass spectrometry (ms) using swathprophet. *Molecular and Cellular Proteomics*, 14:1411–1418, 2015.

[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 2015.

[12] A. T. Kong, F. V. Leprevost, D. M. Avtonomov, D. Mellacheruvu, and A. I. Nesvizhskii. MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nature Methods*, 14(5):513–520, 2017.

[13] S. Li, R. J. Arnold, H. Tang, and P. Radivojac. On the accuracy and limits of peptide fragmentation spectrum prediction. *Analytical Chemistry*, 83(3):790–796, 2011.

[14] B. MacLean, D. M. Tomazela, N. Shulman, M. Chambers, G. L. Finney, B. Frewen, R. Kern, D. L. Tabb, D. C. Liebler, and M. J. MacCoss. Skyline: an open source document editor for creating and analyzing targeted proteomics experiments. *Bioinformatics*, 26(7):966–968, 2010.

[15] T. Muth and B. Y. Renard. Evaluating de novo sequencing in proteomics: already an accurate alternative to database-driven peptide identification? *Briefings in Bioinformatics*, 2017. Epub ahead of print.

[16] G. Obozinski, L. Jacob, and J. P. Vert. Group lasso with overlaps: the latent group lasso approach. *arXiv*, 2011.

[17] R. Peckner, S. A. Meyers, J. D. Egertson, R. S. Johnson, J. G. Abelin, S. A. Carr, M. J. MacCoss, and J. D. Jaffe. Specter: linear deconvolution as a new paradigm for targeted analysis of data-independent acquisition mass spectrometry proteomics. *bioRxiv*, 2017.

[18] A. C. Peterson, J. D. Russell, D. J. Bailey, M. S. Westphall, and J. J. Coon. Parallel reaction monitoring for high resolution and high mass accuracy quantitative, targeted proteomics. *mcp*, 11:1475–1488, 2012.

[19] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes*. Cambridge UP, 1986.

[20] B. Y. Renard, M. Kirchner, H. Steen, J. A. Steen, and F. A. Hamprecht. NITPICK: peak identification for mass spectrometry data. *BMC Bioinformatics*, 9:355, 2008.

[21] A. L. Rockwood, S. L. Van Orden, and R. D. Smith. Rapid calculation of isotope distributions. *Analytical Chemistry*, 67:2699–2704, 1996.

[22] H. L. Röst, G. Rosenberger, P. Navarro, L. Gillet, S. M. Miladinovic, O. T. Schubert, W. Wolski, B. C. Collins, J. Malmstrom, L. Malmstrom, and R. Aebersold. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. *Nature Methods*, 32(3):219–223, 2014.

[23] M. W. Senko, S. C. Beu, and F. W. McLafferty. Determination of monoisotopic masses and ion populations for large biomolecules from resolved isotopic distributions. *Journal of the American Society for Mass Spectrometry*, 6(229–233), 1994.

[24] S. Shalev-Shwartz and A. Tewari. Stochastic methods for l1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011.

[25] Andrew B Stergachis, Brendan MacLean, Kristen Lee, John A Stamatoyannopoulos, and Michael J MacCoss. Rapid empirical discovery of optimal peptides for targeted proteomics. *Nature Methods*, 8(12):1041–1043, 2011.

[26] R. J. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.

[27] R. J. Tibshirani, H. Hoefling, and R. Tibshirani. Nearly isotonic regression. *Technometrics*, 53(1):54–61, 2012.

[28] Y. S. Ting, J. D. Egertson, J. G. Bollinger, B. Searle, S. H. Payne, W. S. Noble, and M. J. MacCoss. PECAN: a library free peptide detection tool for data-independent acquisition tandem mass spectrometry data. *Nature Methods*, 14(9):903–908, 2017.

[29] C.-C. Tsou, D. Avtonomov, B. Larsen, M. Tucholska, H. Choi, A.-C. Gingras, and A. I. Nesvizhskii. DIA-Umpire: a comprehensive computational framework for data-independent acquisition proteomics. *Nature Methods*, 12(3):258–264, 2015.

[30] C.-C. Tsou, C.-F. Tsai, G. C. Teo, Y.-J. Chen, and A. I. Nesvizhskii. Untargeted, spectral library-free analysis of data-independent acquisition proteomics data generated using orbitrap mass spectrometers. *Proteomics*, 16:2257–2271, 2016.

[31] H. Tsugawa, T. Cajka, T. Kind, Y. Ma, B. Higgins, K. Ikeda, M. Kanazawa, J. VanderGheynst, and O. Fiehn and. Ms-dial: data-independent ms/ms deconvolution for comprehensive metabolome analysis. *Nature Methods*, 12:523–526, 2015.

[32] C. R. Weisbrod, J. K. Eng, M. R. Hoopmann, T. Baker, and J. E. Bruce. Accurate peptide fragment mass analysis: multiplexed peptide identification and quantification. *Journal of Proteome Research*, 11:1621–1632, 2012.

[33] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society B*, 68(1):49–67, 2006.