Improving peptide detection in mass spectrometry-based proteomics

Andy Lin

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

Doctor of Philosophy

University of Washington

2021

Reading Committee:

William S. Noble, Chair

Michael J. MacCoss

Walter L. Ruzzo

Program Authorized to Offer Degree:
Department of Genome Sciences

University of Washington

Abstract

Improving peptide detection in mass spectrometry-based proteomics

Andy Lin

Chair of the Supervisory Committee:
Professor William S. Noble
Department of Genome Sciences

Over the last 30 years, the field of computational mass spectrometry-based proteomics has made great strides. Specifically, the development of database search engines has allowed for the automatic annotation of observed spectra. In addition, the application of target-decoy competition for the purposes of estimating the false discovery rate of a set of peptide-spectrum matches has been instrumental for improving the statistical evidence for a set of confidently detected peptides.

While great advances have been made, additional progress is still possible. This work describes three methods for improving computational proteomics methods. The first method describes a new database score function, combined p-value, that aims to take advantage of two advances in database searching: high-resolution MS/MS spectra and statistical calibration. The next method presents a variant of the target-decoy competition process for estimating the false discovery rate. Specifically, this variant is applicable when a subset of peptides in a sample are relevant to the hypothesis being asked. Finally, the last method describes MS1Connect, which measures the similarity of a pair of proteomics runs for the goal of inferring metadata of proteomics runs. Metadata is information about data. For example, given some data, metadata would include information regarding who generated the data and how the data was generated. Metadata is critical for the proper analysis of proteomics data but often it is missing or incorrect. Therefore, methods are needed that

can predict metadata of proteomics data. As part of this method, we have also developed MS1Connect, a new score for measuring the similarity of a pair of mass spectrometry runs. We demonstrate that this score can be used for accurate metadata inference of species labels for mass spectrometry runs.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

GLOSSARY

DDA: data-dependent acquisition

DIA: data-independent acquisition

FDP: false discovery proportion

FDR: false discovery rate

LC-MS/MS: liquid chromatography-tandem mass spectrometry

MS/MS: tandem mass spectrometry or tandem mass spectrum

MS1: precursor mass spectrum

MS2: tandem mass spectrum

M/Z: mass-to-charge ratio

PSM: peptide-spectrum match

TDC: target-decoy competition

ACKNOWLEDGMENTS

Bill Noble has been an incredible mentor, teacher, and PI during my time in graduate school. Thank you for putting up with my blatant interest in forensics.

I would like to thank all the collaborators I have worked with during my time at the Department of Genome Sciences. I could have never finished any of my projects without your help. Thanks to Jeff Howbert for helping me develop the combined p-value score function, Deanna L. Plubell for running samples for me, and Uri Keich for teaching me the details of false discovery rate and target-decoy competition.

To the members of committee, thank you for your advice and for letting me defend. In particular, I would like to thank Jeff Bilmes, who taught me a lot about machine learning, and Mike MacCoss, for your help with proteomics.

I would also like to acknowledge all the members of the MacCoss, Villén, and Bruce labs for their insight. I apologize for all the times I bothered you all in person.

In addition, I would like to thank all the current and past members of the Noble lab. In particular I would like to thank the proteomics crew of Wout Bittremieux, Yang Y. Lu, William E. Fondrie, and Lindsay K. Pino.

To all the people at GS beer hour, thanks for making graduate school more fun. In particular, I would like to thank the quarantine beer hour crew for making Friday nights during the pandemic bearable.

Finally, to my parents and sister, thank you for being patient with my journey through graduate school.

DEDICATION

To all the people who have helped me over the last five and a half years, thank you.

Chapter 1

INTRODUCTION

Proteins are the functional unit of the biological world. While DNA is the blueprint of life, proteins are the physical objects that perform the critical functions of life. A protein consists of a linear chain of amino acids that can fold into complex 3D structures. In general, there are 20 different canonical amino acids that bond together to form a protein.

Proteomics is the large scale study of proteins. Studying proteins is important because the presence and abundance of various proteins allow insight into various biological phenomena. For example, comparing the protein abundances of diseased samples against wild-type samples allows us to better comprehend the mechanism of that disease.

One of the tools used in the field of proteomics is liquid chromatography-tandem mass spectrometry. In this analysis, proteins are extracted from samples and then subjected to enzymatic digestion to create peptides. The most common digestion enzyme used is trypsin, which cleaves the peptide backbone bond following arginine and lysine residues. Following the digestion step, the peptide mixture is separated by liquid chromatography and then directly injected into a mass spectrometer to be analyzed by mass spectrometry. The field of computational mass spectrometry-based proteomics aims to analyze the data output of a mass spectrometer.

1.1 Computational methods are essential for mass spectrometry-based proteomics analysis

A goal in computational mass spectrometry-based proteomics is to identify the peptide that generated a specific spectrum. A spectrum is the output of a mass spectrometer instrument,

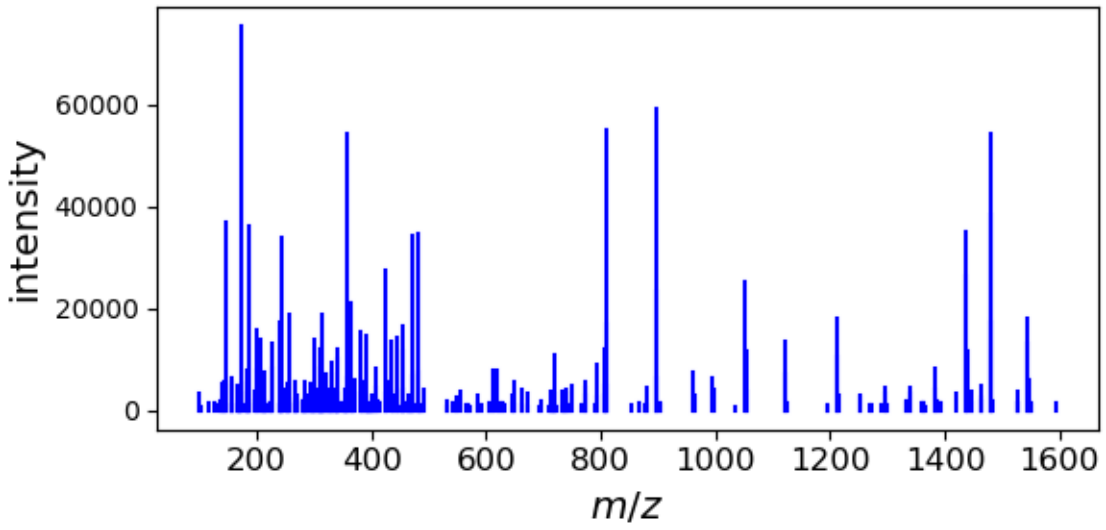


Figure 1.1: **Example of mass spectrum** An example of a tandem mass spectrum collected at 47.84 minutes where the x-axis is m/z and the y-axis is intensity.

and a single run can contain ten of thousands of spectra. An individual spectrum can be considered to be a bag of ion peaks where each peak is associated with the following tuple of values: intensity, m/z, and retention time (Figure 1.1). Each peak typically corresponds to either a peptide or a peptide fragment. The intensity of an ion is loosely associated with the abundance of that ion. The m/z, or mass to charge ratio, is simply the mass of the ion, in Daltons, divided by its charge. Finally, the retention time is the time, typically in seconds, when the mass spectrometer detects an ion. Retention time is affected by biophysical properties of the ion, such as size and hydrophobicity, since samples are typically separated by liquid chromatography prior to being injected into the instrument.

There are two types of spectra: MS1 and MS2. The peaks in MS1 spectra correspond to entire peptides whereas the peaks in MS2 spectra correspond to peptide fragments. Peaks in MS1 spectra are typically used to quantify the abundance of a peptide by calculating the area under the curve of the peak, with respect to intensity and retention time. On the other

hand, peaks in MS2 spectrum are typically used to identify the peptide that generated that particular experimental MS2 spectrum.

1.1.1 Database searching infers which peptide generated which spectra

The database search step is one of the most critical steps in proteomics data analysis because it identifies the peptide that generated each experimental MS2 mass spectrum. While researchers are typically more interested in downstream analyses, such as detecting changes in protein quantitation across experimental conditions, this analysis is not possible without first conducting a database search. As a result, a large number of database search engines has been developed.[94]

The two inputs given to a database search are a set of experimental spectra and a user defined set of peptide sequences. This peptide database consists of all the peptides that are reasonably expected to be found in the sample and typically consists of the proteome of the species being analyzed. In addition, this database is typically augmented by contaminant proteins that are artificially introduced into the sample during sample preparation. For example, human keratin proteins are often found in a non-human sample.

In a database search, spectra are searched against a database of peptide sequences. For each spectrum, a list of candidate peptides are extracted from the database. A peptide makes the candidate list if the peptide mass and observed precursor mass associated with the MS/MS spectrum match within some tolerance. Then, the search engine scores the experimental spectrum against the peptide list, resulting in a list of scored peptide-spectrum matches (PSMs). The top-scoring PSM for each spectrum is retained. A PSM with a good score means that the observed spectrum is more likely to be generated by that peptide. In practice, the major difference between various search engines is how they score a PSM.

A PSM is scored by measuring how similar the observed spectrum is to a theoretical spectrum. The theoretical spectrum is derived directly from the peptide sequence. While the exact details differ among different search engines, the spectrum historically contains peaks corresponding to every possible singly-charged b- (peptide prefix) and y-ion fragment

(peptide suffix).

1.1.2 False discovery rate control assigns statistical confidence to a set of PSMs

Following a database search, a user is left with a list of PSMs and their scores. A large number of these PSMs are incorrect because, by random chance, the best scoring peptide did not generate the observed spectrum. PSMs with low scores are more likely to be incorrect while PSMs with high scores are more likely to be correct. By setting a score threshold, we can select a subset of PSMs that are enriched for correct PSMs. Specifically, a score threshold is set that balances statistical power and the number of incorrect PSMs.

The field of proteomics sets this threshold by controlling for the number of incorrect PSMs by controlling the false discovery rate (FDR).[19] The false discovery rate controls the expectation of the false discovery proportion, where a false discovery corresponds to an incorrect PSM. The false discovery proportion is the number of incorrect accepted PSMs out of the total number of accepted PSMs while the expectation can be considered like an average. A set of PSMs controlled at a 5% FDR means that we expect about 5% of PSMs to be incorrect. In practice, researchers define an FDR threshold, typically 1%, and find the score threshold where the set of PSMs above that threshold has an FDR of 1%. Then, that list of PSMs is accepted as confident discoveries and further analyzed.

Since it is impossible to know whether a particular PSM is correct or incorrect, we estimate the FDR of a PSM set using target-decoy competition.[19] In target-decoy competition, a set of spectra are searched against a concatenated database of target and decoy peptides. Target peptides are the user defined set of peptides expected to be found in the sample while decoy peptides are shuffled or reversed versions of the target peptide. The target-decoy competition process aims to model the score distribution of incorrect PSMs with decoy PSMs. Since decoy peptides are shuffled target sequences, these peptide sequences will not be present in the sample. As a result, the decoy PSM score distribution gives us an empirical null distribution. This empirical null distribution, in turn, should match the score distribution of incorrect target PSMs.

For target-decoy competition to properly control the FDR, two criteria must be met: the independence assumption and the equal chance assumption.[19,27] The independence assumption states that spectra must be independent from each other. Given two spectra, one spectrum being matched to a peptide is independent of the other spectra being matched to the same peptide. The equal chance assumption states that an incorrect PSM will equally likely match to a target or decoy peptide. Without this assumption the decoy PSM score distribution is unable to model the incorrect target PSM score distribution.

Functionally, the FDR of a set of PSMs is estimated as follows. The set of experimental spectra are searched against a concatenated target-decoy database. The best scoring PSM for each spectrum is collected and all the PSMs are ranked by score. After setting some score threshold s, the FDR is calculated by

$$\widehat{\mathrm{FDR}}(s) \coloneqq \min\left(1, \frac{D(s)+1}{T(s)}\right), \tag{1.1}$$

where $D(s)$ is the number of decoy PSMs with score $\geq s$ and $T(s)$ is the number of target PSMs with score $\geq s$. Note that the min operation is used to prevent the estimated FDR from exceeding 100%. In addition, the +1 in the numerator is used to ensure the FDR estimate is not liberally biased.[50]

1.2 Organization of dissertation

The remainder of this thesis describes three different projects that advances the field of computational proteomics. The next chapter describes a new score function for scoring PSMs. Specifically, this score function combines two different advances in database search scoring, statistical calibration and high-resolution MS/MS spectra, together into a single score function. Chapter three focuses on estimating the FDR of a set of PSMs when a subset of the peptides in the sample are relevant to the hypothesis. The standard FDR estimation process assumes all the peptides in a sample are relevant to the biological hypothesis being asked. We present a variant of the FDR estimation to account for situation when only a subset of the peptides in the same are relevant. Next, chapter four contributes a method for inferring

missing sample metadata. Knowing the metadata associated with a sample is critical for properly analyzing proteomic data. Our method infers metadata of mass spectrometry runs by measuring the similarity of a query run to a repository of runs. Repository runs that are highly similar to the query run are expected to have the same metadata as the query run. Therefore, we can use the metadata of the similar highly run to infer metadata of the query. As part of this, we have also developed a new method for measuring the similarity of proteomics runs. Finally, concluding thoughts are given in chapter five.

Chapter 2

COMBINED P-VALUE: A NEW SCORE FUNCTION FOR HIGH RESOLUTION MS/MS DATA

This chapter is adapted from the following work:

A. Lin, J. J. Howbert, and W. S. Noble. Combining high-resolution and exact calibration to boost statistical power: A well-calibrated score function for high-resolution MS2 data. *Journal of Proteome Research*, 17:3644–3656, 2018.

2.1 Introduction

In the analysis of protein tandem mass spectrometry data produced in a bottom-up fashion using traditional, data-dependent acquisition, the database search step is critical. In this step, each observed spectrum is assigned to a peptide sequence drawn from a given database, and the resulting peptide-spectrum match (PSM) is assigned a score. Ideally, a "good" score implies that the peptide was likely to have been responsible for generating the observed spectrum. Of course, in the context of a typical scientific study, the end goal is usually downstream of the PSMs—e.g., to detect and quantify proteins, or to characterize proteins whose quantification changes across experimental conditions. However, none of these downstream steps can be accomplished if the database search step fails. Furthermore, although in principle *de novo* approaches can help to identify some observed spectra, in practice *de novo* approaches do not approach the power of database search strategies to detect hundreds or thousands of peptides in a given complex mixture.[66] Consequently, a database search engine forms the backbone of most shotgun proteomics analysis pipelines.

Given the importance of database search, it is not surprising that dozens of search engines have been developed since the advent of the first such search tool, SEQUEST, in 1994[22]

(reviewed by[68]). The algorithms employed by all of these engines are remarkably similar to each other. For each spectrum, the search engine extracts from the given database all peptides whose masses fall within a user-specified tolerance of the inferred precursor mass associated with the observed spectrum. Each of these candidate peptides is then scored against the observed spectrum, and the top-scoring peptide is reported as a PSM. Thus, in practice, the defining characteristic of any search engine lies in the details of its peptide-to-spectrum score function. The history of development of shotgun proteomics search engines can be seen primarily as a history of development of PSM score functions.

Some new score functions are driven by technology. For example, over the past decade, the resolution at which tandem mass spectra can be efficiently collected has improved dramatically. Typical data sets offer fragment ion resolution in the range of 5–10 ppm, compared to the ∼1 Da resolution that was common a decade ago. This improved resolution means that score functions designed for low resolution data did not necessarily generalize well to higher resolutions. Consequently, new score functions, such as the one in Morpheus,[98] have been explicitly designed to make good use of high-resolution mass accuracy.

On the other hand, some new score functions are driven by conceptual advances. A notable trend was the introduction of score functions that aim to achieve good calibration.[1,31,44] We say that a score function is *calibrated* if a score of x assigned to one spectrum has the same meaning or significance as a score of x assigned to a different spectrum. In practice, many PSM score functions are not well calibrated with respect to spectra, that is, they tend to assign systematically different scores to different spectra. For example, many score functions will assign a different range of scores to spectra with +2 charge versus +3 charge. Performing database search with such a function yields a loss of statistical power.[37] One way to improve the calibration of a given score function is to compute, for a given spectrum, the distribution of scores for all possible peptides. The cumulative density function of the resulting distribution then provides a well calibrated score, called a p-value. MS-GF+ demonstrated how to carry out this style of calibration using a dynamic programming procedure,[44] and a similar approach was adopted subsequently by RAId_aPS[1] and Tide.[31]

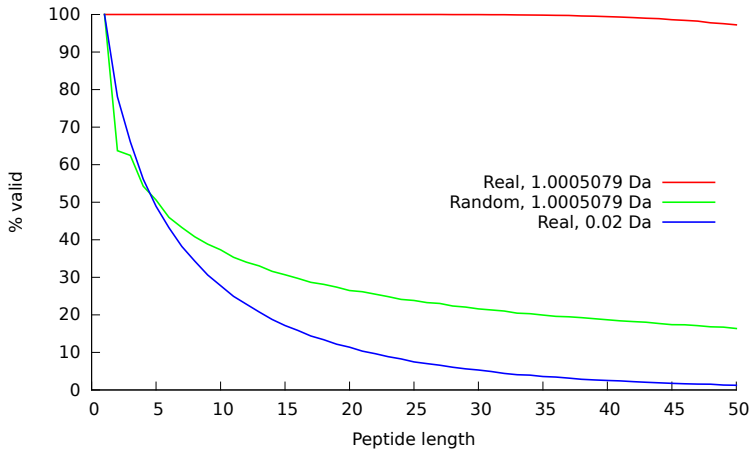


Figure 2.1: **Why dynamic programming cannot be performed using high-resolution mass bins.** The figure plots, as a function of peptide length, the proportion of randomly generated peptide sequences that obey Equation 2.1. The two series marked "Real" use monoisotopic masses from the 20 real amino acids; the series marked "Random" uses masses that have a random number in the range $\langle 0, 1]$ added to each mass. Two bin sizes (1.005079 Da and 0.02 Da) are used. For each series, a total of 100,000 peptides were simulated. With small bin sizes, the large proportion of peptide sequences whose masses violate Equation 2.1 causes the dynamic programming to fail.

Unfortunately, a signficant drawback to the dynamic programming calibration procedure is that it typically breaks down when employed in conjunction with data generated using high-resolution fragment mass accuracy. To explain the problem, it is first necessary to outline how the dynamic programming algorithm works. Any dynamic programming procedure involves building up a table of solutions to problems of increasing size. In the case of the procedures employed by MS-GF+, RAId_aPS, and Tide, the entry in row i and column j of the dynamic programming table contains a count of the total number of peptide sequences whose (discretized) mass is i and whose (discretized) score with respect to the current spectrum is j. The procedure works by filling in values in this table for increasingly large values of i and j, computing each new value by summing over existing entries in the table (see Methods for details).

The problem arises in the discretization of the mass axis. The logic associated with filling in the table requires sequentially adding together discretized amino acid masses. For this sequential summation to work properly, it must be the case that, for any given peptide, the sum of the discretized amino acid masses equals the discretized sum of the amino acid masses. More concretely, using a bin size of w and for a peptide consisting of amino acids $a_1, a_2, \ldots, a_n$, it must be the case that

$$\text{round}((a_1 + a_2 + \ldots a_n)/w) = \text{round}(a_1/w) + \text{round}(a_2/w) + \cdots + \text{round}(a_n/w). \quad (2.1)$$

Whether and how frequently Equation 2.1 is violated depends upon properties of the amino acid masses and the size of the bins used to discretize the mass axis. It is easy to see that, for arbitrary amino acid masses and peptides of reasonable length, Equation 2.1 will frequently be violated ("Random" series in Figure 2.1). On the other hand, when we use real amino acid masses and a bin size of $\sim$1 Da, short peptides uniformly obey Equation 2.1. This is because peptide masses are naturally discrete. Longer peptides do occasionally break the rules because peptide masses are not perfectly discrete; however, for peptides of the size typically considered by shotgun proteomics, this rule-breaking is quite rare. The story changes, however, when we modify the bin size used to discretize the fragment m/z axis.

Because such bins do not align well with the natural discreteness of the peptide mass axis, Equation 2.1 is violated quite frequently. Consequently, the dynamic programming procedure ends up spreading the counts associated with peptides of mass i among bins in rows $i-1$, i, and $i+1$. We are thus left with a conundrum: we have two techniques to achieve improved statistical power—using increased resolution on the fragment m/z axis or using dynamic programming to achieve good score calibration—but we cannot use both of these techniques simultaneously.

One solution to this problem is to modify the score function. MS-GF+ does this by creating a score function that takes into account both the intensity of each observed peak as well as its participation in a pair of peaks with a mass difference equal to the mass of an amino acid. The latter term, which is implemented as a weight associated with a given peak, can be computed in high resolution, even if the peaks themselves are scored in low resolution. The dynamic programming procedure can then be carried out in the usual fashion, simply by incorporating these weights.

In this work, we propose an alternative solution. We begin with the XCorr score function, which was included in the very first search engine, SEQUEST, and continues to be used in SEQUEST and a variety of other search engines, including Comet,[21] Tide,[12] and RAId_aPS.[1] However, rather than modifying the XCorr score function to take into account high-resolution mass information, we create a new score function, the "residue-evidence" (res-ev) score, that considers pairs of peaks, similar to the MS-GF+ approach. We then score each observed spectrum twice: once with the low-resolution XCorr score that focuses on individual peaks, and once with the high-resolution score that focuses on pairs of peaks. We use dynamic programming to convert each of these scores to p-values, and we employ a previously described method to estimate the p-value for the product of dependent p-values.[6]

In this work, we demonstrate that the new res-ev score function provides improved performance on some high-resolution data sets, and that the res-ev and XCorr score functions are complementary to one another. Finally, we demonstrate that the combined XCorr+res-ev p-value yields state-of-the-art performance across a variety of data sets, outperforming MS

Amanda and Morpheus and performing comparably to MS-GF+, despite having no trainable parameters. The combined p-value and res-ev p-value score functions are available in the Tide search engine, which is part of the Crux toolkit (`http://www.crux.ms`).

2.2 Methods

2.2.1 The XCorr score

The XCorr score function was first described in 1994 as part of the SEQUEST search engine[22] and is still in use today in the commercial SEQUEST product from Thermo Scientific as well as search engines such as Comet,[21] Tide,[12] X!Tandem[11] and RAId_aPS.[1]

In our implementation, the first part of the XCorr score involves preprocessing the observed spectrum in six steps, as follows:

1. Peaks with m/z values within 1.5 Th of the precursor m/z are eliminated.

2. The mass axis is discretized by creating a vector O of mass bins with bin width = 1.0005079 Da. Each bin O_i is assigned an intensity value, which is the maximum of the intensities of observed peaks whose masses fall within the mass range of O_i. The bin width is chosen to match the natural quasi-integer masses of peptides and peptide fragments, which in turn derive from the quasi-integer masses of the primary constituent elements of peptides (C, H, N, O, S).

3. The intensity of each bin O_i is replaced by its square root.

4. Peaks with intensity less than 5.0% of the maximum intensity peak are eliminated from the spectrum.

5. O is divided into 10 equal length segments, and the intensities within each segment are normalized so the maximum intensity in the segment is 50.

6. A scaled version of O is subtracted from itself at each position across a defined window of offsets: $\hat{O}_i = O_i - \frac{1}{151}\sum_{\tau=-75}^{75} O_{i-\tau}$

Next, the preprocessed observed spectrum $\hat{O}$ is converted to an "evidence vector" E. This conversion employs "quasi-integer" mass bins, where the bin size of 1.0005079 Da is selected to match the empirical distribution of peptide mass values. Each bin E_i specifies the cumulative evidence for cleavage at some hypothetical position on the backbone of the precursor peptide. More precisely, E_i holds the weighted sum of all intensities in $\hat{O}$ whose mass is consistent with a cleavage producing a b ion with quasi-integer mass $m_b = i$:

$$E_i = \sum_{m \in I} w_m \cdot \hat{O}_m \tag{2.2}$$

where I are the quasi-integer masses of the b, y, and neutral loss ions consistent with m_b and n, and $w_m = 1$ for b- and y-ions and $w_m = 0.2$ for neutral losses. We consider neutral losses of carbon monoxide (CO, also known as "a-ions"), ammonia (NH_3) and water (H_2O) groups. If the precursor charge assigned to the spectrum is > 2, then fragments with charge > 1 are possible. In this case all peaks are replicated into higher-mass bins with the appropriate mass. In particular, for a precursor of charge n, fragment ions up to charge $n - 1$ are considered. In most implementations of XCorr, if two or more predicted peaks fall in the same mass bin, then the intensity in that bin is the maximum of those peaks' intensities. However, in order to facilitate calibration via dynamic programming, we wish to make the XCorr score function fully additive. Consequently, Equation 2.2 uses the sum of the peak intensities rather than the maximum.[31]

We then predict a very simple theoretical spectrum from the sequence of the peptide. For this step, we use a discrete mass vector B, again using a bin width of 1.0005079 Da. For each possible backbone fragmentation of the peptide, B is populated with a single binary marker at the mass of the corresponding b ion.

The final XCorr score is a simple dot product between the evidence vector E and the theoretical spectrum B. Thus, the score is essentially the sum of the evidence for all the cleavage events across the length of the peptide.

2.2.2 The residue evidence score

The computation of the residue evidence score proceeds in the same steps outlined above for XCorr: preprocessing of the observed spectrum, aggregation of evidence, generation of a theoretical spectrum, and calculation of a score based on the evidence and the theoretical spectrum. Unlike the XCorr processing, which aggregates evidence into a vector indexed by quasi-integer mass, the res-ev score aggregates evidence in a matrix indexed by mass and amino acid (Figure 2.2A).

The residue evidence score preprocessing employs a subset of the steps previously described for XCorr.

1. Peaks with m/z values within 1.5 Th window of the precursor m/z are eliminated.

2. The intensity of each peak is replaced by its square root.

3. Peaks with intensity less than 5.0% of the maximum intensity peak are eliminated from the spectrum.

4. The spectrum is divided into 10 equal length segments, and the intensities within each segment is normalized so the maximum intensity in the segment is 50.

5. The intensities are rank normalized, such that the peak with rank i is assigned an intensity of $1/i$.

There are two important changes in this preprocessing compared to that for the XCorr score function. First, discretization of the peaks' mass values is deferred until after the high-resolution residue evidence has been quantified and aggregated. Second, the final subtractive step, which induces a cross-correlation penalty in the XCorr score function, is omitted altogether. Another difference is the addition of two peaks, representing the mass of the N-terminal group and the mass of the precursor minus the C-terminal group (typically a hydroxyl group). Both of these peaks have intensities of zero.

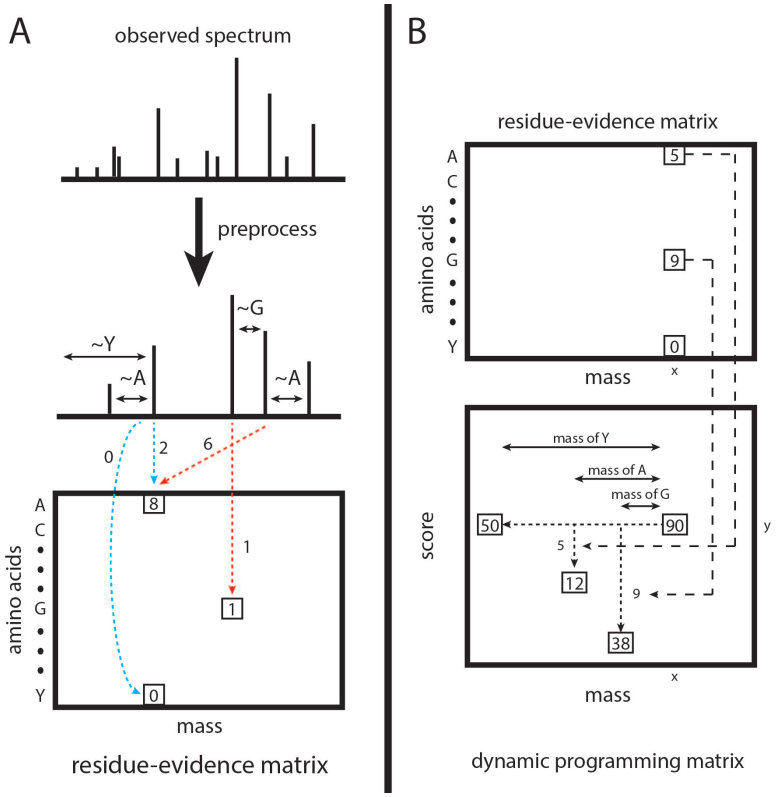


Figure 2.2: **Calculation and calibration of the res-ev score.** (A) The observed spectrum is preprocessed similarly as for XCorr. For each possible pair of peaks, it is determined whether the pair gives evidence that a peptide fragment ends with a particular amino acid. The evidence for each amino acid being terminal at each nominal mass bin is accrued in a matrix of residue evidence. Blue dotted lines represent evidence that is accrued when the peaks are assumed to be 1+ b ions. Red dotted lines represent evidence accrued when the peaks are assumed to be 1+ y ions. However, evidence is added to the residue-evidence matrix indexed by its corresponding charge 1+ b ion, so that all evidence related to a particular fragmentation is aggregated together. (B) Each cell in the dynamic programming matrix represents the number of possible peptide sequences, given the spectrum, with a particular mass and score. In our example, there are 90 possible peptide sequences with a mass of x and a score of y. Dynamic programming is carried out by, for each cell, summing the values of ∼20 previously calculated cells (corresponding to the number of amino acids). In our mock example, the cell labeled 90 was the result from summing the cells labeled 50, 12, and 38. We choose the cells to be summed as follows. Assume that a peptide fragment ends in a glycine. From the cell labeled 90, we find the mass bin that corresponds to the mass after subtracting the mass of glycine (x - 57). From the residue-evidence matrix, we know how much evidence there is that a peptide fragment with mass x ends in a glycine (in this case, 9). Therefore, we subtract 9 from the score (y - 9). This leads up to the cell that is labeled 38. Therefore, we know that 38 peptide fragment sequences could result in a specific score and mass. This process is done for the remaining amino acids. In our diagram, we only show the process for 3 amino acids.

Next, we quantify and aggregate evidence for each type of residue inducing a b-ion cleavage at each possible mass bin. The residue evidence is defined as follows. Let an arbitrary pair of MS2 peaks A and B have measured masses m_A and m_B, such that $m_A > m_B$ and the difference in mass $m_{diff} = m_A - m_B$. We say evidence *exists* for a (charge 1+) b ion fragment with mass m_A, terminating in amino acid residue X, if the deviation $\text{abs}(m_{diff} - m_X) < m_{tol}$, where m_{tol} is the maximum deviation tolerated between m_{diff} and m_X. In practice, m_{tol} is on the order of the mass spectrometer's MS2 resolution. The *magnitude* r assigned to this residue evidence is scaled so as to reward small deviations:

$$r = \max(0, 1 - \mathsf{abs}(m_{diff} - m_X)/m_{tol}) \tag{2.3}$$

i.e., r takes a value of 1 when the deviation is 0, and 0 when the deviation is equal to or greater than m_{tol}. This magnitude r is then multiplied by the sum of the rank intensities of the two peaks, prior to being stored.

Residue evidence is stored in a two-dimensional *residue evidence matrix* R. The columns of R are indexed by discretized masses m_j, and the rows of R correspond to the amino acids a_i found in the peptide database (typically around 20 rows). The increment of evidence r generated according to Eq. 2.3 is added to element $R_{a_i m_j}$, where $a_i = X$ and m_j is obtained by discretizing m_A with a bin width $W = 1.0005079$ Da.

Each pair of peaks A and B is also considered as a putative pair of charge 1+ y ions, charge 2+ b ions, and charge 2+ y ions, and additional residue evidence is generated using appropriate modifications to Eq. 2.3. In all cases, however, the evidence is added to the element of R indexed by the discretized mass of the corresponding charge 1+ b ion, so that all evidence related to a particular locus of fragmentation is aggregated together.

Once all possible residue evidence has been accumulated into matrix R, the values in R undergo a linear discretization to integer values, such that the minimum value in R is 0 and the maximum is some specified integer r_{max}. This integer discretization ensures that scores will have integer values, which is required for the subsequent dynamic programming.

The theoretical spectrum corresponding to a candidate peptide is very simple. For each

possible prefix sequence of the peptide a tuple is created, consisting of two elements: the identity of the prefix's C-terminal amino acid and an integer formed by discretizing the prefix mass with bin width $W = 1.0005079$. For a candidate peptide P of length n, the full representation B then consists of $n-1$ such tuples: $\{(a_k, m_k)\}_{k=1\ldots n-1}$.

Finally, assume that candidate peptide P of length n has a minimal binary representation B as described above. Then the residue evidence score Ψ between P and spectrum S is the sum of elements selected from the residue evidence matrix R (derived from S) according to the tuples in B:

$$\Psi(P,S) = \sum_{k=1\rightarrow n-1} R_{a_k m_k} \tag{2.4}$$

2.2.3 Calibrating the residue evidence score via dynamic programming

The following assumes a spectrum S with precursor mass is m_S is being scored.

Let $P^{(1\rightarrow n)}$ be a peptide of length n, with mass $m^{(1\rightarrow n)} = m_S$ and amino acid sequence $a_1, a_2, \ldots, a_n$. Because $\Psi(P,S)$ is additive, the score for matching S with $P^{(1\rightarrow n)}$ can be obtained by first calculating the score for the prefix sequence $P^{(1\rightarrow n-1)} = a_1, a_2, ..., a_{n-1}$, then adding the evidence $r = R_{a_n m_S}$ from the residue evidence matrix R. Note that this process is equally valid for any subsequence $P^{(1\rightarrow k)} = a_1, a_2, ..., a_k$ with mass $m^{(1\rightarrow k)}$,

$$\Psi(P^{(1\rightarrow k)}) = \Psi(P^{(1\rightarrow k-1)}) + R_{a_k m^{(1\rightarrow k)}} \tag{2.5}$$

Let $C_{s,m}$ be the count of peptides with mass m that produce a discretized score s. If (hypothetically) all the peptides have the same terminal amino acid a with mass m_a, then we would have

$$C_{s,m} = C_{s-R_{am}, m-m_a} \tag{2.6}$$

Allowing for all naturally occurring amino acids $a_i \in A$, with masses m_{a_i}, the count becomes

$$C_{s,m} = \sum_{a_i \in A} C_{s-R_{a_i m}, m-m_{a_i}} \tag{2.7}$$

Since $\Psi(P,S)$ is additive, Eq. 2.7 is valid for all masses $1 \leq m \leq m_S$. Eq. 2.7 defines the basic recursion of the dynamic programming.

The dynamic programming computation of C is conducted in a two-dimensional array, where the rows are indexed by s and the columns by m (Figure 2.2B). The number of rows is determined by an estimate of the largest possible score for S:

$$s_{max} = \sum_{i=n-q+1}^{n} Z_{(i)} \tag{2.8}$$

where $Z_{(i)}$ refers to the sorted column maxima from R and $q = \lceil m_S / \min\{m_{a_i \in A}\} \rceil$.

We initially set:

- $C_{0,T_N} \leftarrow 1$, where T_N is the mass of the N-terminal group.
- $C_{s,m} \leftarrow 0$ for all $s \neq 0$ or $m \neq 1$. This includes a range of indices $s < 1$ and $m < 1$ that are accessed during the dynamic programming.

The elements of the array are then computed sequentially:

for $m = T_N$ to $m_S - T_C$ **do**
  for $s = s_0$ to s_{max} **do**
    $C_{s,m} = \sum_{a_i \in A} C_{s - R_{a_i} m, m - m_{a_i}}$
  end for
end for

Above, the values T_N and T_C represent the masses of the N-terminal and C-terminal groups, respectively. Hence, the last column of the matrix C typically represents mass $m_S - 17$, since the C-terminal group is usually a hydroxyl. This column holds the desired distribution of Ψ_R over all possible peptides consistent with m_S.

By using Eq. 2.7 in the dynamic programming, we make the assumption that all peptides are *a priori* equally likely. This is not biologically plausible, and, in fact, leads to distributions of $\Psi(P, S)$ that lack appropriate statistical properties. This problem can be solved by considering the relative abundances of amino acids in the recursive counting:

$$C_{s,m} = \sum_{a_i \in A} C_{s - R_{a_i} m, m - m_{a_i}} \cdot p_{a_i} \tag{2.9}$$

where p_{a_i} is the probability of finding amino acid a_i in a large collection of naturally occurring peptides, with $\sum_{a_i \in A} p_{a_i} = 1$. Note that it may be important to use different estimates of p_{a_i} for the N-terminus, C-terminus, and non-terminal positions, depending on the specificity of the enzyme used for digestion.

Assume we have calculated, using dynamic programming, the distribution of scores C_{s,m_S} over all possible peptides for spectrum S, where $0 \leq s \leq s_{max}$. Then the p-value relative to this distribution for a specific peptide P, matched to S with residue evidence score $\psi = \Psi(P, S)$, is

$$p(\psi, C_{s,m_S}) = \frac{\sum_{s \geq \psi} C_{s,m_S}}{\sum_{0 \leq s \leq s_{max}} C_{s,m_S}}. \tag{2.10}$$

These p-values can be used in place of raw residue evidence scores during a standard database search.

2.2.4 Combining correlated p-values

The res-ev p-value and the XCorr p-value provide complementary yet not fully independent estimates of the quality of a given peptide-spectrum match (PSM). Accordingly, we employ a previously described method for assigning a p-value to the product of n correlated p-values,[6] using the following equation:

$$Pr(Z_n \leq p) \approx p^y \sum_{i=0}^{\lfloor m \rfloor - 1} \frac{(-\ln p^y)^i}{i!} + p^y (m - \lfloor m \rfloor) \frac{(-\ln p^y)^{\lfloor m \rfloor}}{\lfloor m \rfloor !} \tag{2.11}$$

where n is the number of p-values being multiplied (in our case, $n = 2$), Z_n is the product of the p-values, m is a parameter that can range from 1 to n, and $y = m/n$. The value of m indicates the degree of correlation among the n p-values, where total correlation (i.e., identical p-values) corresponds to $m = 1$, and total independence corresponds to $m = n$. In this setting, we used decoy p-values to empirically estimate $m = 1.2$ (Figure 2.3) by minimizing the previously described error function:

$$E(m) = \sqrt{\sum_{i=1}^{n} [\log(p_i(m)) - \log(i/(n+1))]^2} \tag{2.12}$$

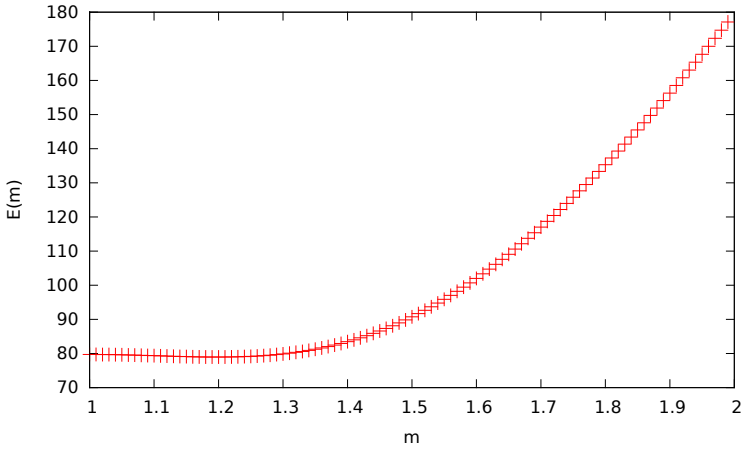


Figure 2.3: **Using decoys to set the m parameter.** The figure plots the error $E(m)$ (Equation 2.12) as a function of the parameter m. To generate this plot, the *Plasmodium* data set was analyzed using both XCorr p-value and res-ev p-value. All decoy p-values for each spectrum were retained, yielding a set of 197,029 pairs of p-values. Values of $E(m)$ were computed for $m = 1.00, 1.01, \ldots, 1.99$. The minimum error value occurs around $m = 1.20$.

Species	instrument	spectra	precursor (ppm)	proteins	peptides
Plasmodium	LTQ Velos-Orbitrap	12,748	50	11,737	746,911
E. coli	Q-Exactive	53,083	50	3,895	2,094,174
Human	LTQ Orbitrap Elite	5,796	10	110,829	2,062,622
Ocean	Q-Exactive HF	98,137	10	N/A	35,546,224

Table 2.1: **Mass spectometry datasets.** The database used for the ocean data set is comprised of individual peptides derived from high-throughput sequencing reads, rather than full-length proteins.

where $p_i(m)$ is the ith largest p-value (of the product of p-values) in a set of n decoy p-values. Intuitively, this error function attempts to minimize the difference between the observed p-value distribution and an ideal, uniform distribution.

2.2.5 Data sets

We used four previously described tandem mass spectrometry data sets to validate our methods (Table 2.1). The data sets were selected to represent a diversity of both sample types and instrument types. The mass spectrometry data used in this study have been deposited to the PRIDE Archive (`http://www.ebi.ac.uk/pride/archive/`) via the PRIDE partner repository with the data set identifier PXD009265.

Plasmodium falciparum fraction:[74] *P. falciparum* 3D7 was grown at 37° Celsius in RPMI-1640 culture medium. Following synchronization, infected cells were lysed using saponin. An 8 M urea lysis buffer was used to create parasite extracts, which were then reduced and alkylated. Proteins were digested using Lys-C, and the resulting peptides were labeled with TMT. Following TMT labeling, strong cation exchange chromatography (SCX) was use to fractionate the sample into 20 fractions. Fractions were analyzed on a LTQ Velos-Orbitrap mass spectrometer (Thermo Scientific). All MS1 and MS2 scans were acquired at high resolution. The data from fraction 13 was used in this study and contained 12,748 scans.

The protein database used in the database search was downloaded from NCBI in October 2013 (*Plasmodium falciparum 3D7*).

Ocean metaproteome:[60] Water samples from the northern Chukchi Sea bottom waters were collected in the summer of 2013. To remove larger eukaryotes, each 15 L water sample was prefiltered through a 10 μm and then a 1 μm filter. The remaining liquid was collected onto a glass fiber filter and frozen. Cells were lysed using bead beating in 6 M urea. A total of 100 μg of total protein were used for digestion. Prior to digestion, 300 ηg of human ApoA1 protein was added and then the sample reduced and alkylated. Proteins were digested with trpysin and then desalted. Peptide separation was conducted using a NanoAquity HPLC with a 4 cm precolumn and a 30 cm analytical column. Peptides were eluted at a rate of 300 ηL/min for 2 hours using a nonlinear gradient. Data was collected on a Q-Exactive HF (Thermo Scientific). The mass spectrometer was operated in a Top 20 data-dependent acquisition mode with a 5 second dynamic exclusion window. Ions only between 400-1600 m/z were collected. This resulted in a dataset with 98,137 scans. The database used in the database search consists of a metapeptide database that was derived from shotgun metagenomic sequencing of the same ocean sample (`https://noble.gs.washington.edu/proj/metapeptide/metapeptides_CS.fasta`). Briefly, a metapeptide database is a peptide database whose sequences are derived from raw read sequences that have been translated into peptides in all six reading frames.[60]

Human fraction:[43] Histologically normal adrenal gland tissue from three deceased individuals were pooled together using equal amonts of protein from each donor. Samples were lysed using SDS. The protein sample was fractionated by SDS-polyacrylamide gel electrophoresis (SDS-PAGE). Then the protein bands were destained, reduced, and alkylated. Protein digestion was peformed using an in-gel trypsin digestion. Following digestion, the peptides were desalted. The resulting peptide sample was separated with a 60 min linear gradient using reversed-phase liquid chromatography on an Easy-nLC II nanoflow liquid chromatography system (Thermo Scientific). Data was acquired using a

LTQ-Orbitrap Elite (Thermo Scientific). The mass spectrometer was operated in a Top 20 data-dependent acquisition mode with a 30 second dynamic exclusion window. MS1 scans were acquired at a mass resolution of 120,000 at 400 m/z. MS2 scans were acquired at a mass resolution of 30,000 at 400 m/z. This study used data from the first fraction, which contained 5,796 scans. Each chromosome's "protein.faa" was downloaded from RefSeq (`ftp://ftp.ncbi.nlm.nih.gov/refseq/H_sapiens/mRNA_Prot`) in September 2016 and concatenated to form the human protein database.

E. coli fraction:[26] A yfgM knockout strain and WT strain of *E. coli* MC4100 was cultured at 37° Celsius in LB broth (Difco, Sparks, MD). Cells were harvested using centrifugation once the OD_{600nm} reached ∼.08. Cellular pellets were suspended in a lysis buffer and then lysed by rapidly passaging the cells through a hypodermic syringe needle and by sonication. Proteins were then reduced, alkylated, and then reduced a second time. A 4 hour digestion step with Lys-C was followed by an overnight trypsin digestion. The resulting peptides were chemically labeled using stable isotope dimethyl labeling. The yfgM knockout lysate was labeled with the "Medium" isotope and the the WT sample was labeled with the "Heavy" isotope. These lysates were mixed together in a 1:1 ratio and then fractionated into 45 fractions using strong cation exchange. Samples were analyzed on a Q-Exactive (Thermo Scientific) coupled to a Easy UHPLC (Thermo Scientific) system. Peptides were eluted during a 3 hour gradient with a flow rate of 100 ηL/min. The Q-Exactive instrument was operated in a Top 20 data-dependent acquisition mode. MS1 scans were acquired at a mass resolution of 35,000. MS2 scans were acquired at a mass resolution of 17,500. The 21st fraction was used for this study and contained 53,083 scans. The protein database used in the database search was downloaded from Uniprot in December 2017 (*Escherichia coli* str. K-12 substr. MC4100).

2.2.6 Target-decoy evaluation

For this work, we used the following publicly available database search engines.

- Crux version 3.1 (`http://crux.ms`; linux version) was used to generate combined p-value, res-ev p-value, XCorr p-value, and high-resolution XCorr[31,62,73]
- MS-GF+ version v2016.12.12 (`https://omics.pnl.gov/software/ms-gf`)[45]
- MS Amanda version 1.0.0.7504 (`http://ms.imp.ac.at/?goto=msamanda`; linux version)[14]
- Morpheus version 272 (`http://cwenger.github.io/Morpheus/`; linux version)[98]

We took great care to ensure a fair comparison of results across all database search engines. One of the more important ways we accomplished this goal was to try to guarantee that all the database search engines considered a common set of target and decoy peptides. To this end, we took several non-standard steps in our analysis. First, we predigested our protein fasta files *in silico* using the tide-index tool in Crux. This predigestion did not include suppression of cleavage by proline, because not all search engines use this rule. Decoy peptides were generated by tide-index by shuffling the amino acid sequence of each peptide, leaving the N-terminal and C-terminal amino acids in place. For this digestion, no missed cleavages were allowed, and N-terminal peptides with a leading methionine were included in two copies, with and without the methionine. Peptides shorter than six amino acids and peptides with one or more non-enzymatic termini were not considered. The resulting target and decoy peptides were placed into a new .fasta file. The words 'target' and 'decoy' were appended to the peptide headers of the peptides in their respective .fasta files. Then the target and decoy .fasta file were concatenated to create a target-decoy database for MS Amanda.

Because it is not possible to turn off N-terminal methionine clipping in MS Amanda, in order to ensure that the other three database search tools were exposed to the same peptides as MS Amanda, we then subjected the predigested .fasta files to a second round of "digestion". In this second round, no missed cleavages were allowed, N-terminal methionines were allowed to be clipped, and peptides shorter than five amino acids were removed. Since

the .fasta files that went through the second round were pre-digested, this next step only performed clipping of N-terminal methionines. The resulting target and decoy peptide files were then concatenated to create a target-decoy database for MS-GF+, Morpheus, and Crux. This predigestion strategy ensured that all search engines considered the same set of candidates. Note that, subsequent to the searches, we also checked that each detected peptide was indeed present in the .fasta database.

In addition to ensuring the database search engines considered the same set of targets and decoys, we tried to match the experimental parameters in each database search as exactly as possible. We removed any MS2 scans that had fewer than 10 peaks in it. All searches were run with full digestion (*i.e.* no missed cleavages). No non-enzymatic termini and no isotope errors were allowed. The maximum precursor charge was set to 25 (*E. coli*), seven (human and ocean), or nine (*Plasmodium*). The *E. coli*, human, and ocean sample were run with trypsin as the digestion enzyme, while the *Plasmodium* sample was run with Lys-C as the digestion enzyme. The proline rule was ignored for all runs. The precursor mass tolerance was set to 50 ppm for the *E. coli* and *Plasmodium* runs and 10 ppm for the human and ocean runs. We set the fragment mass tolerance at 0.02 Da for combined p-value, res-ev p-value, MS Amanda, and Morpheus for all four datasets; however, we were unable to set the fragment mass tolerance for MS-GF+ as it is not a user-level parameter. For MSGF+, we can somewhat control the fragment mass tolerance by correctly setting the user-level parameters of 'inst' and 'm'. For the human and *Plasmodium* dataset, we set 'inst' to 1 and 'm' to 3. These settings correspond to high-resolution MS2 scans that were generated by HCD. For the ocean and *E. coli* sample, we set 'inst' to 3 and 'm' to 3. These settings correspond to high-resolution MS2 scans that were generated by a Q-Exactive. For all four datasets, we allowed a fixed carbamidomethyl modification to cysteine and a variable methionine oxidation modification. In addition we allowed a variable light, intermediate, and heavy dimethyl label (28.0313, 32.0564 and 36.0757 Da, respectively) on lysines and the N-terminus for the *E. coli* run. For the *Plasmodium* run, we included a fixed TMT modification (229.16293 Da) on lysines and the N-terminus. Methionine clipping was turned

off for Crux, Morpheus, and MS-GF+ since the input .fasta file already contained clipped peptides. All search engines runs were done in target-only mode since the peptide headers in the concatenated target-decoy .fasta file already denoted whether it was a target or decoy peptide.

A custom R script was used to combine the results of the various search engines together into a single table. Each row represents the PSM that each database search detected for a particular scan. For each row (scan) the combined p-value, res-ev p-value, XCorr p-value, SpecEvalue (MS-GF+), weighted probability (MS Amanda), and Morpheus scores are listed. In addition, the peptide that each score function detected, and whether that peptide is a target or decoy, is also listed. The value 'NA' is placed into empty cells that result from one score function scoring a scan and another score function not scoring that particular scan. This phenomenon is due to each program having a different threshold for the minimum number of peaks required to score a scan. A second R script used the PSM table as input to calculate false discovery rates and generated the plots for this publication. We used the following false discovery rate equation: FDR = (number of decoys + 1) / number of targets.[50]

2.2.7 Percolator analysis

For the Percolator analysis, the Tide search was performed as described previously, except that during index creation, the "digestion" option was set to "partial-digest" and one missed cleavage was allowed ("missed-cleavages=1"). This setting allows Percolator to more effectively re-rank various types of PSMs, while taking into account their digestion conditions. We then applied the Crux implementation of Percolator directly to the Tide search results. The resulting feature vector contains, in addition to the standard Percolator features, three separate scores for each PSM: the negative logarithms of the combined p-value, res-ev p-value, and XCorr p-value. All default Percolator parameters were used except that "only-psms" was set to true. Note that PSM-level FDR is estimated by Percolator using target-decoy competition (including the +1 correction to the number of decoys).

2.3 Results

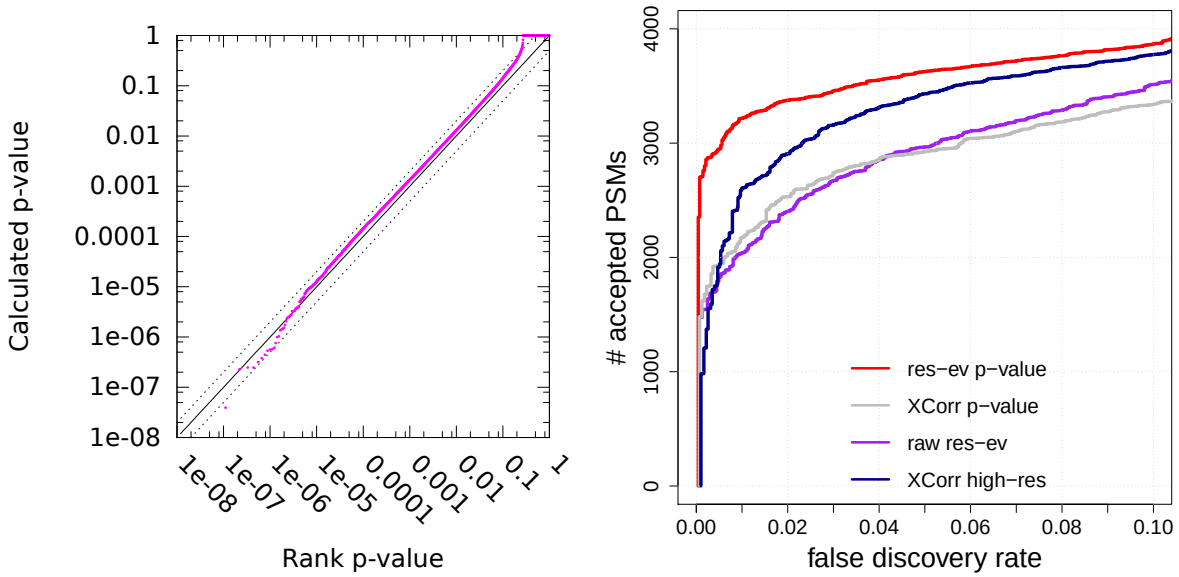
2.3.1 Statistical validation of residue-evidence p-value

For any observed spectrum, we can use dynamic programming to determine the exact distribution of residue-evidence scores that result from each possible peptide sequences whose discretized mass matches the discretized precursor mass. We can then compare the score from a particular PSM to this distribution and calculate a p-value, i.e., we compute the probability of observing a residue-evidence score greater than or equal to the score of a particular PSM.

To test the validity of the resulting p-values, we searched real data against a decoy database. Specifically, we searched the *Plasmodium* data set against a shuffled *Plasmodium* database (see Methods for details) using a wide precursor mass tolerance of 3 Da. Because the decoy peptides have been shuffled, we expect all of the resulting PSMs to be incorrect. Hence, in this setting, our p-values should be uniformly distributed; i.e., the probability of observing a p-value less than or equal to, say, 0.05 should be 5%. A quantile-quantile plot (Figure 2.4A) of the calculated p-values against the rank of the p-values confirms that the residue-evidence p-values are generally uniform. However, we noticed a trend away from $y = x$ among large p-values (horizontal line in the upper right hand corner of Figure 2.4A), as well as an overall upward shift of the p-value distribution shown in the figure. These two phenomena arise because of the discrete nature of the res-ev score. In practice, many PSMs result in a residue-evidence score of 0, leading to an inflation of p-values of 1.0 and a consequent decrease in the remaining p-values. Overall, the near uniformity of the empirical res-ev p-values indicates that they provide an accurate assessment of the statistical confidence associated with a given PSM.

2.3.2 Residue-evidence works well for high-resolution data

Having established the validity of the res-ev p-value, we next sought to measure the statistical power of the score function in the context of a real database search. For this test, we again



A B

Figure 2.4: **Calibration of the residue evidence score via dynamic programming.** (A) The figure plots the p-value, as calculated via dynamic programming, versus the rank p-value, for decoy PSMs from a *Plasmodium* dataset. The lines $y = x$ (solid line), $y = 2x$ (dotted line) and $y = 0.5x$ (dotted line) are included for reference. (B) The figure plots, for the *Plasmodium* dataset, the number of PSMs accepted as a function of q-value threshold, for four different database search methods: XCorr calibrated via dynamic programming using low-resolution m/z bins, uncalibrated XCorr using high-resolution m/z bins, uncalibrated res-ev, and res-ev calibrated via dynamic programming. Note that a series corresponding to calibrated XCorr using high-resolution m/z bins is not included, because dynamic programming cannot be carried out in conjunction with small m/z bins, as explained in the Introduction.

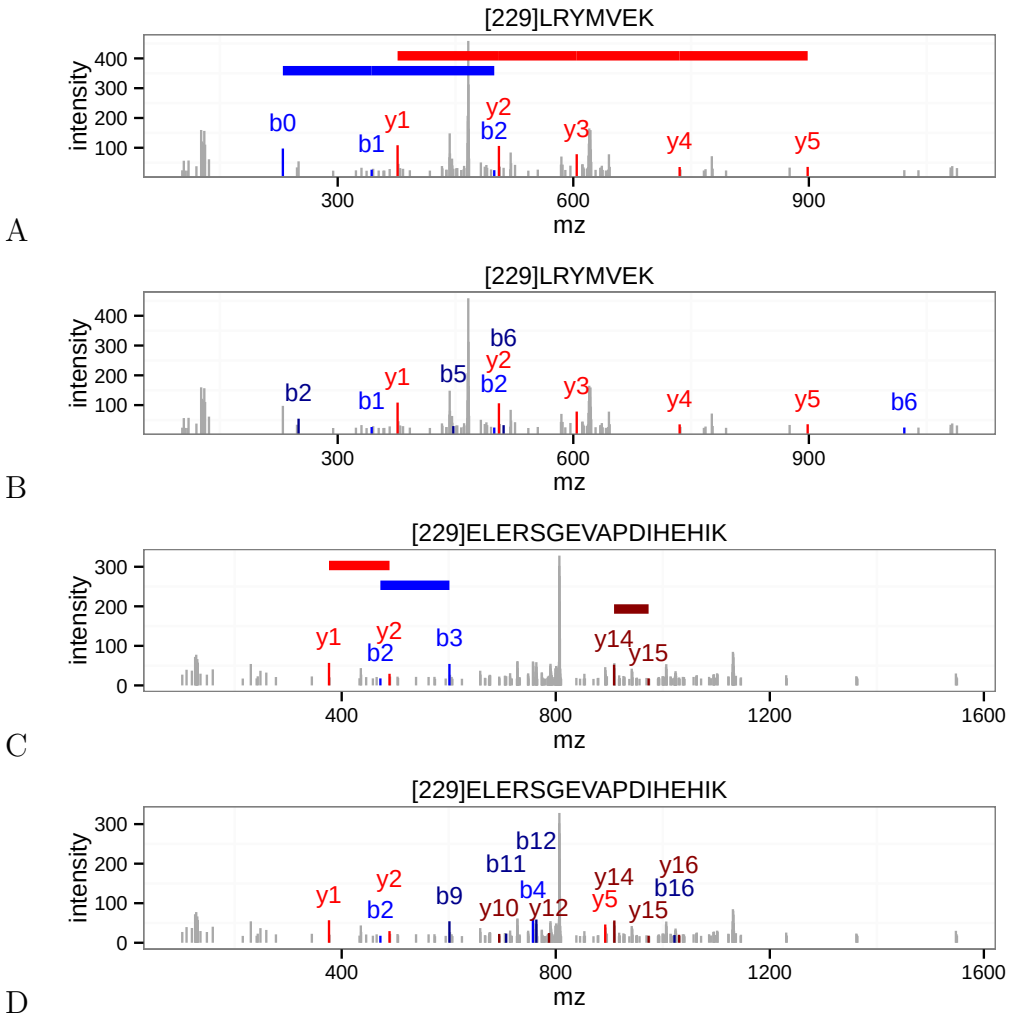


Figure 2.5: **Disagreements between the XCorr score and the residue-evidence score.** (A) An annotated *Plasmodium* spectrum (scan 5468) that received a low (i.e., good) p-value from the residue-evidence score. Colored horizontal lines indicate the locations of peak-pairs that contribute to the residue-evidence score. Note that the mass of [229] corresponds to the tandem mass tag modification. (B) Same as (A), but annotated using XCorr. This scan received a high XCorr p-value. (C) *Plasmodium* scan 11156, annotated with res-ev, with a high p-value. (D) Same as (C), but annotated using XCorr, with a low p-value. In each panel, peaks colored in blue, dark blue, red, and dark red represent b+1, b+2, y+1, y+2 ions, respectively.

used the *Plasmodium* data set, but we searched against a concatenated database of both real ("target") and shuffled ("decoy") peptide sequences. From the resulting ranked list of PSMs, we used target-decoy analysis to estimate the false discovery rate (FDR) associated with each observed PSM score.[19] We measured FDR out to a maximum of 10%, reasoning that higher FDR thresholds are not likely to be of much practical value. For comparison, we repeated our search with three other score functions: the uncalibrated res-ev score function, the uncalibrated high-res XCorr score function, and the XCorr p-value. Note that the latter necessarily discards the high resolution of the fragment m/z axis, because the XCorr dynamic programming procedure requires ∼1 Da bins.

The results (Figure 2.4B) clearly show that, for this particular data set, the res-ev p-value score function outperforms the three competing methods. Focusing on the commonly used FDR threshold of 0.01, we see that the res-ev p-value detected 3,217 PSMs. This corresponds to an increase of 1178 (57.78%) PSMs relative to the raw residue-evidence score, 1,047 (48.25%) PSMs relative to the XCorr p-value and 609 (23.35%) PSMs relative to high-resolution XCorr. Thus, this experiment suggests that taking simultaneous advantage of statistical calibration and high-resolution data improves preformance.

A complementary measure of the quality of a PSM score function is the "target match percentage" (TMP), which is defined as the fraction of spectra for which the top-scoring match involves a target peptide.[3] For a perfectly random score function, we expect the TMP to be ∼50%. The best possible TMP is 100%; however, in practice any real data set will contain spectra that cannot be identified, either because the corresponding generating peptide is not in the given peptide database or because the spectrum was generated by a non-peptide contaminant. TMP provides a measure of the quality of a score function that is independent of a score function's calibration. This is because the TMP never involves comparing scores for PSMs involving different spectra. Hence, the distribution of PSM scores for spectrum A can be dramatically different from the distribution of PSM scores for spectrum B, but the TMP achieved by the score function can still be high.

In the *Plasmodium* TMP analysis, the res-ev p-value (and, by definition, also the raw res-

ev score) achieved the best TMP of 65.08% (8,172 PSMs). High-resolution XCorr yielded the second best TMP of 64.20% (8,061 PSMs). Not surprisingly, XCorr p-value, which discards high-resolution m/z information, had the worst TMP (63.94%, or 8,029 PSMs).

In order to better understand the differences in scoring between XCorr and residue-evidence, we looked at several spectra where XCorr p-value and res-ev p-value greatly disagreed on the significance of their best PSMs. Figure 2.5 shows two such spectra (scans 5468 and 11156) from the *Plasmodium* dataset that have been annotated by both XCorr and residue-evidence.

Scan 5468 (Figure 2.5A–B) corresponds to a case where the PSM is given a small residue-evidence p-value and a large XCorr p-value. Specifically, although this scan was assigned the same peptide (LRYMVEK) by both score functions, the resulting PSM received a p-value of 5.13×10^{-4} (0.28% FDR) from residue-evidence and a p-value of 7.20×10^{-1} (48.50% FDR) from XCorr. The source of this difference is not immediately obvious, because the numbers of peaks annotated by the two score functions are similar. The only additional ions that XCorr identifies over residue-evidence are the doubly charged b2, b5, and b6 ions and the singly charged b6 ion. This PSM scores well according to the res-ev score function because the spectrum contains two long "ladders" of consecutive peaks (y1 to y5, and b0 to b2). These ladders are particularly unlikely according to a null model in which each peak is treated independently. Conversely, the poor score from XCorr may arise because most of the annotated peaks have low intensities. Note that the res-ev score did not annotate the doubly charged b5 and b6 ions because the mass difference between these two peaks was too different from the mass of glutamate. This speaks to the power of using high resolution on the MS2 m/z axis.

In contrast, scan 11156 (Figure 2.5C–D) illustrates why some PSMs score well using XCorr and poorly using residue-evidence. This scan was assigned the same peptide (ELERSGEVAPDIHEHIK) by both score functions, but the resulting PSM received a high p-value (8.45×10^{-1}, 47.4% FDR) from residue-evidence and a low p-value (1.11×10^{-2}, 3.4% FDR) from XCorr. The disparity between the two score functions arises because, using

residue-evidence, only three pairs of peaks contribute to the score. In contrast, the XCorr score includes individual components corresponding to sixteen different fragment ions (b2, b3, b4, b7, b9, b11, b12, b16, y1, y2, y5, y10, y12, y14, y15, and y16). In Figure 2.5D, there are only fourteen colored fragment ions because the b3 and b9 ion correspond to the same peak and the b7 and the y16 ion also correspond to the same peak. The lack of a long "ladder" of successive b- or y-ion peaks keeps the residue-evidence score low, relative to XCorr.

2.3.3 Combining the two scores yields equal or improved power

The two scans in Figure 2.5 clearly suggest the need for a score function that can combine the res-ev p-value and the XCorr p-value, thereby potentially correctly identifying both scans 5468 and 11156. Estimating the p-value for the product of a pair of independent p-values is relatively straightforward; however, in our case, the res-ev and XCorr p-values are clearly not independent since they are derived from the same PSM. To verify this lack of independence, we computed the res-ev p-value and XCorr p-value for four different data sets. These data sets were selected for diversity: they represent different proteome sizes, digestion enzymes (trypsin and Lys-C), instruments (LTQ Orbitrap Velos, Q-Exactive, and LTQ-Orbitrap Elite), and instrument resolutions. The strong $y = x$ component in each of the resulting density plots (Figure 2.6, top row) indicates a strong lack of independence. We also note that, in general, res-ev shows an enrichment of very small p-values compared to XCorr.

To combine these two scores, we applied a previously described method for estimating the statistical significance of the product of correlated p-values[5] (see Methods for details). We hypothesized that the resulting combined p-value would perform better than res-ev or XCorr because these two score functions take advantage of different types of evidence in the spectrum: the res-ev p-value focuses on adjacent pairs of peaks, whereas the XCorr p-value focuses on single peaks.

For both of the scans in Figure 5, the combined approach assigns a p-value that is

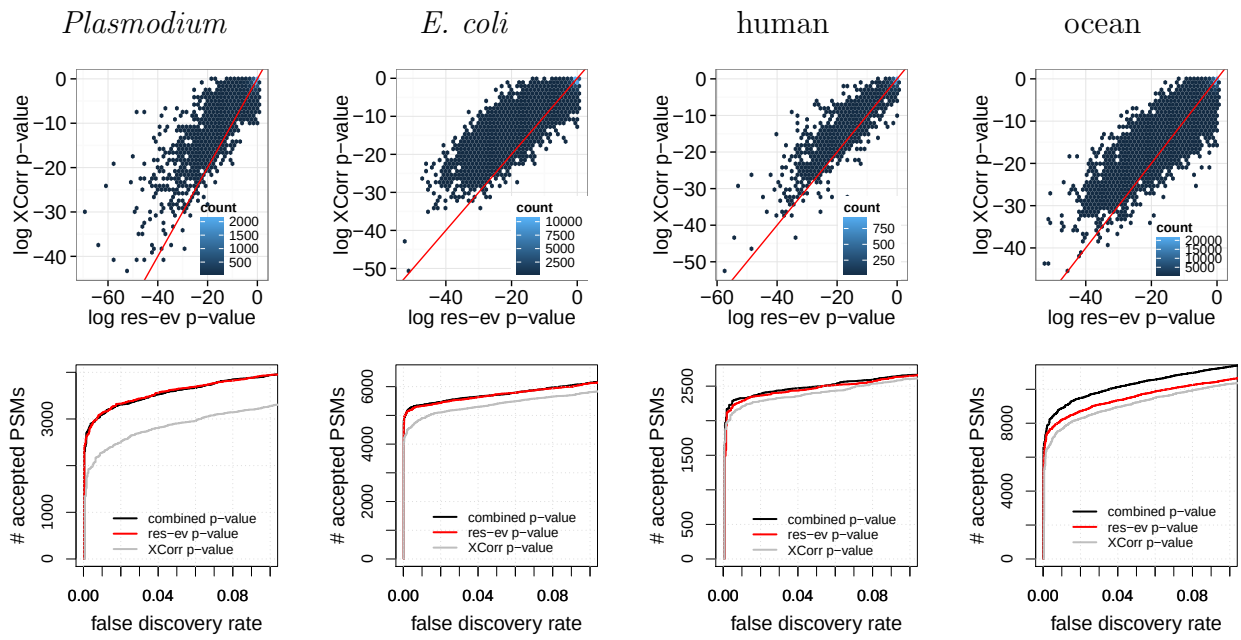


Figure 2.6: **Combining XCorr and res-ev.** (Top row) Each panel plots, for a specified dataset, a density plot of p-values from XCorr (y-axis) versus res-ev (x-axis). The points are binned into hexagons, and the color of each hexagon represents the number of points within each bin. The red line represents $y = x$. (Bottom row) Each panel plots the number of PSMs accepted as a function of FDR threshold, for three different database search methods: XCorr p-value, res-ev p-value, and combined p-value.

intermediate between the p-values assigned by XCorr and res-ev. In particular, for scan 5468 the p-values are 7.20×10^{-1} (XCorr, 48.5% FDR), 1.29×10^{-2} (combined, 1.82% FDR), and 5.13×10^{-4} (res-ev, 0.29% FDR); and for scan 11156, the p-values are 1.11×10^{-2} (XCorr, 3.39% FDR), 9.19×10^{-2} (combined, 7.2% FDR) and res-ev p-value (8.45×10^{-1}; 47.38% FDR). Hence, the combined p-value allows us to detect both of these peptides, whereas XCorr p-value and res-ev p-value would each individually miss one PSM at a 10% FDR threshold.

To test the combined p-value more systematically, we compared the performance of res-ev p-value, XCorr p-value and the combined p-value on four data sets. The results (Figure 2.6, bottom row) show that the combined p-value is generally the best-performing method. Notably, however, in three out of the four cases, the performance of combined p-value is comparable to res-ev p-value. In two out of the three cases, combined p-value identified only 35 (0.67%, *E. coli*) and 51 (2.24%, human) more PSMs than res-ev p-value at a 1% FDR threshold. In the third case, combined p-value did marginally worse than res-ev p-value: at a 1% FDR, combined p-value detected 29 (0.94%) fewer PSMs than res-ev p-value for the *Plasmodium* dataset. Only in the ocean dataset did the combined p-value yield a large increase in performance (817 PSMs, or 11.01%). Conversely, XCorr p-value tends to perform poorly in all cases. Overall, these results suggest that combining res-ev with XCorr does not lead to decreased performance and occasionally yields a performance increase relative to using the res-ev p-value alone.

2.3.4 Comparison with existing methods

Finally, we compared the combined p-value with three existing methods that take advantage of high-resolution tandem mass spectra: MS Amanda, Morpheus, and MS-GF+. MS Amanda and Morpheus are designed to take advantage of high-resolution tandem mass spectra but are not statistically calibrated. MS-GF+, like res-ev p-value and combined p-value, takes simultaneous advantage of statistical calibration and high-resolution MS2.

We found that, in general, combined p-value and MS-GF+ outperformed MS Amanda

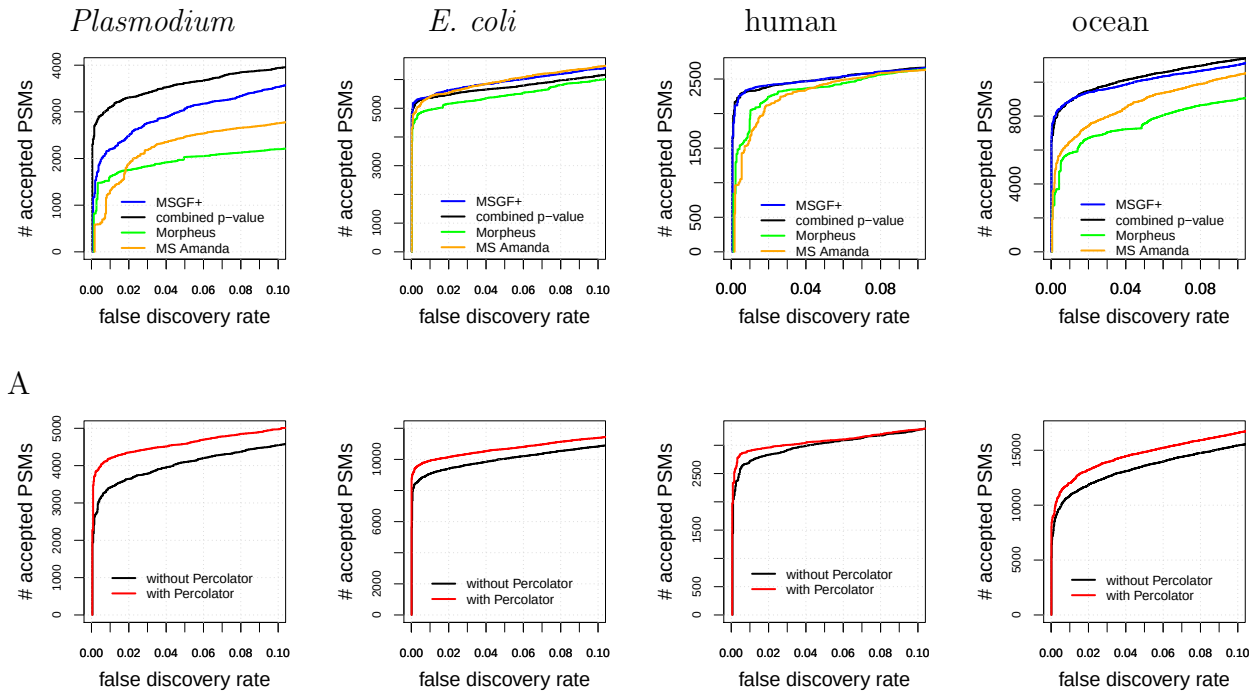


Figure 2.7: **Comparison with existing methods.** (A) The panel plots, for four datasets, the number of PSMs accepted as a function of FDR threshold for four different database search methods: MS-GF+, combined p-value, MS-Amanda, and Morpheus. (B) Similar to (A), but the two series correspond to the combined p-value with and without post-processing via Percolator.

and Morpheus over the entire 0–10% FDR range (Figure 2.7A). For example, combined p-value detected 1781 (135.33%), 711 (44.02%), and 2414 (37.33%) more PSMs than MS Amanda at a 1% FDR for the *Plasmodium*, human, and ocean samples, respectively. Similar respective improvements of 1479 (91.41%), 404 (21.02%), and 3074 (52.95%) were observed for the combined p-value relative to Morpheus, as well as for MS-GF+ relative to MS Amanda and Morpheus. The one exception was the *E. coli* dataset, where MS Amanda performed comparably to MS-GF+ and slightly better than the combined p-value. Nonetheless, we interpret the performance improvement that the combined p-value and MS-GF+ offer over MS Amanda and Morpheus as evidence of the value of having a statistically calibrated score.

When we focus on the comparison between the combined p-value and MS-GF+, no clear winner emerges. For three datasets, *E. coli*, human, and ocean, MS-GF+ marginally outperformed combined p-value at a 1% FDR threshold. Qualitatively, we observe on the *E. coli* dataset that the difference between MS-GF+ and the combined p-value is relatively small for low FDR thresholds—e.g., at a 1% FDR, MS-GF+ yields only 42 (0.78%) more PSMs than the combined p-value—but increases for larger FDR thresholds. Conversely, for the human dataset the difference between MS-GF+ and the combined p-value is largest (24 PSMs or 1.03%) at around 1% FDR, but then this difference all but disappears for large FDR thresholds. In contrast to the prior two datasets, for the ocean dataset MS-GF+ identfied 8 (0.09%) more PSMs relative to combined p-value at a 1% FDR threshold. However, at larger FDR thresholds, combined p-value consistently performed better then MS-GF+. Finally, for the *Plasmodium* dataset, we observed that combined p-value performed dramatically better than MS-GF+ across the entire FDR range, with an improvement of 899 PSMs (40.90%) at a 1% FDR.

Comparing the target match percentages of combined p-value, MS-GF+, Morpheus, and MS Amanda yielded unexpected results. In contrast to the comparison shown in Figure 2.7, where Morpheus consistently performed worse than the other score functions, in the TMP comparison Morpheus performed better than the other score functions. Morpheus's TMP was higher than the second-best TMP by 1.83%, 0.63%, 1.43%, and 1.38% for the *Plasmodium*,

	combined p-value	MS-GF+	Morpheus	MS Amanda
Plasmodium	65.03%	63.96%	66.86%	60.80%
human	72.57%	71.23%	73.20%	72.19%
E. coli	56.47%	56.72%	58.64%	57.21%
ocean	55.71%	55.46%	57.44%	56.06%

Table 2.2: **Target match percentages.** The TMPs of four score functions (rows) for four datasets (columns). The TMP is defined as the percentage of spectra that match a target peptide.

human, *E. coli*, and ocean datasets, respectively. Among the remaining three methods, the TMP values for most of the data sets were remarkably similar to each other, spanning ranges of 71.23–72.57% (human), 56.47–57.21% (*E. coli*), and 55.46–56.06% (ocean), respectively. The TMP values were slightly more variable for the *Plasmodium* dataset, but even in this case, Morpheus was a clear winner. These results suggest that Morpheus is doing a very good job of identifying the correct candidate peptide for each spectrum, and perhaps suffers in the calibration of its scores from one spectrum to the next.

Note that the fact that the combined p-value and MS-GF+ yield different results on these datasets suggest that the two scores may be complementary. Indeed, a scatter plot of the p-values produced by the two methods (Figure 2.8) suggests considerable disagreement between the two. Accordingly, a method that combined all three score functions (XCorr, res-ev and MS-GF+) into a single p-value or combined the score functions in a post-processing phase[16] might yield even better results.

2.3.5 Using Percolator in conjunction with combined p-value improves power

In practice, in most proteomics experiments a post-processor such as Percolator[34] or PeptideProphet[39] is used to reanalyze the database search results to improve performance. Therefore, we tested whether the performance of combined p-value can be improved by post-processing

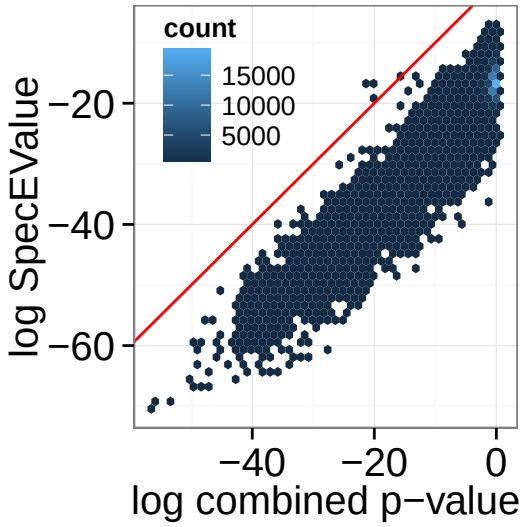


Figure 2.8: **Comparison of MSGF+ versus the combined p-value.** A scatter plot of the MS-GF+ SpecEValue against the combined p-value for the ocean dataset.

via Percolator. We ran the combined p-value database search for all four datasets, as previously described, except that we allowed the peptide database to contain semi-tryptic peptides and peptides with one missed cleavage. This approach provides Percolator the opportunity to re-rank PSMs while taking into account their digestion conditions. Following the database search, we reanalyzed the database results with Percolator. We found that combined p-value with Percolator performed better than combined p-value by itself for all four datasets over the entire 0–10% FDR range (Figure 2.7B). At a 1% FDR, Percolator identifies an additional 778 (22.8%), 844 (9.30%), 182 (6.69%), and 1180 (10.89%) PSMs for the *Plasmodium*, *E. coli*, human, and ocean datasets, respectively.

2.4 Discussion

The residue-evidence p-value is reminiscent of the original XCorr score employed by SEQUEST but is designed to take simultaneous advantage of statistical calibration and high-

resolution tandem mass spectra. By combining residue-evidence p-values with XCorr p-values, we obtain state-of-the-art performance in identifying tandem mass spectra. The resulting search engine is freely available in the open source Crux mass spectrometry toolkit (`http://crux.ms`).

The number of search engines available to process shotgun proteomics mass spectrometry data is large and growing (reviewed in[68]). Given this diversity of approaches and the different results produced by each search engine, it is only natural to attempt to search the same data with multiple methods and combine the results in a post-processing stage. Indeed, a wide variety of methods have been developed that adopt this approach, including methods that aggregate PSMs and then re-estimate FDRs on the aggregated results,[87,93,95] combine statistical confidence measures,[2] compute probabilities for each method and then combine these probabilities,[67,83,84,88] or run a machine learning post-processor on the combined results.[16] Our empirical results suggest that the res-ev score function and its combined p-value provide yet another complementary view of peptide-spectrum matching, which will likely add value in the context of such aggregation schemes.

One potential explanation for the relatively poor performance of MS-GF+ on the *Plasmodium* dataset is the unusual nature of the data itself. Compared to the other data sets that we investigated, this is the only one that uses Lys-C rather than trypsin for digestion, and is also the only data set that includes TMT labeling. MS-GF+'s scoring model includes a supervised machine learning component. We hypothesize that performing well on the *Plasmodium* data might require a model that was trained on data with TMT labeling and digestion with Lys-C. In contrast, the combined p-value approach is invariant to properties of the data, such as digestion and labeling schemes.

One drawback to any scheme that involves computing multiple scores per PSM is the resulting increase in running time. In the case of the res-ev score, because the dynamic programming operation effectively includes an additional dimension (corresponding to the 20 amino acids), calibration is more expensive than calibration of XCorr scores. In our experiments, we observe that a search using res-ev p-value takes $\sim$3 times as long as a search

using XCorr p-value (specifically, 3.2x for the *Plasmodium* data set and 2.7x for the human data set). The combined p-value running time is very close to the sum of the running times for the two separate scores.

A potential area for future work lies in the method for combining XCorr and res-ev p-values. We empirically set the parameter m, which represents the degree of dependency between the two types of p-values, to be 1.2. However, in principle this value could be re-estimated for each new data set, using a strategy similar to that shown in Figure 2.3. However, our empirical results (not shown) suggest that the behavior of the method is not strongly dependent upon the choice of m.

The strong performance of the Morpheus search engine as measured by TMP suggests that this score does a good job of identifying the generating peptide for a given spectrum. On the other hand, the poor overall performance of Morpheus suggests that the score function is poorly calibrated. This is not surprising, because the score is simply the sum of two terms: the number of matched product ions, and the fraction of the observed peak intensities that can be assigned to matched products. Thus, longer peptides or spectra with more peaks will tend to achieve higher Morpheus scores on average. Our results suggest that a calibrated version of this score function should be able to achieve very good empirical performance.

In addition to proposing a new score function, we have provided a new benchmark for use in evaluating novel score functions. On the surface, it seems deceptively easy to compare results across score functions: run the score functions on the set same of input spectra and peptides and then compare the results. However, in reality, it is much harder to fairly compare score functions because search engines differ in many ancillary ways: digestion rules, decoy generation, etc. By merging all of the PSMs from different search engines, for a given dataset, into a single table indexed by scan number, and by ensuring that all the reported peptides appear in a shared peptide list, we ensure that the performance evaluation focuses on properties of the score function, rather than less interesting properties of the digestion rules or candidate peptide selection procedure.

Chapter 3

IMPROVING POWER WHILE CONTROLLING THE FALSE DISCOVERY RATE WHEN ONLY A SUBSET OF PEPTIDES ARE RELEVANT

3.1 Introduction

In a typical proteomics database search, mass spectra are searched against a database consisting of peptides reasonably expected to be found in the sample. For example, mass spectra generated from human cells would be searched against the human proteome. Following the database search, target-decoy competition (TDC) is used to control the false discovery rate (FDR) in the reported set of peptide-spectrum matches (PSMs).[19,20] Using TDC with an FDR threshold α, typically $\alpha = 1\%$, we can identify a set of detected PSMs for which the expected proportion of false discoveries, or FDP, is $\leq \alpha$. Although this process is standard across the field and is valid for many proteomic analyses, there are situations where this FDR control strategy can be problematic.

Specifically, consider the case where researchers are only interested in a subset of "relevant peptides" present in the sample. Determining whether a peptide is relevant or not is up to the user, but typically it makes sense to define a peptide as *relevant* when the detection of that peptide is pertinent to the hypothesis being asked. Similarly, irrelevant peptides can typically be defined as peptides that are inconsequential to the question being asked. Of course, defining the set of relevant peptides should be done before looking at the data, just as in hypothesis testing we should formulate the null and alternative hypotheses prior to performing the test.

For example, a common class of irrelevant peptides is human contaminants, typically keratin, which can be artificially introduced into a non-human sample during sample prepa-

ration. Human keratin contaminants are irrelevant because detecting these peptides does not affect the biological interpretation of the data. In this specific scenario, the peptide database will mostly consist of relevant peptides, since the list of contaminants is typically small. However, there are other scenarios where the proportion of relevant peptides in the protein database is small. One example is when proteomics is used to study the biology of *Plasmodium falciparum*, the causative agent of malaria. When *P. falciparum* is studied in a laboratory, it must be cultured in a medium containing human red blood cells. Therefore, any *P. falciparum* sample will also contain human peptides. In this setting, the detection of human peptides is irrelevant since the goal of the experiment is to study the biology of *P. falciparum*. In practice, researchers search their experimental spectra against the concatenated human and *P. falciparum* proteomes. Since the human proteome is much larger than the *P. falciparum* proteome, the proportion of relevant peptides in the combined database is small.

The proportion of relevant peptides in the database can be even smaller. Generally this occurs when investigators are pursuing a focused biological hypothesis such as the effect of a drug on a single molecular pathway or the effect of a perturbation on a single organism in a microbial community. One concrete example that we consider in some detail here is the detection of the protein toxin ricin, RCA60. Detection of the ricin toxin is important because the possession, transfer, or use of this toxin is federally regulated throughout most of the world. The challenge for law enforcement is that the castor plant and the seeds of the castor plant, where ricin is expressed, are not regulated. In fact, there are many legitimate reasons to possess and transfer the plant and seeds. For example, the castor plant is a common ornamental plant, and castor seeds are used in the production of castor oil. As a result, prosecutors must be able to directly detect the ricin toxin. All other proteins expressed in the castor plant are not useful for prosecution. This means that a single protein is relevant while the entire remaining proteome is irrelevant.

In such settings, where our interest lies in a small subset of relevant peptides, one needs to be particularly careful when applying TDC to control the FDR. Indeed, one intuitively

appealing strategy involves controlling the FDR on PSMs from the full database search but only reporting the relevant PSM subset.[55,90] It has been previously argued that this strategy, which we call "search-then-select," generally does not properly control the FDR because the relevant PSM score distribution does not necessarily match the irrelevant PSM score distribution.[18,101] For example, in the case where researchers are interested in post-translationally modified peptides, the score distribution of irrelevant unmodified peptides can differ from relevant modified peptides.[24,25]

Motivated by this problem, three different modifications to the standard TDC protocol have been proposed when controlling the FDR among a subset of relevant PSMs (Table 3.1 and Figure 3.1). One method is to simply search the spectra against the database consisting only of relevant peptides ("subset-search").[69,70] A second method searches the spectra against relevant and irrelevant peptides but applies FDR control only to relevant PSMs ("group-FDR"[17]) (also sometimes referred to as "separate FDR"[101]). Finally, in the third method, spectra are searched against all peptides but FDR is controlled with respect to only the set of relevant peptides using information from all PSMs.[25,86,101] For this analysis, we chose the method from Sticker *et al.* ("all-sub") as a representative example of this third type of method.[86]

Our interest in this problem was rekindled when analyzing data where the only relevant peptides were those of the ricin protein. This is an example where the relevant peptides comprise a very small subset of the peptides in the database. However, with this data we also encountered a new phenomenon—"neighbor" peptides—that has not previously been explicitly taken into account in this context. A neighbor peptide is an irrelevant peptide that has a similar precursor mass and fragmentation (MS2) spectrum as a relevant peptide. As explained below, ignoring the existence of neighbor peptides can compromise an FDR controlling procedure. To understand the crux of the problem imagine that a relevant peptide is missing from the sample but a neighbor of it is present. When searching only against the relevant peptides, the spectrum generated by the present neighbor will offer a very good (false) match to the relevant peptide even though the latter is not present in the sample.

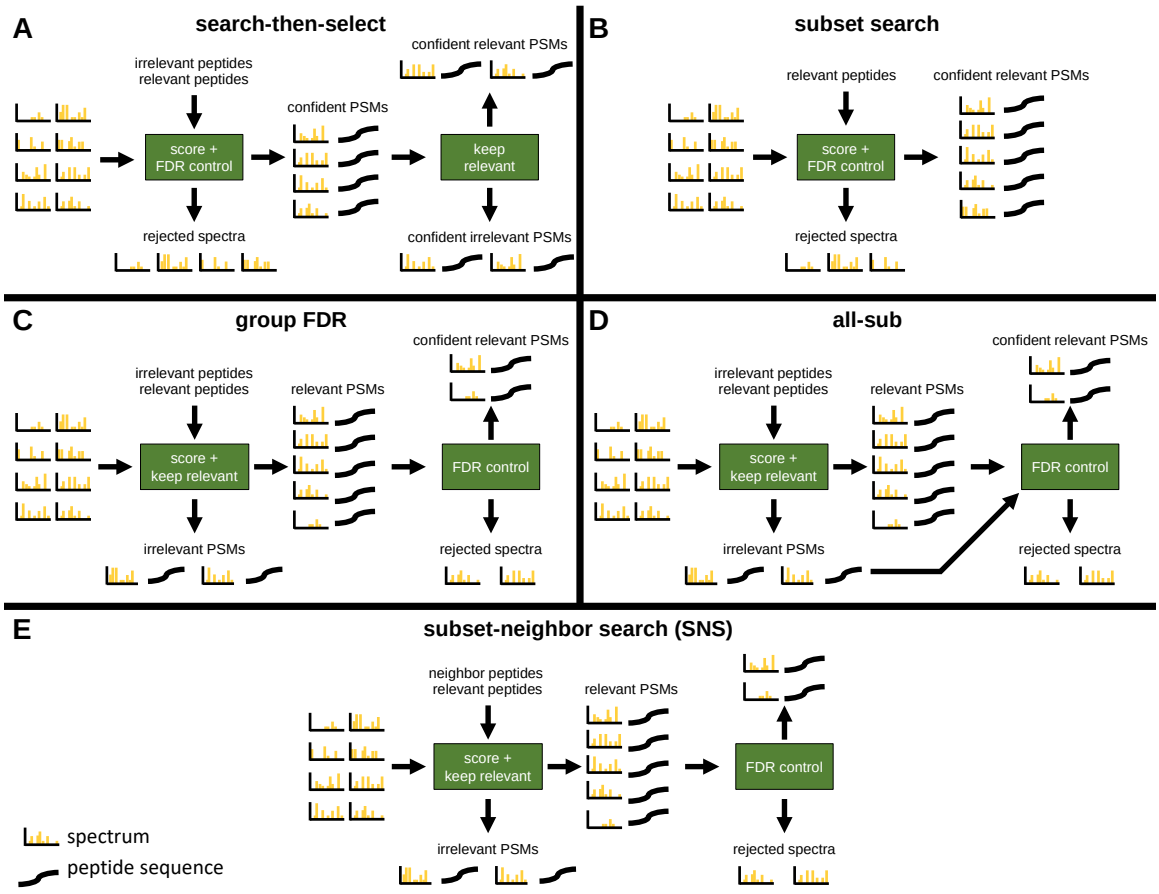


Figure 3.1: **Graphical overview of methods.** "Keep relevant" means that any PSM that involves a relevant peptide is kept while every other PSM is removed. "Keep irrelevant" means any PSM that involves an irrelevant peptide is kept while every other PSM is removed. Neighbor peptides are defined and explicitly considered separate from irrelevant peptides only in "subset-neighbor search" (SNS). The difference between C and E is the input into the "score + keep relevant" box. The input to E are neighbor and relevant peptides. The input to C are irrelevant peptides, which includes neighbor peptides, and relevant peptides.

Our investigation of existing analysis procedures starts by evaluating their ability to control the FDR. Others have already pointed out that search-then-select fails to control the FDR.[18,24,25,101] Here we also demonstrate that all-sub can fail to properly control the FDR when the relevant peptides comprise a small subset of the peptides in the database. In contrast, our initial analysis indicated that group-FDR and subset-search do not fail this test. However, subsequent analysis shows that subset-search can fail when a sufficient number of neighbor peptides are thrown into the mix. Specifically, in the analysis of our ricin data, we observe cases of spectra, presumably generated by neighbors of ricin peptides, that offer a very good match to the corresponding ricin peptide, even though the latter may not be in the sample. These potentially incorrect PSMs cannot be accounted for by the target-decoy competition in subset-search, since this process does not account for the existence of peptides that are not relevant but are neighbors.

These experiments left us with group-FDR as the only procedure that properly controls the FDR for the general case of searching a small subset of relevant peptides. However, like search-then-select and all-sub, group-FDR suffers from the problem of searching the relevant spectra against a large irrelevant database, thereby compromising its power.[70] This observation motivated our introduction of a new method, called "subset-neighbor search" (SNS), that tries to retain most of subset-search's ability to limit the search to the relevant part of the database while fixing the latter's failure to control the FDR by explicitly accounting for neighbor peptides (Table 3.1 and Figure 3.1).

Our analysis shows that SNS offers greater statistical power than group FDR, i.e., SNS typically delivers more discoveries than group FDR at a fixed FDR threshold. Given that none of the other methods offer robust FDR control, SNS is our recommended method when a small subset of the peptides in a sample is relevant.

3.2 Methods

Notation is summarized in Table 3.2.

	search-then-select	subset-search	all-sub	group-FDR	SNS
Consider neighbor peptides?					✓
Database to search	R+I+N	R	R+I+N	R+I+N	R+N
Apply group-FDR?			variation	✓	✓

Table 3.1: **Summary of methods.** We represent irrelevant peptides with an 'I', neighbor peptides with an 'N', and relevant peptides with an 'R'. If a database consists of multiple groups of peptides, then a '+' is used. Therefore, a database consisting of both relevant and neighbor peptides would be represented as 'R+N'. Group-FDR is with respect to R for the "Database to Search" step.

3.2.1 Neighbor peptides

We say that two peptides are "neighbors" if their masses are similar and if they share sufficiently many singly-charged b- and y-ions. More formally, we say peptides p_1 and p_2 are neighbors if (1) the difference in the associated peptide masses m_1 and m_2, specified in units of ppm, is less than a specified mass tolerance t_m:

$$\frac{|m_1 - m_2|}{\frac{1}{2}(m_1 + m_2)} 10^6 \leq t_m,$$

and (2) the proportion of b- and y-ions shared by p_1 and p_2 is greater than a specified fraction t_i:

$$\frac{2B_{12}}{B_1 + B_2} > t_i,$$

where B_1 (respectively, B_2) is the number of possible singly-charged b- and y-ions that can be associated with peptide p_1 (p_2), and B_{12} is the number of shared such ions between the two peptides. Here, two (theoretical) ions are considered shared if their m/z values, discretized to bins of size 0.05 Da, fall in the same bin. In this work, we set t_m equal to twice the precursor mass tolerance used in the associated database search, and we set $t_i = 0.25$.

3.2.2 FDR control methods

We consider five methods for controlling the FDR (the first four of which have been previously published, Table 3.1 and Figure 3.1). Pseudocode descriptions of each algorithm can be found in Appendix A.

Search-then-select This method takes as input a set S of observed spectra, a database $\mathcal{T}$ of target peptides, a corresponding database $\mathcal{D}$ of decoy peptides, a database $\mathcal{T}_r \subset \mathcal{T}$ of relevant peptides, and a confidence threshold α. The spectra are searched against the concatenated target-decoy database $\mathcal{T} \cup \mathcal{D}$, yielding an optimal PSM for each spectrum. The FDR for the set of PSMs is calculated using Equation 1.1 where $T(s)$ is the number of target PSMs with score $\geq s$ and $D(s)$ is the number of decoy PSMs with score $\geq s$. In the algorithm, the set of confident discoveries $A = A(\alpha)$ is defined as all PSMs whose score is $\geq s_\alpha$, where

$$s_\alpha := \min_s \widehat{\mathrm{FDR}}(s) \leq \alpha \tag{3.1}$$

and α is the user defined FDR threshold (typically 1%). Thus far, the procedure corresponds to the standard target-decoy competition protocol for FDR control.[19,20] In the search-then-select protocol, the subset of target peptides in A that do not involve a relevant peptide are filtered out, leaving only target PSMs involving peptides of interest. This final set of PSMs is designated as the set R of accepted PSMs.

subset-search The subset-search method[69,70] is similar to the standard target-decoy competition protocol, except that it operates only on the set of relevant peptides $\mathcal{T}_r$. The procedure takes as input the observed spectra S, a set of relevant target peptides $\mathcal{T}_r$, the set of corresponding decoy peptides $\mathcal{D}_r$, and the significance threshold α. The FDR is estimated using (1.1), and the discovery list A is determined using the score cutoff s_α from (3.1).

Group-FDR In the group-FDR method[17] (also known as separate FDR[101]), the inputs include observed spectra S, a relevant target database $\mathcal{T}_r$, a relevant decoy database $\mathcal{D}_r$, an

Variable	Definition
S	set of all observed spectra
$\mathcal{T}$	target database of all peptides
$\mathcal{D}$	decoy database of all peptides
$\mathcal{T}_r$	target database of relevant peptides
$\mathcal{D}_r$	decoy database of relevant peptides
$\mathcal{T}_i$	target database of irrelevant peptides
$\mathcal{D}_i$	decoy database of irrelevant peptides
$\mathcal{T}_n$	target database of neighbor peptides
$\mathcal{D}_n$	decoy database of neighbor peptides
α	user defined FDR threshold
α'	user defined filtering FDR threshold
$T(s)$	number of target PSMs with score better than s
$D(s)$	number of decoy PSMs with score better than s
$T_r(s)$	number of relevant target PSMs with score better than s
$D(-\infty)$	total number of decoy PSMs
$T_r(-\infty)$	total number of relevant target PSMs
$D_r(-\infty)$	total number of relevant decoy PSMs
P	set of peptides that result from a database search
M	set of scores that result from a database search
R	set of accepted PSMs
p_i	a peptide
m_i	precursor mass associated with peptide p_i
B_i	number of b- and y-ions associated with peptide p_i
$B_{i,j}$	number of shared b- and y-ions between peptides p_i and p_j
t_m	mass tolerance to define neighbor peptides
t_i	shared ion fraction tolerance to define neighbor peptides

Table 3.2: **Variables and their definitions.** Note that there is no overlap between the relevant and irrelevant peptides. Thus, $\mathcal{T} = \mathcal{T}_r \cup \mathcal{T}_i$ and $\mathcal{T}_r \cap \mathcal{T}_i = \emptyset$. In addition, if neighbor peptides are not defined, then they are considered to be irrelevant. However, if neighbor peptides are defined, then they are considered distinct from irrelevant peptides.

irrelevant target database $\mathcal{T}_i$, an irrelevant decoy database $\mathcal{D}_i$, and a threshold α . In this search, neighbor peptides ($\mathcal{T}_n$) are absorbed into the irrelevant peptide set ($\mathcal{T}_i$); therefore $\mathcal{T}_n = \emptyset$. A database search is conducted against a concatenated database consisting of all targets and decoys $\mathcal{T} \cup \mathcal{D}$ (which is the same as $\mathcal{T}_r \cup \mathcal{T}_i \cup \mathcal{D}_r \cup \mathcal{D}_i$). After the search, PSMs that involve irrelevant peptides or their associated decoys are filtered out. The FDR is then estimated on the remaining set of PSMs using (1.1), and the discovery list is determined using the score cutoff s_α defined by (3.1).

All-sub The all-sub method[86] is conceptually similar to group-FDR except for a variation in how the FDR is estimated. In particular, the input to all-sub is the same as for group-FDR. Spectra are searched against the concatenated target-decoy database $\mathcal{T} \cup \mathcal{D}$, but instead of using (1.1), the FDR is estimated using

$$\widehat{FDR}(s) = \min\left(1, \frac{D_r(-\infty)+1}{T_r(-\infty)}\right)\frac{D(s)T_r(-\infty)}{D(-\infty)T_r(s)}, \tag{3.2}$$

where $T_r(s)$ is the number of relevant target PSMs with scores $\geq s$, $T_r(-\infty)$ is the total number of relevant target PSMs, $D_r(-\infty)$ is the total number of relevant decoy PSMs, and $D(-\infty)$ is the total number of decoy PSMs.

Subset-neighbor search (SNS) As its name suggests, SNS is the only method presented here that explicitly accounts for neighbors. SNS is the same algorithm as group FDR except that neighbor peptides play the role of irrelevant peptides. Specifically, the spectra are searched against a database of relevant and neighbor peptides $\mathcal{T}_r \cup \mathcal{T}_n \cup \mathcal{D}_r \cup \mathcal{D}_n$. Following the search, PSMs that involve neighbor peptides or their associated decoys are removed from consideration, and the list of discoveries is determined from (1.1) and (3.1), as described above, using threshold α.

3.2.3 Datasets

To evaluate the various FDR control methods, we use four tandem mass spectrometry datasets (Table 3.3, Appendix Table A.1–A.3), which have been deposited in the PRIDE

relevant file	scans	irrelevant file	scans
UPS_1_22.ms2	10552	yeast_1_45.ms2	71631
UPS_2_23.ms2	10178	yeast_2_46.ms2	71611
UPS_3_24.ms2	9794	yeast_3_47.ms2	71931

Table 3.3: **UPS1 and yeast data.** The table list the number of scans found in each UPS1 and yeast run. In the database search, the relevant file on the left was concatenated to the irrelevant file on the right. Note that all file names start with "UWPRLumos_20190515_DP_DDA_".

Archive (`http://www.ebi.ac.uk/pride/archive`) with the dataset identifier PXD022778.

UPS1/Yeast This dataset consists of six mass spectrometry runs. Three runs came from a yeast whole cell lysate, and the remaining three runs came from the Universal Proteomics Standard Set 1 (UPS1, Sigma-Aldritch). The resulting spectra were then merged to create three *in silico* mixtures.

To prepare the yeast sample, yeast strain BY4741 (MATa his3Δ1 leu2Δ0 met15Δ0 ura3Δ0) (Dharmacon) was cultured in YEPD to mid-log phase, harvested, and lysed with 8M urea buffer solution and bead beating (7 cycles of 4 minutes beating with 1 min rest on ice). The resulting cell lysate was reduced, alkylated, and digested for 16 hours. Next, the peptide digest was desalted using a mixed-mode (MCX) method, dried down overnight via speedvac, and brought up with synthetic iRT peptide standards (Pierce Peptide Retention Time Calibration Mixture) to 1μg/μl total proteome. A bicinchoninic acid assay (Pierce BCA Protein Assay Kit) was used to determine total protein content.

To prepare the UPS1 sample, the Universal Proteomics Standard Set 1 (Thermo Scientific) was reduced, alkylated, and digested for 16 hours in the same manner as the yeast sample.

For both prepared samples, peptides were separated with a Waters NanoAcquity UPLC

and emitted into a Orbitrap Fusion Lumos (Thermo Scientific, San Jose, California). Pulled tip columns were created from 75 μm inner diameter fused silica capillary (New Objectives, Woburn, MA) in-house using a laser pulling device and packed with 3 μm ReproSil-Pur C18 beads (Dr. Maisch GmbH, Ammerbuch, Germany) to 30 cm. Trap columns were created from 150 μm inner diameter fused silica capillary fritted with Kasil on one end and packed with the same C18 beads to 3 cm. Solvent A was 0.1% formic acid in water and solvent B was 0.1% formic acid in 98% acetonitrile. For each injection, 3 μl (approximately 1 μg total protein on column) was loaded and eluted using a 90-minute gradient from 5 to 35% B, followed by a 40 minute wash. Data were acquired using data-dependent acquisition (DDA).

To acquire DDA data, the Orbitrap Fusion Lumos was set to positive mode in a top-20 configuration. Precursor spectra (400–1600 m/z) were collected at 60,000 resolution to hit an AGC target of 3×10^6. The maximum inject time was set to 100 ms. Fragment spectra were collected at 15000 resolution to hit an AGC target of 10^5 with a maximum inject time of 25 ms. The isolation width was set to 1.6 m/z with a normalized collision energy of 27. Only precursors charged between +2 and +4 that achieved a minimum AGC of 5×10^3 were acquired. Dynamic exclusion was set to "auto" and to exclude all isotopes in a cluster. Both the UPS1 peptides and the yeast peptides were injected into the mass spectrometer three times (Table A.1).

UPS1/Yeast A second UPS1/yeast dataset was used for a power analysis, and the data for this dataset was downloaded from PRIDE (project number PXD001819).[76] In this study, UPS1 proteins were spiked into a yeast cell lysate at nine different concentrations: 50 amol/μg, 125 amol/μg, 250 amol/μg, 500 amol/μg, 2.5 fmol/μg, 5 fmol/μg, 12.5 fmol/μg, 25 fmol/μg, 50 fmol/μg. Three technical replicates were generated for each of the nine samples. Runs corresponding to UPS1 spike-in concentrations of $\leq$2.5 fmol/μg were removed from consideration because these runs had zero confident UPS1 peptide detections at 1% FDR. This resulted in 12 usable runs for our analysis.

The yeast lysate was created in an 8M urea/0.1 M ammonium bicarbonate buffer at a

protein concentration of 8 μg/μL. UPS1 proteins were spiked into 20 μg of the yeast lysate to create nine different concentrations of UPS1. Following the spike-in step, the sample was reduced and alkylated. Digestion was done overnight using trypsin in a 1M urea buffer. Following the digestion step, each sample was desalted and analyzed in triplicate on a nanoRS UHPLC system (Dionex, Amsterdam, The Netherlands) coupled to an LTQ-Orbitrap Velos mass spectrometer (Thermo Fisher Scientific, Bremen, Germany) in top 20 data-dependent acquisition mode with dynamic exclusion set to 60 seconds.

Two μL of each sample were loaded into a C-18 column (75 μm IDx15 cm, in-house packed with C-18 Reprosil) where solvent A consists of 5% acetonitrile and 0.2% formic acid and solvent B consists of 80% acetonitrile and 0.2% formic acid. The flow rate was 300 nL/min flow rate. For the first 75 minutes of the gradient the percentage of solvent B increased from 5% to 25%. During the next 30 minutes, solvent B increased to 50% and finally during the last 10 minutes solvent B increased to 100%. MS1 scans were acquired in the Orbitrap on the 300–2000 m/z range with the resolution set to 60,000.

Ricin Data for this dataset was downloaded from PRIDE (project ID PXD007933). In this study, castor seeds from various castor plant cultivars were collected for sample processing.[63] Crude castor seed extracts were prepared from castor seeds via five different methods, designated M0 through M4. Method M0 involves mashing the seed to a semi-uniform consistency. M1 first removes the seed coat by soaking the seeds in sodium hydroxide, then mashes the peeled seeds. M2 takes the product from M1 and washes it with acetone. M3 and M4 involve a protein precipitation step where either magnesium sulfate or acetone, respectively, is used. Following preparation of the crude castor seed extracts, the extracts were inactivated by heating to 100 C for 1 hour.

The crude castor seed extracts were placed in a buffer (PBS from 10x concentrate, Fluka, containing 11.9 mM phosphates, 137 mM sodium chloride, 2.7 mM potassium chloride, pH 7.4, to which 0.01% Tween-20 was added) and then centrifuged for 10 minutes at 4°C and 16,000 g. Following centrifugation, the resulting aqueous layer was incubated with urea

and 500 mM dithiothreitol (DTT) for 60 minutes at 37°C. Then, 400 mM iodoacetamide (IAA) was added and then incubated in the dark at 37°C for 60 m to alkylate cysteine thiol groups. Once alkylated, samples were diluted using ammonium bicarbonate buffer to which calcium choloride was added. Samples were digested overnight at 30°C using trypsin. After digestion, samples were acidified and desalted by solid phase extraction.

A Waters NanoAcquity dual pump LC system (Waters, Milford, Massachusetts) was used to separate peptides on the column. Peptides were separated on a fused silica capillary column which consisted of a trapping column (4 cm x 150 μm inner diameter) and an analytical column (70 cm x 75 μm inner diameter, 360 μm outer diameter). Peptides were eluted at a rate of 300 nL/min for 150 m using a nonlinear gradient. A Q Exactive Plus or Q Exactive HF mass spectrometer (Thermo Scientific, San Jose, California) was used to collect the data. Both MS and MS/MS spectra was collected at high resolution. Specifically, the Q Exactive Plus precursor spectra were acquired at 35,000 resolution (mass resolving power) for precursor spectra and MS/MS spectra at 17,500 resolution, while on the Q Exactive HF, the resolution settings were 60,000 for precursor spectra and 15,000 for MS/MS spectra.

The ricin dataset consists of 54 different runs of castor seed extracts, corresponding to multiple replicates for each of the five sample preparation protocols, and are summarized in Appendix Table A.2.

Human A subset of 12 human mass spectrometry runs from PRIDE project PXD011189 were downloaded. This study developed a new sample preparation protocol, sample preparation by easy extraction and digestion (SPEED),[13] and compared it to three previously developed methods of filter-aided sample preparation (FASP), single-pot solid-phase-enhanced sample preparation (SP3), and urea-based in-solution digestion (Urea-ISD). To generate biomass HeLA cells (ATCC CCL-2) were grown in DMEM media at 37 °C and supplemented with 10% FCS and 2 mm l-Glutamine. After sample collection, cells were washed with PBS and then pelleted and stored at -80 °C until the lysis step.

In the SPEED method, samples were resuspended in trifluoroacetic acid (TFA) and then

incubated at room temperature for 2 minutes. After the samples were neutralized using a solution of 2 M TribaseNext and TFA, Tris(2-carboxyethyl)phosphine (TCEP) and 2-Chloroacetamide (CAA) was added. After the TCEP and CAA was added the samples were incubated for 5 min at 95 °C. Following the incubation step, a trypsin digestion step was conducted for 20 hour at 37 °C. Finally, the peptide mixture was acidified and desalted.

In FASP,[99] samples were suspended in a solution of 4% SDS, 100 mm Tris/HCl, and 100 mm DTT, incubated at 95 °C for 5 min, and then sonicated at 4 °C to induce lysis. Following the lysis step, samples were processed using a Microcon-30kDa Centrifugal Filter Units (Merck). With this unit the sample underwent several rounds of centrifugation in a solution of 8 M urea and 0.1 M Tris-HCl. During one round of centrifugation the peptide were alkylated using IAA. Following all rounds of centrifugation, the filter device was rinsed and desalted. The filtrate were digested for 20 h at 37 °C using trypsin at a protein/enzyme ratio of 50:1.

In the SP3 method,[85] cells were lysed in a solution of 1% SDS, 1x complete Protease Inhibitor Mixture (Roche, Basel, Switzerland), and 50 mm HEPES buffer. During the lysis step the samples were incubated at 95 °C for 5 min and further sonicated for 300 seconds at 4 °C. Following the lysis step samples were reduced and alkylated using DTT and IAA, respectively. Then, 2 μL of paramagnetic beads was added to the mixture. After the beads were immobilized by a magnetic rack for two minutes, they were washed and resuspended in a trypsin solution and digestion was carried out for 20 h at 37 °C. Peptides were eluted off the beads using a 2% dimethyl sulfoxide in water.

In the urea-ISD method, cells were lysed by suspending the sample in a pH 8 solution of 8 m urea, 50 mm Tris-HCl, and 5 mm DTT. Then, it was sonicated for 10 min at 4 °C. Following the lysis step, the samples were incubated for 1 h at 37 °C and then centrifuged for 5 min. Samples were alkylated for 30 min at room temperature in the dark using IAA. Proteins were digested using trpysin for 20 hours at 37 °C in a urea solution with 50 mm Tris-HCl.

Once all the samples had been prepared they were all analyzed on a EASY-nanoLC 1200

relevant database	peptides	irrelevant database	peptides	# neighbor peptides
UPS1	2,552	yeast	611,861	3,459
ricin	80	non-ricin castor plant	2,340,370	155
human protein P18206	364	all other human proteins	2,653,948	1,863
human protein Q9Y490	506	all other human proteins	2,653,757	2,418
human protein P07900	102	all other human proteins	2,654,127	358
human protein P08238	90	all other human proteins	2,654,130	182
human protein P10809	136	all other human proteins	2,654,122	521

Table 3.4: **Databases used in database searches.** For the UPS1/yeast database and the five iterations of the human database, peptides in common between the relevant and irrelevant database were removed from analysis. Any relevant ricin peptide also found in the non-ricin castor plant proteome was considered to be relevant. The number of irrelevant peptides includes the set of neighbor peptides.

(Thermo Fisher Scientific) coupled online to a Q Exactive Plus mass spectrometer (Thermo Fisher Scientific). One *mu* g of sample was injected into a 50 cm Acclai PepMap column (Thermo Fisher Scientific) using a linear 180 min gradient of 3 to 28% acetonitrile in 0.1% formic acid at a 200 nL/min flow rate. The Q Exactive Plus was operated in a top 10 data-dependent acquisition mode with a dynamic exclusion window of 30 seconds and collected scans in the m/z range of 300–1650. MS1 scans were acquired with a resolution of 70,000 and fragment scans were recorded with a resolution of 17,500.

For all datasets, files in .ms2 format[61] were generated from the Thermo RAW file vendor format using Proteowizard version 3.0.[41]

3.2.4 Database search

For each database search, two protein databases were employed, one designated "relevant" and one "irrelevant" (Table 3.4). The databases for the castor plant (which contains the ricin protein), yeast strain ATCC 204508, and human were downloaded from Uniprot (`https://www.uniprot.org/`) in March 2018, May 2019, and January 2019, respectively. Protein sequences for the UPS1 proteins were downloaded from the Sigma Aldrich website (`https://www.sigmaaldrich.com`) in December 2018. The UPS1 and yeast sequences were concatenated into a single protein database. Each protein database was digested to peptides *in silico* using the tide-index tool in Crux version 3.2, allowing up to one missed cleavage, up to three methionine oxidations, and with clipped N-terminal methionines.[62,73] For the UPS1/yeast database we considered UPS1 peptides to be relevant and yeast peptides irrelevant. For the castor plant database we considered the ricin protein relevant and all other castor proteins irrelevant. Finally, for the human database we considered a single randomly chosen detectable human protein as relevant and all other human proteins irrelevant. This process was repeated five times for the human database (Uniprot ID: P18206, Q9Y490, P07900, P08238, and P10809). For the UPS1/yeast and human database, peptides found in both relevant and irrelevant proteins were removed from the analysis. On the other hand, castor plant peptides found in both relevant (ricin) and irrelevant (non-ricin) proteins were considered relevant. In all cases, decoy peptides were generated by randomly shuffling the target sequence while keeping the N-terminal and C-terminal amino acids fixed. This was done to maintain the amino acid frequencies of the N-terminus, C-terminus, and internal amino acids. A different decoy database was generated for each run, yielding 54 decoy castor plant databases, three decoy UPS1/yeast databases for the concatenated runs, 12 decoy UPS1/yeast databases for the previously published runs, and 60 decoy human databases.

Database searches were conducted using Tide with the combined p-value score function[52] against a concatenated target-decoy database. The precursor mass tolerances, estimated by Param-Medic,[58] were found to be 31 ppm for the concatenated UPS1/yeast runs, 80 ppm

for the castor plant runs, 63 ppm for the previously published UPS1/yeast runs, and 38 ppm for the human runs. All Tide parameters were set to their default values, except an isotope error of 1 was allowed and the "top-match" parameter (i.e., number of reported PSMs per spectrum) was set to 10,000. A post-processing script implemented each FDR estimation method. Once the list of PSMs was finalized, the Crux assign-confidence command was used to estimate FDR for all methods except for all-sub. Each PSM is assigned a q-value where the value is the minimum FDR at which the PSM is confidently detected. For all-sub, we created our own Python implementation of this method to estimate all-sub q-values.

3.2.5 Evaluating the validity of FDR control methods

To gauge whether each of the considered methods properly controls the FDR we tested whether the empirical mean of the false discovery proportion (FDP) is significantly different than the FDR threshold. The idea is that since the FDR is defined as the expectation of the FDP, controlling the FDR at level α implies that the empirical mean of the FDP, averaged across multiple independent runs, should converge to a number $\leq \alpha$. In particular, that empirical mean should not exceed α in a statistically significant manner.

Here we computed the empirical mean of the FDP by randomly dividing our data into ten, roughly equal parts, and we used a heuristic explained below to reliably approximate the FDP in each of the ten runs and hence its mean over the same ten runs.

We then performed a one-sided t-test asking whether the observed mean of the FDP is significantly larger than α. If the answer was positive then we had a reason to doubt the validity of the proposed FDR-controlling method. By the same token, an insignificant deviation does not prove the validity of the method, but it does lend some confidence in it.

To simulate multiple independent runs we randomly split a UPS1 and yeast run into 10 equal parts. After splitting, each UPS1 part was matched to a yeast part to create 10 sub-runs. Each of these sub-runs was used as input to a database search, as previously described, with each database search using a different decoy database.

Following the database search, and designating UPS1 peptides as relevant and yeast

FDR control method	p-value 1	p-value 2	p-value 3
subset-search	0.25695	0.02642	0.20952
all-sub	**0.0012944**	**0.0008953**	**0.0024908**
group-FDR	0.05458	0.04435	0.05003
subset-neighbor search	0.59906	0.04705	0.56590

Table 3.5: **Assessing FDR control.** Each p-value comes from a single t-test measuring whether the mean of the estimated FDP over the 10 sub-runs is significantly larger than the selected 5% FDR threshold. Each column uses a different computationally concatenated UPS1 and yeast run, and each row refers to a different FDR estimation procedure. Boldface values are significant at a Bonferroni corrected threshold of 0.004 (0.05/12). The analysis suggests that all-sub fails to control FDR.

peptides as irrelevant, we applied to the resulting set of PSMs the selection procedures of subset-search, all-sub, group-FDR, and SNS at a 5% FDR threshold.

The FDP in the FDR controlled set of PSMs was estimated by dividing the number of demonstrably incorrect PSMs (i.e., the number of times a yeast spectrum matched to a UPS1 peptide—additional detail can be found below) by the total number of discoveries. The FDP values from the ten sub-runs were used as input to the one-sided t-test. This process was repeated for the other two UPS1/yeast runs, yielding 30 sub-runs and three p-values.

3.3 Results

3.3.1 All-sub can fail to control FDR when the subset of interest is small

First we investigated whether subset-search, all-sub, and group-FDR each properly control FDR when the subset of interest is small. Note that we only test three methods because search-then-select has been previously shown to improperly control the FDR.[18,24,25,101]

To test these methods, we first estimated the FDP within an FDR-controlled set of PSMs.

To do so, we computationally mixed together an irrelevant yeast run with a relevant UPS1 run. As a result, any yeast spectrum that is matched with a UPS1 peptide is demonstrably incorrect. This allows us to give a lower bound on the FDP. It should be noted that in this real data we do not precisely know the FDP; however, we designed our experiment so that in practice we believe our lower bound (i.e., estimated FDP) is fairly close to the actual, unknown one. Specifically, due to the large difference in size between the irrelevant yeast and relevant UPS1 database, we expect most incorrect PSMs that occur by chance would involve yeast peptides. After the FDP has been estimated, we performed a t-test to determine whether the mean of the FDP is significantly larger than the FDR threshold α. A mean FDP that is significantly larger than the FDR indicates that the corresponding FDR control method is probably invalid.

Our analysis suggests that all-sub fails to properly control the FDR (Table 3.5). Using the all-sub method, we calculated p-values of 0.0012944, 0.0008953, and 0.0024908 for the three different concatenated UPS1/yeast datasets. These p-values suggest that all-sub improperly controls the FDR because they are smaller than the Bonferroni corrected threshold of 0.004, where the uncorrected p-value threshold is 0.05 and $n = 12$.

Note that the above argument cannot be used against the remaining methods as their corresponding p-values are all above the Bonferroni corrected threshold. This of course does not prove that these three methods correctly control the FDR; however, in the absence of neighbors one can argue that all three of these methods satisfy the conditions that guarantee FDR control by TDC.[27] Specifically, when we use TDC to control the FDR we make the implicit assumption that each incorrect PSM is equally likely to be a target win or a decoy win independently of all other PSMs as well as of the score of the PSM. The rationale behind this assumption is that, in the absence of neighbors, for an incorrect match the target database looks essentially the same as the randomly generated (or reversed) decoy database (as well as the dynamic exclusion that supports the independence assumption). If this rationale holds when searching the entire database then in principle the same should hold when we focus only on the relevant database when using group-FDR and SNS. Furthermore, the same holds

for subset-search as long as the irrelevant set does not include neighbors, which leads us to the next section.

3.3.2 Subset-search can fail to control the FDR in the presence of neighbors

Although our previous analysis indicates that subset-search properly handles a small subset of interest, we found that the method can struggle to control the FDR in the presence of neighbor peptides. The issue is that, since subset-search does not search the spectra against the irrelevant peptides, a spectrum generated by an irrelevant neighbor peptide will likely receive a high score against the corresponding relevant peptide. Considering all high scoring incorrect PSMs that are due to the presence of irrelevant neighbor peptides in the database, it is clear that these PSMs are more likely to be target wins (the spectra matching the closely resembling relevant peptide) than decoy wins (an unrelated random peptide). Hence, our assumption that each incorrect PSM is equally likely to be a target or a decoy win is violated. As a result, these incorrect target PSMs are more likely to be accepted as correct PSMs by target-decoy competition.

To give a concrete example, consider the theoretical MS2 spectrum of relevant ricin peptide "VGLPINQR" and irrelevant castor plant peptide "RIPLANGR" (Figure 3.2). These two peptides have a mass difference of approximately 12 ppm and have 69.56% of MS2 peaks in common. The PSM between the experimental scan (top row Figure 3.2) and the relevant peptide yields a combined p-value score of 1.25×10^{-4}, which is larger (i.e., worse) than the combined p-value score, 1.13×10^{-4}, of the PSM between the scan and the neighbor peptide. Hence, if the target database does not contain the neighbor peptides, then the database search would match this experimental scan with the relevant peptide even though it has an even better match with a neighbor peptide.

To test our hypothesis that subset-search may be problematic in the presence of peptide neighbors, we investigated the confident set of PSMs, in each run, identified by subset-search at 1% FDR. For each such scan we asked whether that scan would have scored more highly if neighbor peptides had been present in the database. We repeated this process for each

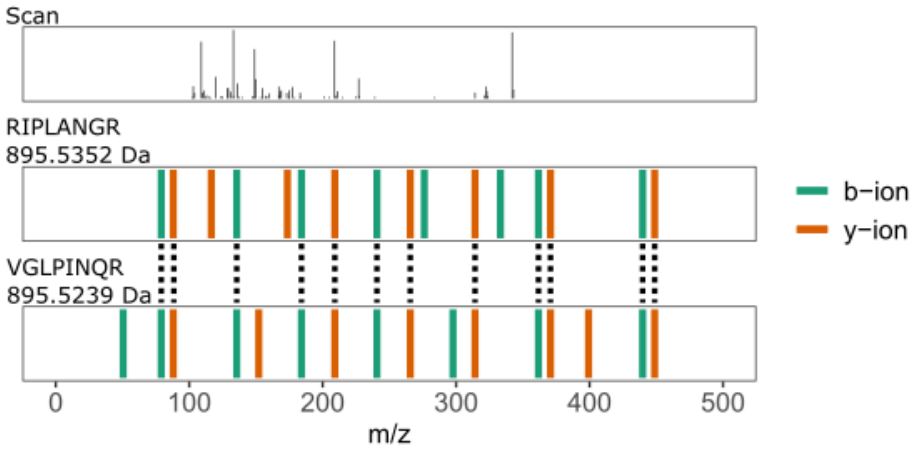


Figure 3.2: **Example of a neighbor peptide** This figure plots an experimental spectrum with a precursor charge of two (top) along with the best scoring neighbor peptide (middle) and the best scoring relevant peptide (bottom). Peptide "VGLPINQR" is a relevant ricin peptide (Uniprot ID: B9T8T0) and peptide "RIPLANGR" is an irrelevant castor plant peptide (Uniprot ID: B9T289). The mass difference between the two peptides is approximately 12 ppm with the mass of "VGLPINQR" being 895.5239 Da and the mass of "RIPLANGR" being 895.5352 Da. These two peptides have ∼70% of their MS2 peaks in common. Dotted lines connect MS2 peaks in the same 0.05 Da bin. The combined p-value score (lower is better) between the experimental scan with the relevant peptide is 1.25×10^{-4}, whereas the combined p-value score between the scan and the neighbor peptide is 1.13×10^{-4}.

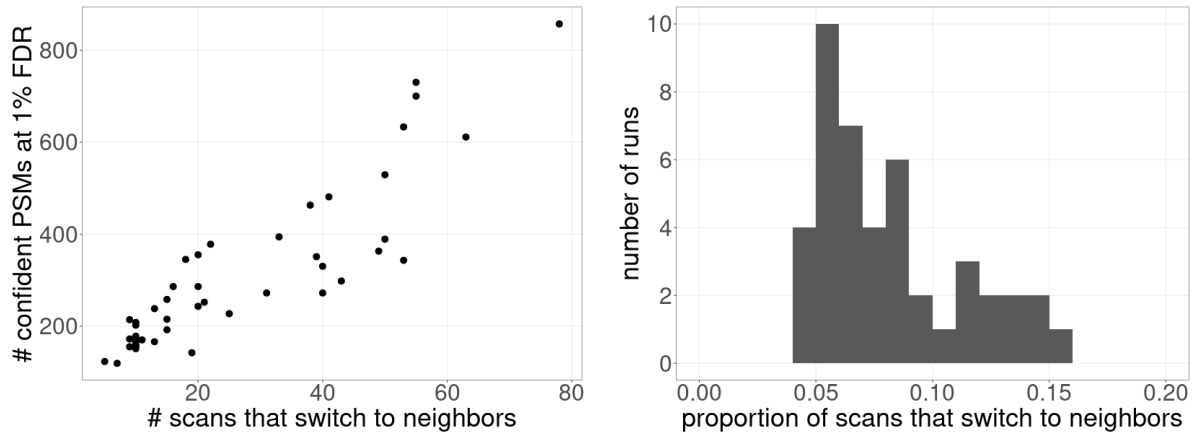


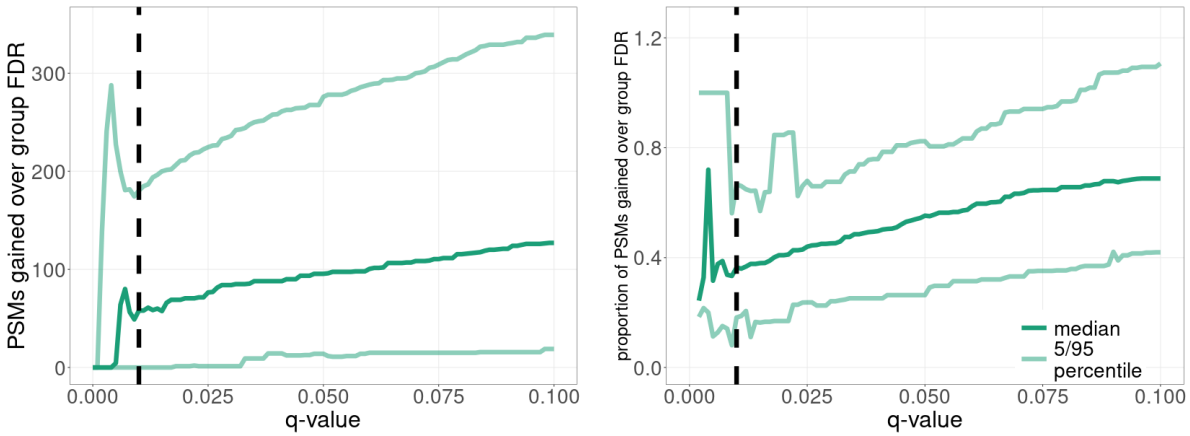
Figure 3.3: **Magnitude of the neighbor peptide problem** This plot shows how subset-search does not properly control the FDR in the presence of many neighbors. We took the set of confidently detected PSMs, as detected by subset-search at 1% FDR. From this set of confident PSMs we determined the number of scans that would have scored better to a neighbor peptide if that peptide had been present in the database. This process was repeated for 54 different castor runs. (A) Each point is the number of scans in a run that would have matched to a neighbor peptide if neighbor peptides were searched as a function of the number of confident PSMs. (B) A histogram of the values from (A), where the x-axis is divided by the y-axis to obtain the proportion of scans that switch to neighbors. Note there are only 44 points because 10 runs had zero confident PSMs at 1% FDR.

of the 54 different castor seed runs. We discovered that anywhere from 5 to 78 scans, in each set of confidently detected PSMs, switched their best scoring target from a relevant peptide to a neighbor peptide (Figure 3.3A). These scans comprise anywhere from 4.1% to 15.5% of the number of confidently identified scans (Figure 3.3B). Since the proportion of incorrect PSMs, due to lack of neighbor peptides in the database, is much larger than the FDR threshold of 1% FDR, this experiment suggests that indeed subset fails to control the FDR in this scenario.

3.3.3 SNS controls for neighbor peptides and outperforms group-FDR

Our analysis so far suggests that, among existing methods, group-FDR is the only method that properly controls the FDR when the relevant database is much smaller than the irrelevant database. However, group-FDR can suffer from low statistical power due to a high multiple testing hypothesis burden. This is because, in group-FDR, relevant spectra are searched against the entire database, which includes a large set of irrelevant peptides.[70]

For example, consider the ricin dataset. Looking at the difference in the median number of PSMs detected by subset-search and group-FDR at various FDR thresholds between 0–10% suggests that subset-search outperforms group-FDR across the entire q-value range of 0-10% (Figure 3.4). (We chose to use the median instead of the mean because we expect a different concentration of the relevant protein in each run, thereby changing the expected number of relevant PSMs.) This observation motivated us to develop a new FDR control method that would share much of the power advantage of subset-search while correctly controlling the FDR by explicitly accounting for neighbors. Our method, called "subset-neighbor search" (SNS), is similar to group-FDR except that irrelevant, non-neighbor peptides are excluded from the database. Thus, the non-relevant portion of the SNS database contains only the neighbor peptides, whereas in the case of group-FDR it includes neighbor and irrelevant peptides. We hypothesized that SNS should, in general, offer more discoveries than group-FDR because spectra are not searched against irrelevant peptides, resulting in a lower multiple testing hypothesis burden.



A B

Figure 3.4: **Comparison of subset-search and group-FDR.** This figure compares the performance of subset-search against group-FDR in the ricin dataset with respect to (A) the number of PSMs and (B) the proportional increase in number of PSMs. For each mass spectrometry run, we determine the difference in the number of PSMs detected between subset-search and group-FDR at various FDR thresholds. After collating these values across all runs, we plot the median value and 5/95 percentiles over 54 runs. The vertical dashed line is at the conventional 1% FDR threshold. Note that the plotted values in B are undefined for some q-values near 0 where neither subset-search nor group-FDR detects any PSMs.

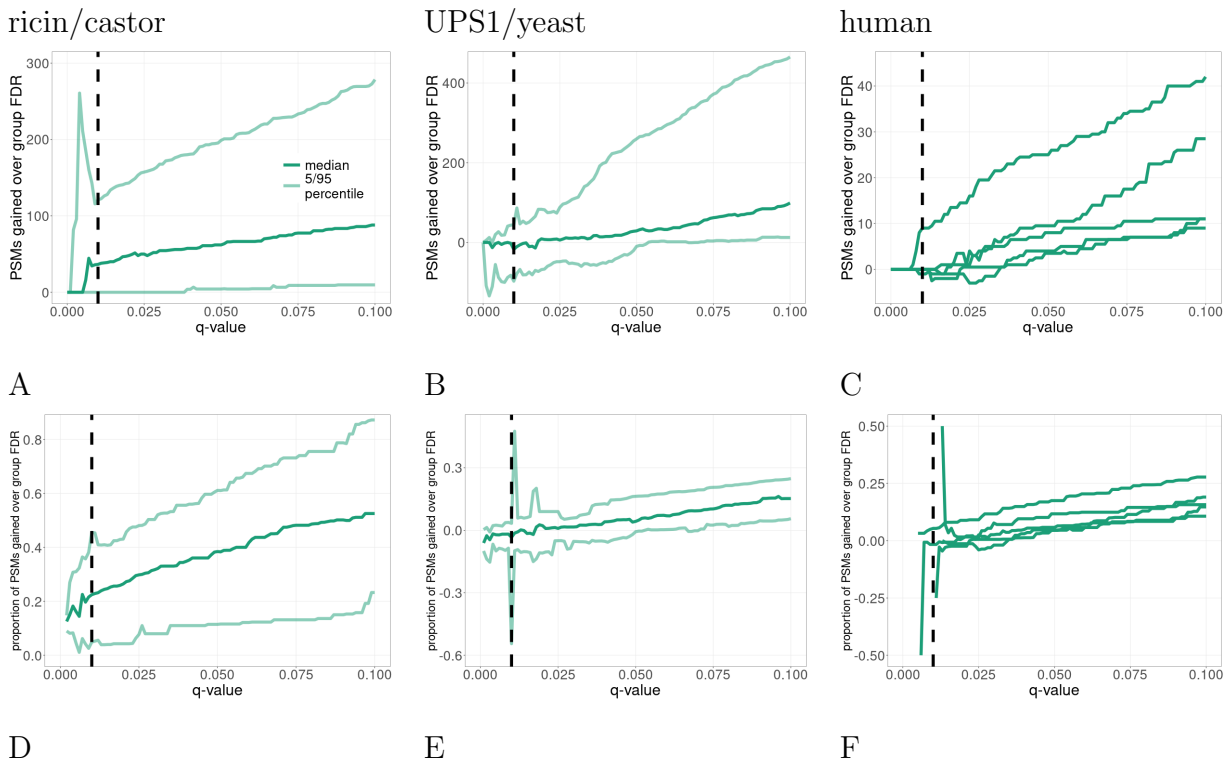


Figure 3.5: **Comparison of SNS and group-FDR.** This plot compares the relative performance of SNS against group-FDR for the ricin (A and D), UPS1/yeast (B and E), and human dataset (C and F). For each mass spectrometry run, we determine the difference in the number of PSMs (A–C) detected between SNS and group-FDR at various q-value thresholds. In addition, we calculate the corresponding proportional increase (D–F). For F, we assigned a value of 0.5 when SNS detects any number of PSMs while group-FDR detects 0 PSMs and -0.5 when group FDR-detects any number of PSMs while SNS detects 0 PSMs. The vertical dashed line is at the conventional 1% FDR threshold. Note the plotted values in D–F are undefined for some q-values near 0, where neither SNS nor group-FDR detects any PSMs. For the human data (C and F), we only plot the median lines, where each line represents a different relevant protein.

To test this in practice, we compared the performance of SNS and group-FDR on three different datasets: ricin, non-concatenated yeast/UPS1, and human. These three datasets provide several examples of cases where the relevant portion of the database is a tiny portion of the overall database. The ricin example provides a real world example of a situation where an investigator is interested in a tiny subset of the possible database (where the relevant subset is on the order of 10^{-5} the size of the overall database). We use the yeast/UPS1 dataset as an example where the relevant portion is larger but still a small subset of the overall database (on the order of 10^{-3}). Finally, we use the five iterations of the human database to show additional evidence (on the order of 10^{-4}–10^{-5}).

Empirically, we found that SNS indeed generally outperforms group-FDR across the three datasets (Figure 3.5). In the ricin dataset, SNS outperforms group-FDR across the entire FDR range of 0–10%. At a 1% FDR threshold, SNS outperforms group-FDR by a median difference of 36.5 PSMs (Figure 3.5A). Looking at percent differences, SNS outperforms group-FDR by a median percent difference of 22.5% (Figure 3.5C). Comparing the performance of SNS and group-FDR in the yeast/UPS1 dataset (Figure 3.5B and E), these two methods have comparable performance for small q-values ($\leq 2.5\%$). However, for larger q-values ($\geq 2.5\%$), SNS outperforms group-FDR. Specifically, looking at a 5% q-value threshold, SNS outperforms outperforms group-FDR by a median of 29 PSMs (4.3%). Finally, the five different iterations of the human dataset generally follow the trends previously described (Figure 3.5D and F). At low q-values, SNS and group-FDR have similar power. Specifically at 1% FDR, SNS and group-FDR have similar performance in four out of the five iterations. As the q-value threshold increases, SNS steadily outperforms group-FDR. At a 5% q-value threshold SNS improves upon group-FDR by 3.5 (4.5%), 9.5 (5.8%), 8 (11.73%), 4 (4.7%), and 25 PSMs (17.5%).

3.4 Discussion

In this paper we focused on scenarios where scientists may only be interested in, say, a single protein, a single type of post-translational modification, a single pathway, or a single

organism in a microbial community. It was previously recognized that in such settings, variants of the standard TDC are needed to ensure proper FDR control, especially as the common approach of search-then-select fails to control the FDR.

Our analysis suggests that all-sub also fails to properly control the FDR when the relevant subset of peptides is small. Although a small relevant set of peptides is not a problem for subset-search on its own, we show that it does become so when the dataset contains a significant number of spectra that were generated by neighbor peptides.

Conceptually, the most natural source of neighbor peptides is from homologous sequences within and between proteomes. However, neighbor peptides are a more insidious problem than homology because two peptides with seemingly very different sequences can have very similar MS2 spectra. For example, although the peptide sequences in Figure 3.2 look different from each other, they have remarkably similar precursor masses and theoretical fragmentation spectra. Thus, neighbor peptides are a potential problem even in the absence of homology. In light of this observation, we feel it is too risky to recommend using subset-search hoping that the neighbors pose an insignificant problem.

This reasoning leaves us with group-FDR as the only established tool that can control the FDR in our context. That said, group-FDR (like search-then-select and all-sub) sacrifices power by searching all the spectra, including the ones generated by the relevant subset of peptides, against a database that includes a large number of irrelevant peptides. This loss of power was the motivation for introducing subset-search to begin with, and so our novel SNS manages to avoid most of this power loss while addressing the neighbor problem.

The detrimental effect of neighbor peptides on FDR estimation was previously discussed in the context of modifications.[46] Here we demonstrated and addressed this effect in the context of subset search. We believe that the existence of neighbor peptides is also likely to have a large effect on the results of multi-pass searches.[32,40,47,100] In these procedures, spectra are searched against a series of peptide databases. Such a setup can be problematic because peptides present in different databases can be neighbors of each other. As a result, spectra could incorrectly match well with a peptide found in the initial database search when

in fact the best match would be found in subsequent searches. Future work needs to be done to quantify and address the neighbor peptides effect in the context of multi-pass search.

In practical terms, although SNS was designed specifically to address the problem of neighbors in a subset search, there is no reason that it cannot be used more widely. Indeed, we believe that whenever one can compile a list of irrelevant neighbors, SNS can and should be used instead of TDC irrespective of the size of the relevant set. The only word of caution here is that, consistent with our previous recommendations,[37] our analysis here relies on a calibrated score such as combined p-value.

Finally, our goal in this paper is not to try to change the standard the field has settled on over the last 15 years or so, namely, that of controlling the FDR using TDC. Rather, we aim to show that some previously published variants of TDC fail to control the FDR in this setup of searching a subset, and to offer alternatives that do. It is important to keep in mind that, especially when dealing with a small set of discoveries, controlling the FDR, as SNS does, does not imply control of the FDP: the FDR is the expected value of the FDP so for any given sample the latter can be significantly higher than the selected FDR-controlling threshold (e.g.,[38]). Ideally we would like to control the FDP and there are some promising new developments in the theory and practice of controlling the FDP.[36,56] However, these new techniques are yet to reach the wider scientific community and we expect their introduction would face significant headwind because controlling the FDP inevitably leads to smaller sets of discoveries compared with controlling the FDR.

Chapter 4

MS1CONNECT: A MASS SPECTROMETRY RUN SIMILARITY MEASURE

4.1 Introduction

Properly analyzing proteomics data requires knowledge regarding sample metadata. Information such as species identity, instrument type, acquisition style, and sample preparation method are needed in order to correctly analyze proteomics data. For example, knowing the species identity of a sample is necessary to conduct a database search. Without this information, spectra have to be searched against a large database such as the NCBI non-redundant protein database. Searching against a very large database is not ideal because it requires a large amount of computational resources, and statistical power is greatly reduced due to a high multiple hypothesis testing burden. Another piece of metadata that is required for analyzing proteomics data is the presence of static modifications such as TMT, SILAC, or cysteine alkylation. If, for example, a sample was labeled using TMT but the researcher performing the database search was unaware of this, then the resulting set of peptide-spectrum matches (PSMs) would be non-nonsensical. Finally, metadata regarding computational analyses is also required in order to produce reproducible analyses.

Even though metadata is important, it is often can be missing or incorrect. The specific reasons why metadata would be missing vary. For example, if data from one study is repurposed for a subsequent study the original study may not have recorded the metadata relevant to the subsequent study. In addition, metadata fields may be lost when transitioning projects between personnel. Finally, researchers may incorrectly or incompletely record metadata.

When presented with a mass spectrometry run with missing metadata, researchers have to

infer the missing metadata. Some metadata can be easily inferred by examining file formats. For example, you can determine the vendor that manufactured the instrument by examining the file extension since each vendor has their own file format (*e.g.* "raw" for Thermo Scientific instruments). In addition, file formats such as .mzML[57] have metadata fields embedded within them. However, information such as species identity and presence of modifications require specialized tools. For example, tools such as Param-medic[58,59] and Preview[42] help users choose the best precursor and fragment tolerances as well as suggest modifications that should be considered. With species identification, there is a wealth of literature dedicated to identifying the species composition of an unknown sample.[7,8,35,64,65,79,97] One common strategy is to look for peptides that are a unique to a taxonomic group.[7,35,64,65]

We instead propose to infer metadata by first calculating the similarity between a query run and a repository of runs. As part of this, we have developed a new method, MS1Connect, that only uses MS1 information to calculate the similarity between a pair of runs. Repository runs with similar metalabels as the query run should be highly similar to the query run. After retrieving the k most similar repository runs, we use the metadata labels from those runs to predict the metalabel of the query run.

While in this work we focus on the application of metadata inference, we note that a general measure of similarity between proteomics runs is useful for other applications. For example, a similarity measure of proteomics runs can be used for classification, clustering, and embedding. In addition, it allows data from multiple distinct experiments to be analyzed jointly.

Due to the varied ways that mass spectrometry-based proteomics data is collected and analyzed, it is difficult to measure proteomics run similarity. Previous methods have measured the similarity by directly comparing the set of MS2 spectra that result from one run against the MS2 spectra from another run.[72,77,92] For example, one method used the distribution of spectra dot products between the two sets of MS2 spectra to determine similarity.[72] These types of methods have been successfully used to reconstruct phylogenetic trees,[72] differentiate between experimental protocols,[92] and to identify species.[77] Our method, like

these methods, does not require a database search nor the metadata required to conduct a database search. However, these previously developed methods, which use MS2 spectra, can not jointly analyze data-dependent acquisition (DDA) and data-independent acquisition (DIA) data. Our method, on the other hand, can jointly analyze DDA and DIA data since it only considers MS1 data.

So far MS1 data has not been used to measure the overall similarity between a pair of proteomics runs. However MS1 data has been previously used for microbial organism identification. Specifically, there is a long history of using MALDI-TOF MS to identify organisms (reviewed in[29] and[91]). More recently there has been some exploration of using MS1 information in LC-MS/MS runs.[49] In addition to organism identification, methods have been developed to align MS1 features maps of a pair of proteomics runs.[10,78] Scoring this alignment could be a measure of run similarity. However, these methods do not report a score of the alignment.

MS1Connect only considers MS1 data and frames scoring the similarity between a pair of proteomics runs as a maximum bipartite matching problem subject to some constraint. A bipartite graph consists of two disjoint sets of vertices and a set of edges that span the two set of vertices. In our setting, each of the two disjoint sets of vertices each represent the set of MS1 features found in a run and edges link MS1 features, in different runs, whose m/z match within some tolerance. In a maximum bipartite matching problem, the goal is to select a set of edges in a bipartite graph that maximizes some objective value subject to some constraint. In this setting, the constraint we use requires that every MS1 feature be associated with at most one edge in the set of selected edges.

The MS1Connect objective function consists of a weighted combination of three modular terms and a fourth supermodular term. Modular and supermodular functions are both set functions. In a modular function the sum is equals to its parts. More specifically, given two sets of disjoint items X and Y and a scoring function f, $f(X) + f(Y) = f(X \cup Y)$. On the other hand, for a supermodular function, the sum is greater than its parts. In this case $f(X) + f(Y) \leq f(X \cup Y)$. There is a robust set of machine learning literature dedicated to

the theory of supermodular maximization.[4,23,33,54] However, these functions have rarely been applied to the analysis of biological data. The one known example uses a surrogate function to estimate the supermodular relationship between fragment ions in database searching.[3]

Each of the MS1Connect terms measures a different aspect of proteomics run similarity. The first modular term favors solutions, *i.e.* a particular subset of edges, with a large number of edges. The second modular term favors selection of edges with high intensity values. The third modular terms favors edges with small normalized retention time shifts. We define the normalized retention time shift as the difference in normalized retention times between the two MS1 features in an edge. Finally, the fourth supermodular term favors solutions that choose pairs of edges that are similar to each other. We say two edges are similar if they have similar normalized retention time shifts and if the normalized retention times of the MS1 features, in the same run, are similar to each other.

We shows evidence that MS1Connect can accurately measure the similarity of two proteomics runs. Specifically, we show that MS1Connect scores can be successfully used to predict the species a sample originated from. Furthermore, we show that supermodular methods generally outperform modular methods. In addition, we demonstrate that MS1Connect, which was trained to predict species, can be used for other classification tasks without any modifications. For example, we show that it can be used to predict what human tissue a sample originated from. We also find MS1Connect can distinguish between different sample preparation methods. Finally, we show that MS1Connect scores are able to recapitulate analyses that result from MS2 data.

4.2 Methods

4.2.1 Representation of a mass spectrometry run

We represent each tandem mass spectrometry run as a bag of MS1 features, where each MS1 feature nominally corresponds to a peptide feature detected in a set of precursor scans. Each MS1 feature is represented as a tuple of three values: m/z, intensity, and retention time (in

seconds). All three of these values are reported by pyOpenMS,[80] the tool we use for MS1 feature detection.

Prior to analysis, we replace two of the three values with normalized versions thereof. The retention time normalization is based on the proportion of total ion current that has been detected up to the given time, yielding a value ranging from zero to one. In addition, we scale the intensities to range from zero to one by dividing by the maximum intensity among all MS1 features in the given run. Finally, prior to normalizing the values, we select the top N most intense MS1 features to represent an entire run, where N is a hyperparameter of MS1Connect. In addition, we remove MS1 features that had a normalized retention time less than 0.05 and greater than 0.95.

To generate the input files for pyOpenMS, we use Proteowizard version 3.0[9] to convert Thermo RAW files to .mzML format.[57]

4.2.2 Representing mass spectrometry run matching as a maximum bipartite matching problem

We frame the measurement of the similarity between a pair of mass spectrometry runs as a maximum bipartite matching problem. In this approach, we aim to select a set of edges in a given bipartite graph that achieves a maximum objective value.

A bipartite graph $G = (U, V, E)$ consists of two disjoint sets of vertices, U and V, and a set of edges $E \subseteq U_{r_1} \times V_{r_1}$, where each edge e connects a vertex $u \in U$ to a vertex $v \in V$. For our specific formulation, U and V are the sets of MS1 features from two different mass spectrometry runs, r_1 and r_2, and the edges E link MS1 features between U and V (Figure 4.1). For this reason, we have one bipartite graph $G_{r_1,r_2} = (U_{r_1}, V_{r_2}, E_{r_1,r_2})$ associated with every run pair r_1, r_2, where $E_{r_1,r_2} \subseteq U_{r_1} \times V_{r_2}$. For notational simplicity, we drop the r_1, r_2 subscripts except when they are needed for run-pair disambiguation.

We include in the graph edges between all pairs of MS1 features whose m/z values match

within some tolerance δ_1:

$$E = \{(u, v) : |m(v) - m(u)| \leq \delta_1\} \subseteq U \times V,$$

where $m(u)$ is the m/z of u and $m(v)$ is the m/z of v. By connecting MS1 features with similar m/z, these edges attempt to connect the same peptide precursor.

The goal in maximum bipartite matching is to select a set of edges A that achieves a maximum score, as measured by a specified objective function S, subject to some matching constraints. Specifically, because we expect that each peptide precursor will be detected at most once per run, we require that a valid matching connects each feature in U to at most one feature in V and vice versa. More formally, for two runs r_1 and r_2, we want to choose a subset of edges $A \subseteq E$ that achieves a maximum value of an objective defined below subject to the following constraints:

$$\forall e \in A, \text{degree}(u(e)) \leq 1 \text{ and } \text{degree}(v(e)) \leq 1,$$

where $u(e)$ retrieves the relevant MS1 feature from U and $v(e)$ retrieves the corresponding MS1 feature in V. To designate these constraints, we define $\mathcal{E}_U = \{A \subseteq E : \forall e \in A, \text{degree}(u(e)) \leq 1\}$ which is the set of all subsets of edges that abide by the required degree constraint of the corresponding nodes on the U side, and we correspondingly define $\mathcal{E}_V = \{A \subseteq E : \forall e \in A, \text{degree}(v(e)) \leq 1\}$ for the V side.

4.2.3 Scoring a candidate matching

Our handcrafted score function S_{r_1,r_2} uses a maximum matching approach which maximizes an objective that consists of a weighted combination of four terms. The score for two runs r_1, r_2 is defined as follows:

$$S_{r_1,r_2} = \left(\sum_{j=1}^{4} \lambda_j M_j(E_{r_1,r_2}) \right) \times \max_{A \subseteq E_{r_1,r_2} : A \in (\mathcal{E}_{U_{r_1}} \cap \mathcal{E}_{V_{r_2}})} \sum_{i=1}^{4} \lambda_i \frac{M_i(A)}{M_i(E_{r_1,r_2})}, \tag{4.1}$$

where M_1 through M_4 are terms that are defined below, λ_1 through λ_4 are convex mixture hyperparameters that weight the relative importance of each term, and E_{r_1,r_2} is the set of

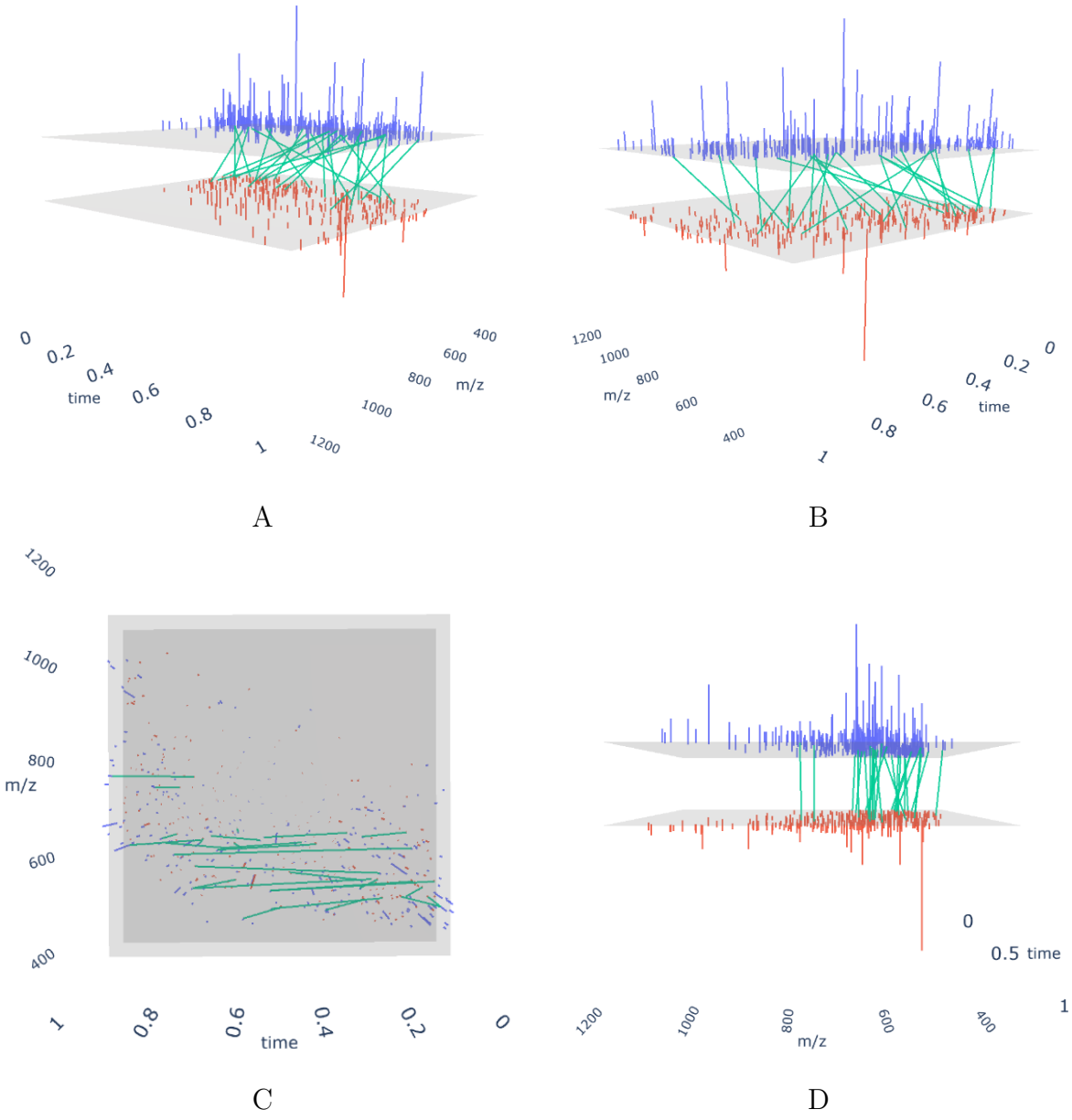


Figure 4.1: **Views of bipartite graph.** Four views of the bipartite graph formed by a pair of mass spectrometry runs. The purples lines are the MS1 features from one run (006_EC-D2O_A17.raw) and the orange lines are the MS1 features from the second run (11B_RINICHI_sTgw10.raw). The green lines represent the set of all possible edges E that link MS1 features with similar m/z. Note in D how edges are nearly vertical and only link MS1 features with similar m/z.

valid edges between runs r_1 and r_2. Each term in the inner maximization is normalized by the score for the full set E with no matching constraints. The inner maximization maximizes over all subsets of edges that abide by both degree constraints, and that is indicated via $A \in (\mathcal{E}_{U_{r_1}} \cap \mathcal{E}_{V_{r_2}})$ since if A is a member of both $\mathcal{E}_{U_{r_1}}$ and $\mathcal{E}_{V_{r_2}}$, then no node incident to any edge has more than degree one in the matching. The leading summation serves to un-normalize the objective value via a multiplicative (r_1, r_2)-dependent constant. This two-step normalization and un-normalization process is used to make the λ hyperparameters interpretable (i.e., each λ_i indicates the relative contribution of each term) and also to ensure that the scores are calibrated over multiple distinct pairs of runs. As a result of the normalization process, the four λ values must sum to one and range between zero and one, inclusive. We next describe the M_i terms each of which measure a different aspect of proteomics run similarity.

The first term M_1 counts the number of edges selected in a solution. We define the function M_1 as

$$M_1(A) = |A| \tag{4.2}$$

Thus, this term favors solutions that contain a large number of edges. Runs that are similar will share many of the same peptides. As a result, these peptides will be linked by edges; thereby generating many edges.

The second term sums the product of the two intensities associated with each edge found in a selected matching, i.e.,

$$M_2(A) = \sum_{e \in A} I(u(e))I(v(e)), \tag{4.3}$$

where $I(u)$ in the intensity of feature u and $I(v)$ in the intensity of feature v. This term favors solutions that select edges with high intensity values. The intensity of a MS1 feature resulting from a peptide is a function of the sequence and abundance of that peptide. Runs that are similar to each other should have similar peptide expression levels. By multiplying the two intensities together, we focus on the most intense peptide features.

The third term sums the negative exponent of the absolute difference of the normalized

retention time shift of all the selected edges:

$$M_3(A) = \sum_{e \in A} \exp(-\alpha |s(e)|), \tag{4.4}$$

where α is a hyperparameter that determines how quickly the exponential function decreases and $s(e)$ is the normalized retention time shift between u and v:

$$s(e) = t(v(e)) - t(u(e)),$$

where $t(u)$ and $t(v)$ are the normalized retentions time of MS1 features u and v, respectively. This term favors edges that have small normalized retention time shifts. Edges with small normalized retention time shifts are better because they are more likely to link the same peptide feature. This is because we expect peptides to elute off the column in approximately the same rank order.

Finally, the fourth term scores pairs of edges rather than a single edge. This term favors solutions that contain many similar pairs of edges. We say a pair of edges are similar to each other when they have similar normalized retention time shifts and the normalized retention times of the MS1 features, in the same run, are similar to each other. We define the term as

$$M_4(A) = \sum_{e_1, e_2 \in A} \exp(-\beta |s(e_1) - s(e_2)|) \exp(-\gamma |t(v(e_1)) - t(v(e_2))|), \tag{4.5}$$

where β and γ are hyperparameters.The first exponent is large when the normalized retention time shift of the two edges are similar. On the other hand, the second exponent is large when pairs of MS1 features in the same run have similar normalized retention times. Altogether, this term scores edge pair similarity while allowing for systematic retention time shifts between two different mass spectrometry runs. Systemic time shifts between two runs are highly likely as the liquid chromatography process has not standardized. Note that, for efficiency, we set M_4 to zero when $|s(e_1) - s(e_2)| > 0.01$ or when $|t(v(e_1)) - t(v(e_2))| > 0.01$.

4.2.4 Computing the Score via Selecting the Best Matching

In order to choose the best matching, and compute the score for a pair of runs in Equation 4.1, we must choose a feasible subset of edges that maximizes our objective function, that is we

must compute $\max_{A \subseteq E : A \in (\mathcal{E}_U \cap \mathcal{E}_V)} M(A)$ where $M(A) = \sum_{i=1}^{4} \lambda_i \frac{M_i(A)}{M_i(E)}$. The first thing to note is that $M(A)$ is the summation over entries of a $|U||V| \times |U||V|$ matrix. That is, $M(A) = \sum_{e_1, e_2 \in A} m(e_1, e_2)$ where $m(e_1, e_2) = \sum_{i=1}^{4} \lambda_i m_i(e_1, e_2)$ and where $m_i(e_1, e_2)$ is the element of the corresponding sub-objective M_i defined above (that is, $M_i(A) = \sum_{e_1, e_2} m_i(e_1, e_2)$). Note that for $i \in \{1, 2, 3\}$ we have that $m_i(e_1, e_2) = 0$ whenever $e_1 \neq e_2$. We also note that $m_4(e_1, e_2) = 1$ whenever $e_1 = e_2$. Hence, we can define $m(e_1, e_2)$ as follows:

$$m(e_1, e_2) = \begin{cases} \sum_{i=1}^{3} \lambda_i \frac{m_i(e_1, e_2)}{M_i(E)}, & \text{if } e_1 = e_2 \\ \lambda_4 \frac{m_4(e_1, e_2)}{M_i(E)}, & \text{if } e_1 \neq e_2. \end{cases} \tag{4.6}$$

Ordinarily, computing such a maximization over an exponential number of subsets would be intractable. It turns out, however, that there is useful structure in the above that allows an efficient algorithm to be used to get an approximate solution. Firstly, we note that $M(A) = \sum_{e_1, e_2 \in A} m(e_1, e_2)$ is itself a function over subsets of edges, and that sums over the submatrix associated with edge set A. Given two subsets A, B, since it is the case that $m(e_1, e_2) \geq 0$ for all e_1, e_2, this function has the property that $M(A) + M(B) \leq M(A \cup B) + M(A \cap B)$, and any such function is known to be *supermodular function*. While this is a function well-known to be supermodular, the reason this function is supermodular can be easily seen by looking at Figure 4.2. We see that $M(A) = (\text{a}) + (\text{c})$, $M(B) = (\text{e}) + (\text{c})$, $M(A \cap B) = (\text{c})$, and $M(A \cup B) = (\text{a}) + (\text{b}) + (\text{c}) + (\text{d}) + (\text{e})$. Secondly, we see that $\mathcal{E}_U$ and $\mathcal{E}_V$ correspond to a set of subsets of edges that have certain properties. I.e., the properties are: (1) $\emptyset \in \mathcal{E}_U$, (2) if $B \in \mathcal{E}_U$ and $A \subseteq B$ then $A \in \mathcal{E}_U$, and (3) if $A, B \in \mathcal{E}_U$ with $|A| < |B|$, then $\exists b \in B \setminus A$ such that $A + b \in \mathcal{E}_U$. These three properties in fact are those that define a matroid.[15,71] In fact, $\mathcal{E}_U$ and $\mathcal{E}_V$ are a particular kind of matroid, called a *partition matroid*. A partition matroid is a kind of matroid which says that we partition the edges E into disjoint blocks and for A to be independent in the matroid means that A must not intersect each block by more than a certain limit. In the present case, the blocks of edges are defined by the edges incident to each of the nodes $u \in U$ and $v \in V$. Thus, the constraint $A \in (\mathcal{E}_U \cap \mathcal{E}_V$ means that for A to be feasible, it must simultaneously be a member of the independent sets of two partition

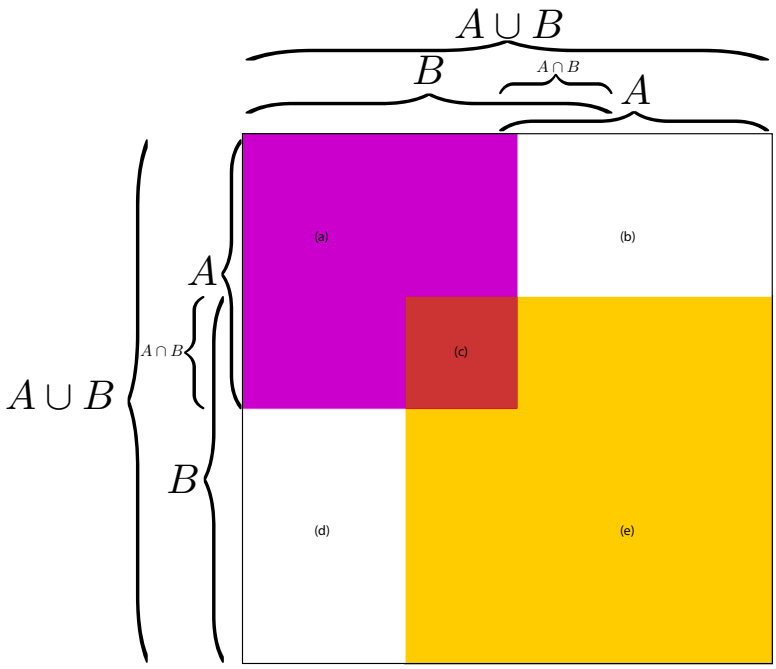


Figure 4.2: **Submatrix Relationships.** A schematic of the relationship between two sets A and B showing how the function is supermodular.

matroids $\mathcal{E}_U$ and $\mathcal{E}_V$. Hence, to produce our run-pair score, we must solve an instance of supermodular maximization subject to two matroid constraints.

Ordinarily, computing the solution to such a problem is inapproximable, as supermodular maximization subject to even simple constraints in general is very hard and even hard to approximate.[23,54] So it might seem that we're out of luck. However, in recent work[4] it was shown that if certain properties of the supermodular function were true, then an approximation algorithm is possible using the greedy procedure (defined below). Ordinarily, the greedy algorithm can do arbitrarily poorly when the function is supermodular (see[54] for a simple example). However, when the function has limited curvature, an approximation is possible. We define the supermodular curvature[4] as $\kappa^f = 1 - \min_{e \in E} f(e)/f(e \setminus E)$.[4] showed that $0 \leq \kappa^f \leq 1$ and if $\kappa^f = 1$, then indeed constrained supermodular maximization is indeed inapproximable. However, if $\kappa^f < 1$ then the greedy algorithm achieves an approximation

bound of $1-\kappa^f$, which means that $f(\tilde{S}) \geq (1-\kappa^f)f(S^*)$ where $\tilde{S}$ is the solution provided by the greedy algorithm, and S^* is the optimal solution. The greedy algorithm is fairly simple. We start with $S \leftarrow \emptyset$ and we then repeat the following step $S \leftarrow S \cup \text{argmax}_{e \in E \setminus S} f(S+e)$ until the constraints are no longer satisfied. After this is run, we multiply in the de-normalization term ($\sum_{i=1}^{4} \lambda_i M_i(E)$) as a post-processing step. This algorithm runs quickly and scalably even with very large sets $|E|$ and we therefore use this procedure for computing the score in Equation 4.1. The curvature of f is very dependent on the matrix M. It turns out that as long as the diagonal entries of M is non-zero, then $\kappa^g < 1$. In Appendix Figure B.2, we show that our best results correspond to the case where the diagonal is non-zero, and this happens as long as $\lambda_4 < 1$ in Equation (4.6), and thus $\kappa^g < 1$, which offers mathematical justification for the performance of our algorithm. Lastly, we note again that the above is described for a generic bipartite graph $G = (U, V, E)$ but in practice, we have one distinct bipartite graph for each pair r_1, r_2 of runs, as described above. Hence, the greedy algorithm is run for each run pair.

4.2.5 Baseline Similarity Measures

In order to ensure that our objective returned sensible results, we compared MS1Connect against several different baseline similarity measures. Our first baseline is the number of m/z bins in common between two sets of MS1 features. For each mass spectrometry run, we create a binary vector of m/z bins where each bin is of width δ_1. The value in the m/z bin is set to one if there is at least one MS1 feature whose m/z falls into that bin. Otherwise, it is set to zero. The score between a pair of runs is the dot product between the two vectors.

In addition, we compared MS1Connect against each individual term of the objective function. For example, calculating the M_1 baseline measure involves setting λ_1 to one while setting λ_2 – λ_4 to zero.

4.2.6 Evaluation metrics

We use three different measures to quantify the performance of a given MS1 similarity score at predicting the metadata label of a query run given a repository of runs with known metadata labels.

The first performance measure is the query average precision (QAP), defined as the average precision across all queries $q \in R$:

$$\text{QAP} = \frac{1}{|Q|} \sum_{q \in R} \sum_{k=1}^{N} P_k(S, q, R \setminus \{q\})(R_k(S, q, R \setminus \{q\}) - R_{k-1}(S, q, R \setminus \{q\})), \tag{4.7}$$

where $P_k(S, q, R)$ and $R_k(S, q, R)$ are the precision and recall, respectively, after k repository runs have been retrieved from R using query q and similarity S. Note that the query run q is not included in the ranking when computing the precision and recall.

The second measure, aggregate average precision (AAP), is similar to QAP except that the average precision is calculated once on an aggregated list of similarity scores. This aggregated list is produced by sorting together all pairs of runs, considering only the upper triangle of the run-by-run matrix.

Finally, the last measure, precision@k, is defined as the precision among the k most similar repository runs, averaged across all queries. Formally, for a similarity score S and a collection R of runs, the measure is defined as

$$\text{precision@k}(S, R, k) = \frac{1}{k|R|} \sum_{q \in R} |\{i \leq k : \ell(q) = \ell(r^{(i)}(S, q, R \setminus \{q\}))\}| \tag{4.8}$$

where $r^{(i)}(S, q, R)$ is the kth-ranked run when runs in R are ranked by similarity S with respect to query q, and where $\ell(q)$ is the metalabel associated with run q. Note that we also compute a variant of precision@k in which, prior to analysis, we remove repository runs that have the same PRIDE project and metadata label as the query run q. Removing these similar runs creates a more difficult task, forcing the similarity measure S to find similar runs in completely independent experiments.

4.2.7 Hyperparameter search

MS1Connect has nine different hyperparameters (Table 4.1) Two of them affect the generation of the bipartite graph while the remaining seven hyperparameters affect the scoring of a specific selected edge set. The first hyperparameter, N, is the maximum number of MS1 features used, in each run, to generate the bipartite graph. The next hyperparameter δ_1 is the m/z threshold used for connecting two MS1 features. Each of the four λ hyperparameters are used to weight each of the four terms in MS1Connect. Finally, α, β, and γ affect the rate in which the exponential functions decrease in M_3 and M_4.

To determine the best performing hyperparameter set, as measured by QAP, we sampled the hyperparameter space using a random grid search. In a random grid search, the range and values of each hyperparameter is predetermined. Then, for each instance of a hyperparameter set, random values are selected from the grid.

Our hyperparameter search for MS1Connect consisted of three different phases (Table 4.1). For all of our phases, we searched the hyperparameter space using a random grid search. The first phase of the hyperparameter search was an initial scan to determine the correct range of values for our hyperparameters. We allowed the number of MS1 features used in the bipartite graph generation to range from 250 to 4000 with two-fold change increments. The m/z tolerance, δ_1, was allowed to take one of nine values between 0.005 and 0.08 Da, inclusive, with a $\sqrt{2}$ fold change increments. All four λ hyperparameters were allowed to range between zero and one, inclusive, with increments of 0.01. Finally, the hyperparameters α, β, γ ranged from 1×10^{-6} to 1.0, with 10-fold change increments.

After the initial scan, we changed the range of two hyperparameters. For the number of MS1 feature hyperparameter, we removed 250 and 500 MS1 peaks while adding 8000 peaks. In addition, for δ_1, we dropped the values of 0.014, 0.02, 0.028, 0.04, 0.057, 0.08 and added the values of 0.0025 and 0.0035 Da. Finally, for the third phase of the hyperparameter search, we fixed the number of MS1 features to be 4000 and δ_1 to be 0.0035 Da. A total of 7,870 hyperparameter sets were sampled out of a total of 3,939,276 possible hyperparameter sets.

hyper-parameter	phase 1		phase 2		phase 3	
	range	step	range	step	range	step
# MS1 features	250 – 4000	2*	1000 – 8000	2*	4000	NA
m/z tolerance	0.005 – 0.08	$\sqrt{2}$*	0.0025 – 0.01	$\sqrt{2}$*	0.0035	NA
α	1×10^{-6} – 1.0	10*	1×10^{-6} – 1.0	10*	1×10^{-6} – 1.0	10*
β	1×10^{-6} – 1.0	10*	1×10^{-6} – 1.0	10*	1×10^{-6} – 1.0	10*
γ	1×10^{-6} – 1.0	10*	1×10^{-6} – 1.0	10*	1×10^{-6} – 1.0	10*
λ_1	0.0 – 1.0	0.1	0.0 – 1.0	0.1	0.0 – 1.0	0.1
λ_2	0.0 – 1.0	0.1	0.0 – 1.0	0.1	0.0 – 1.0	0.1
λ_3	0.0 – 1.0	0.1	0.0 – 1.0	0.1	0.0 – 1.0	0.1
λ_4	0.0 – 1.0	0.1	0.0 – 1.0	0.1	0.0 – 1.0	0.1

Table 4.1: **Hyperparameter search grid.** Table of the hyperparameter grid that was searched during each of the three phases of the hyperparameter search. The step column refers to the increment for each hyperparameter. We evaluated 7,870 out of the 3,939,276 possible hyperparameter sets. *Increments are fold changes instead of linear.

In addition to a hyperparameter search for MS1 connect, we conducted an exhaustive hyperparameter search for each of the five baselines to find the best performance of each baseline. The hyperparameter grid for these searches consists of all relevant possible values found in the previously discussed phases. The best set of hyperparameters for each method can be found in Appendix Table B.2.

4.2.8 Data

Species training dataset. A total of 166 RAW files from 33 PRIDE IDs were downloaded from PRIDE[75] in November of 2020. These 166 runs each originated from a sample that contains one of nine different species (Appendix Figure B.1A). For this dataset we used data from up to five PRIDE IDs per species, except human had six PRIDE IDs, and downloaded up to five RAW files per PRIDE ID. The mass spectrometry runs were collected using various Thermo Scientific instrumentation (San Jose, CA). We did not include any off-line fractionated samples or labeled samples (*e.g.* TMT or ITRAQ). All samples were digested using trypsin.

Species test dataset. An additional 130 RAW files from 23 PRIDE IDs were downloaded from PRIDE in December of 2020 for compilation into a species test dataset. Each run consists of data from one of nine different species (Appendix Figure B.1B). Like the training dataset, we used five RAW files per PRIDE ID; however, we only used data from three different PRIDE IDs per species. One exception is that we only collected *Plasmodium falciparum* data from two different PRIDE IDs. The other exception is that all 15 runs of the *Yersinia pestis* data came from a single PRIDE ID. However, the data from that project itself is an aggregation of several different experiments. None of the PRIDE IDs nor runs in this dataset overlap with the data in species training dataset.

Tissue test dataset. The tissue dataset came from a project that studied the variability of protein expression in human tissues and created a machine learning model that successfully predicted what tissue a sample originated from.[48] This dataset contains 252 RAW files from nine different human tissues (Appendix Figure B.1C). We downloaded this data from PRIDE

(PXD010271) in October of 2018. Briefly, this dataset contains a mixture of healthy and diseased tissue, and all the data was collected by Pacific Northwest National Laboratory on either a Thermo Velos Orbitrap or Thermo Q-Exactve instrument (San Jose, CA). None of the sample were off-line fractionated or labeled.

Bacterial inactivation dataset. This dataset came from a project that studied the effects of bacterial inactivation on protein abundances.[53] Three replicate cultures of *Yersinia pestis* KIMD27 and *Escherichia coli* ATCC 15597 were grown. Each of these replicate cultures was subjected to one of three different inactivation methods: irradiation, ethanol, and autoclaving. The culture subjected to irradiation was exposed to a Colbalt-60 source for 24 h at 0.47 kGy/h for a total exposure of 11.3 kGy. For the autoclave method, samples were exposed for 20 min at 30 psi [206.8 kPa] and 121 °C in a Getinge autoclave. Finally, for the ethanol treatment, samples were incubated in 40% ethanol for 30 min at room temperature. Each sample was injected into a Thermo Scientific LTQ Orbitrap XL mass spectrometer (San Jose, CA) three times for a total of 72 runs.

4.2.9 Database search

A database search was conducted using the Crux version 3.2[62,73] with a concatenated *Y. pestis* KIM10+ and an *E. coli* protein database downloaded from Uniprot[89] in January 2020. The protein database was digested using the tide-index tool, and the search was performed by the tide-search tool. All parameters were set to their default values except that the fragment ion bin width ("–mz-bin-width") was set to 1.0005079.

4.3 Results

4.3.1 MS1Connect can be used for species prediction

To validate whether MS1Connect scores are successfully able to measure the similarity of a pair of runs, we investigated whether our method can predict the species label of a proteomics run.

method	species training set		species test set	
	QAP	AAP	QAP	AAP
MS1Connect	**0.8099**	**0.7425**	0.8504	0.8056
M_1 only	0.7896	0.7075	0.8256	0.7880
M_2 only	0.7153	0.6031	0.6598	0.5778
M_3 only	0.7912	0.7079	0.8264	0.7881
M_4 only	0.8064	0.7396	**0.8522**	**0.8076**
# m/z bins in common	0.7847	0.6978	0.8211	0.7855

Table 4.2: **QAP and AAP.** A table of the per-query average precision (QAP) and the aggregate average precision (AAP) for the species train and test datasets. Bolded values denote the best performance for each column. MS1Connect performs the best on the species training set while M_4 performs the best on the species test set.

First, we compared the performance of MS1Connect against our baseline methods. For the species training data, we found that MS1Connect performed the best with a performance of 0.8099 and 0.7425 for QAP and AAP, respectively (Table 4.2). The M_4 baseline performed slightly worse than MS1Connect. On the other hand, in the species test set, we found M_4 had the best performance (0.8522 and 08076 for QAP and AAP, respectively) while MS1Connect had slightly worse performance (0.8504 and 08056 for QAP and AAP, respectively). While MS1Connect did not have the best score, we note that both supermodular methods outperform all modular methods, showing the utility of supermodularity.

Examining the remaining modular baselines, we found that M_2 baseline performs the worst with a QAP of 0.7153 and an AAP of 0.6031. This is likely due to M_2 not considering retention time. As a result, this method will select edges whose component MS1 features have very different retention times. This relative poor performance indicates the importance of retention time in measuring similarity. While the M_2 baseline had the overall worst performance, it still outperformed random chance. This suggests that the bipartite graph

generation process itself is useful for measuring the similarity between a pair of proteomics runs.

Comparing the performance of QAP to AAP, we found the expected result that QAP is always higher than AAP. This difference results from AAP being a more difficult task. In order to perform well with AAP, the score must be statistically calibrated with respect to queries. The lower performance of AAP implies our scores are not perfectly calibrated. Even though the AAP is lower than the QAP, these two metrics are highly correlated with a Pearson correlation of 0.9829 (Figure 4.4A).

Next, we examined the MS1Connect scores among the runs in the species training data to understand whether the scores make qualitative sense. We expect that runs generated by the same experiment should be highly similar to each other. Beyond that, we also expect runs from the same species to also be similar with each other.

Using the hyperparameter set that yielded the best performance on the training data, we visualized the MS1Connect scores between all pairwise runs in the species training dataset as a heatmap (Figure 4.3). In this heatmap the rows and columns are ordered by expected similarity. In general, the resulting heatmap matches our expectation. For example, there is a strong diagonal component in the heatmap. These small boxes of high scores correspond to runs that were generated by the same experiment. In addition, there is larger structure within the heatmap showing that runs generated from the same species tend to have high scores. For example, all the runs generated by *S. aureus* and *C. albicans* have high scores with each other and do not have high scores with runs from other species. On the other hand, there are examples of runs whose similarity profile does not match our expectation. One prominent example is a set of five *E. coli* runs from the same experiment that only have high MS1Connect scores when comparing the same run to itself.

One unexpected finding was that MS1Connect may be sensitive enough to detect inter-species relationships. Considering the human and mouse runs, we saw that MS1Connect indicates these two species have some degree of similarity. This fits with our phylogenetic understanding of these two species as human and mice are both mammals. These two species

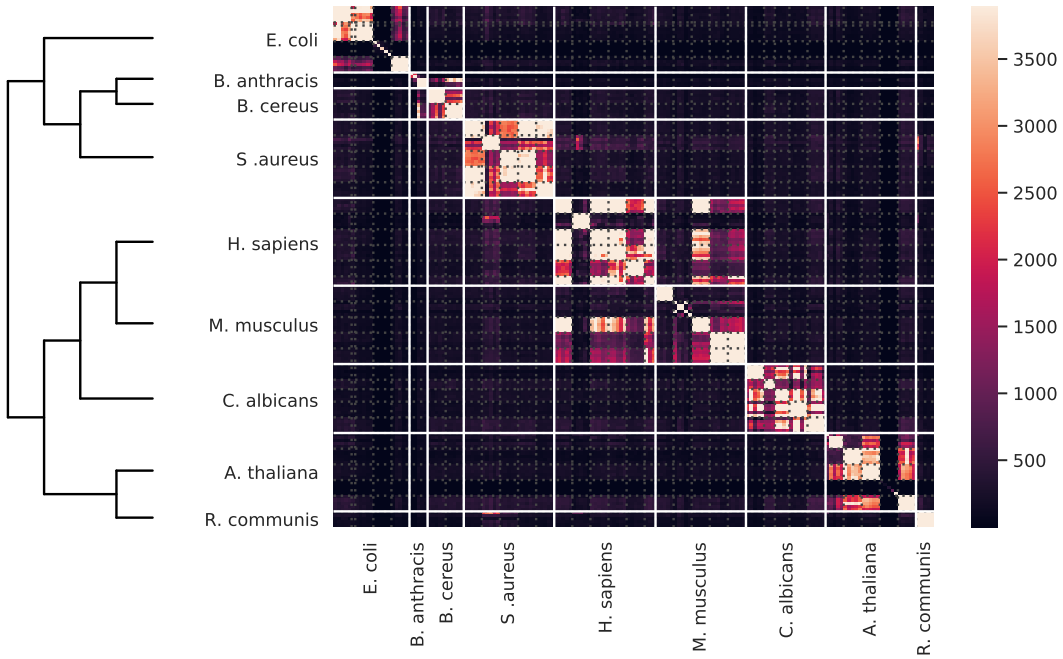
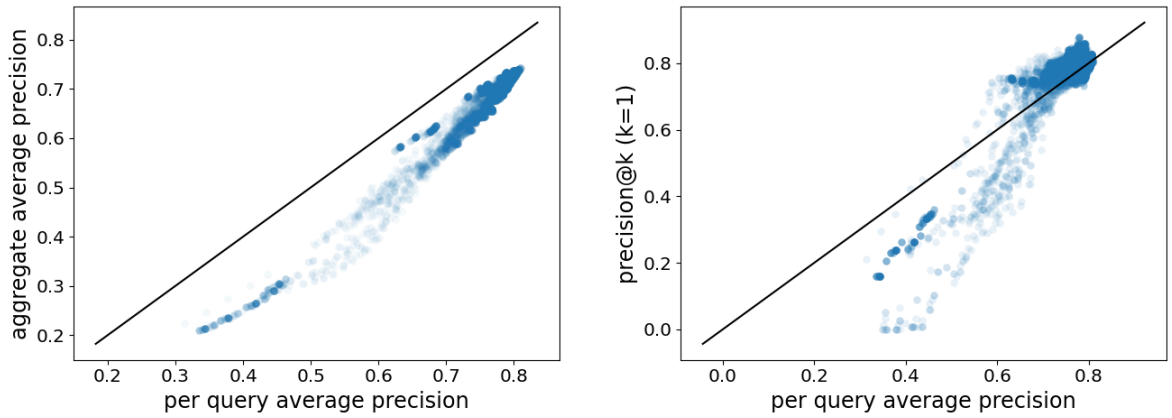


Figure 4.3: **Heatmap of MS1Connect similarities for species training data.** A heatmap and dendrogram of MS1Connect scores showing the structure of our species training dataset. Each cell is colored by the MS1Connect score between a pair of runs. The solid white lines denote the border between different species while the dotted gray lines delineate the border between different experiments (PRIDE ID). The denodrogram shown is the phylogenetic tree of the 9 species found in the training dataset.



A B

Figure 4.4: **Correlation of metrics.** A) Scatter plot of the per-query average precision and aggregate average precision of the species training dataset over 7,870 different hyper-parameterizations. These metrics are highly correlated with a Pearson correlation of 0.9829 The solid black line shows the $y = x$ line. B) Scatter plot of of the same points found in A but plotting QAP against precision@k where $k = 1$. The Pearson correlation of these two metrics is 0.9231.

are highly similar to each other in the context of our dataset, which also includes bacteria, fungi, and plants. Another example is that we saw *B. anthracis* and *B. cereus* runs are similar to each other, which is expected since these two species in the same genus. In the species test data, we saw similar results with the two gram-negative bacterial species: *E. coli* and *S. enterica* (Appendix Figure B.5).

In addition to QAP and AAP, we investigated the performance of our method with respect to precision@k. The difference between these metrics is that the average precision measures the performance of the entire ranking while precision@k focuses on the beginning of the ranking. We calculated the precision@k for both the species training and test set with k ranging from one to four. Note that for each query, prior to analysis, repository runs that had the same metadata label and PRIDE ID were removed, which results in a more difficult

method	training data				test data			
	k=1	2	3	4	k=1	2	3	4
MS1Connect	0.808	0.846	0.846	**0.846**	0.739	0.770	0.774	**0.789**
M_1 only	**0.859**	0.849	0.840	0.840	**0.748**	0.752	0.765	0.763
M_2 only	0.795	0.811	0.795	0.782	0.426	0.448	0.440	0.439
M_3 only	**0.859**	0.849	0.844	0.843	**0.748**	0.752	0.762	0.761
M_4 only	0.822	0.841	0.843	0.843	0.739	**0.783**	**0.777**	0.783
random	0.116	0.014	0.002	0.0002	0.077	0.005	0.0003	2×10^{-5}
# m/z bins in common	0.841	**0.856**	**0.847**	0.845	0.747	0.761	0.764	0.751

Table 4.3: **Precision@k.** A table of the precision@k values for the species training and test data. Bolded values denote the best performance for each column. For each query, prior to analysis, repository runs that had the same metadata label and PRIDE ID were removed.

task.

We discovered that no method consistently outperformed any other method. In the training data, the M_1 and M_3 baselines had the best performance when $k = 1$ with a performance of 0.8590. On the other hand, when $k = 2$ and $k = 3$, the # of m/z bins in common baseline had the best performance with a score of 0.8558 and 0.8466, respectively. Finally, MS1Connect outperforms all other methods for $k = 4$. In the test dataset, the same trends occur in that M_1 and M_3 had the best performance at $k = 1$ (0.7478) and MS1Connect had the best performance at $k = 4$ (0.7891). However, when $k = 2$ or $k = 3$, the M_4 baseline had the best performance (0.7826 and 0.7768) instead of the # of m/z bins in common baseline.

While MS1Connect did not outperform the other methods using this performance metric, we note that we did not train our hyperparameters using precision@k. Therefore, it is unsurprising MS1Connect did not outperform the baselines. However, in spite of this,

MS1Connect was still competitive with the baseline methods. In addition, it greatly outperformed random chance. Furthermore, the performance of MS1Connect increased as k increased. This performance increase is impressive because the task becomes increasingly difficult as k increases. For example, when $k = 4$, the metalabels of all four of the retrieved runs must match the query label in order to receive good performance. This increase in performance is likely because our hyperparameters were trained on the per-query average precision, which measures the entire ranking instead of the beginning of the ranking.

Finally, the performance of all the methods improved dramatically when repository runs with the same label and PRIDE ID as the query were not removed before analysis (Appendix Table B.3). While this result in not surprising we since expect runs from the same experiment to be highly similar to each other, it nevertheless provides additional evidence that our method is able to measure the similarity between a pair of proteomics runs.

4.3.2 Edge similarity term is critical for performance

Since MS1Connect consists of a weighted combination of four different terms we set out to learn the relative importance of each term. First, we performed a hyperparameter search on the species training data, using QAP as the performance metric, to find the set of hyperparameters that yielded the best performance, following the procedure outlined in Section 4.2.7. Then, we searched for trends in the top 1000 scoring hyperparameter sets.

We discovered that a majority of high-scoring hyperparameter sets had the number of MS1 features set to 4000 and the m/z tolerance set to 0.0035 Da (Appendix Figure B.3). Out of the 1000 top scoring hyperparameter sets, 708 of them used 4000 MS1 features and 625 of them set δ_1 to 0.0035 Da.

Based off this finding, we fixed the number of MS1 features to 4000 and fixed the m/z tolerance to 0.0035 Da. Then we examined the performance as a function of the remaining hyperparameters. We found λ_4 to be strongly correlated with performance (Figure 4.5). As λ_4 increased from 0.0 to 0.9, the performance steadily increased. On the other hand, performance decreased as λ_4 increased from 0.9 to 1.0. Functionally, this suggests that the edge

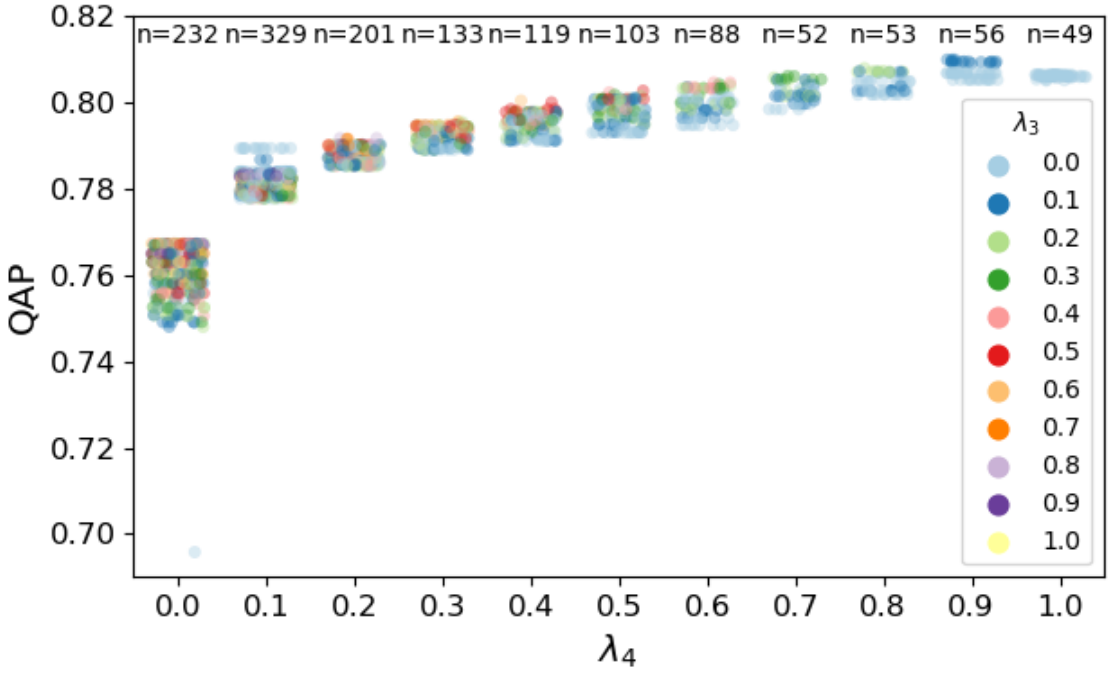


Figure 4.5: **Edge similarity term important for performance.** A strip plot of the per-query average precision (QAP) split by the possible values of λ_4 on the species training dataset when the number of MS1 features is fixed to 4000 and the m/z tolerance is set to 0.0035 Da. Each point is the performance of a specific set of hyperparameters. The text at the top indicates the number of points plotted for each possible λ_4 value. The best performance occurs when $\lambda_4 = 0.9$ and $\lambda_3 = 0.1$

similarity term M_4 is the most important term for obtaining good performance. Specifically, the best performance occurred when $\lambda_4 = 0.9$ and $\lambda_3 = 0.1$. For the remaining hyperparameters, there was no discernible trend between their values and performance (Appendix B.4). However, we found performance increased when $\lambda_4 \geq 0$ compared to when $\lambda_4 = 0$.

The major difference between M_4 and the other terms is that M_4 is a supermodular function while the remaining terms are modular functions. The large weight on λ_4 suggests that using a supermodular function to measure mass spectrometry run similarity is more

effective than using modular functions. M_4 is a supermodular function in that, instead of considering a single edge at a time, it considers a pairs of edges and asks if a pair of edges are similar. If a pair of edges are similar, then the overall score is boosted. We speculate that considering edge similarities improves performance because it reduces the contribution of spurious edges to the overall score. If a single edge is unlike any of its nearby edges, it is less likely to correctly link the same peptide feature in two different runs. On the other hand, if an edge is similar to nearby edges, it gives us additional evidence that the edge is correctly linking two MS1 features.

4.3.3 MS1Connect can differentiate between human tissues

After determining that MS1Connect can predict the species label of a sample, we investigated whether MS1Connect can be generalized to other use cases. Since MS1Connect scores are a measure of proteomic run similarity, we hypothesized these scores are likely able to detect differences from factors other than species. This is true even though the hyperparameter set that was selected was optimized for species prediction. To this end, we investigated the ability of our method to differentiate between human tissues.

We reanalyzed data from a project that created a machine learning classifier to predict what human tissue a sample originated from.[48] The input features for their method consisted of quantitative peptide measurements. In contrast to their method, our method does not require a database search and only uses MS1 data.

A heatmap of the MS1Connect scores shows our method can differentiate between runs from different human tissues (Figure 4.6). In general, runs that originate from the same human tissue all have high MS1Connect scores. For example, the monocyte and liver samples each appear as one large block. However, there are examples where there is block structure within a set of tissue samples. For example, two distinct blocks are found in the blood serum samples. This smaller structure is likely due to differences between experiments since the data from this study is itself a meta-analysis of several projects. However, we are unable to verify this since the original project did not track this information.

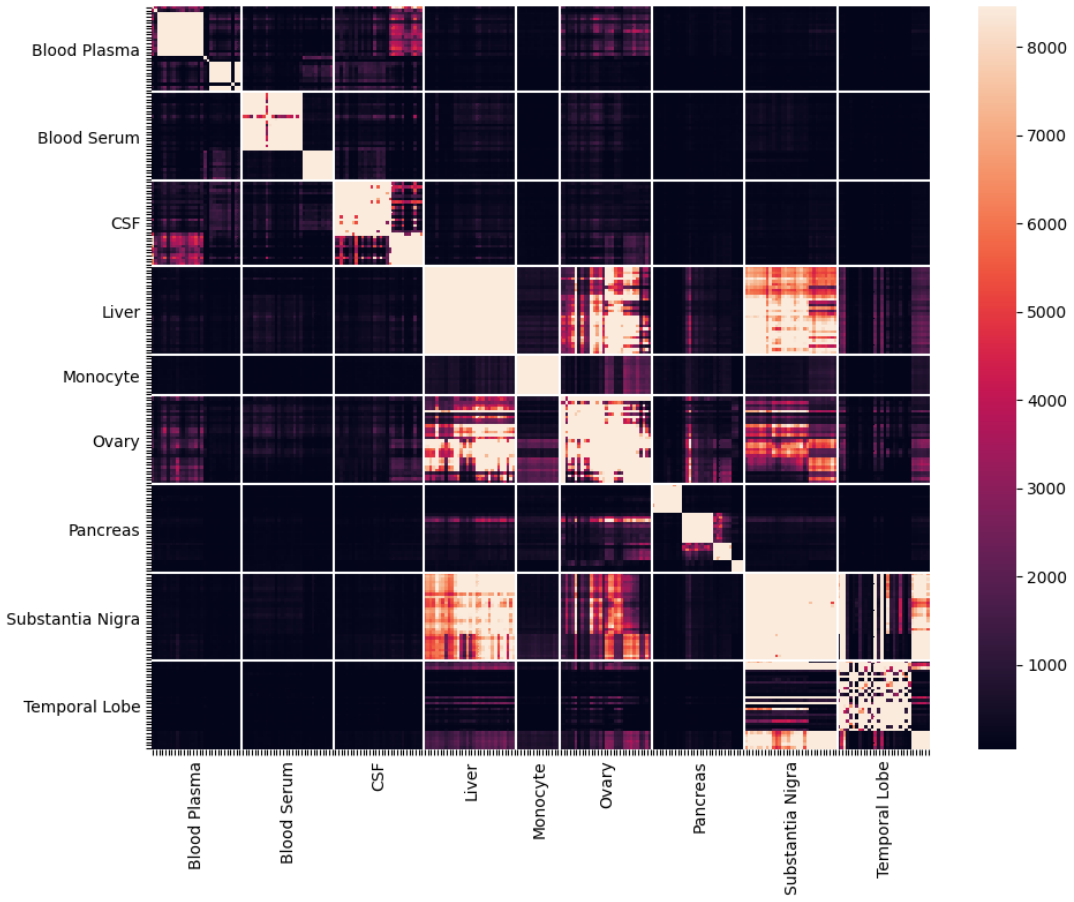


Figure 4.6: **Heatmap of MS1Connect scores for tissue dataset.** A heatmap of the MS1Connect scores for the human tissue dataset. The solid white lines denote the border between different human tissues. In general, samples that originated from the same tissue have high MS1Connect scores.

In addition to the similarity between runs generated by the same tissue, MS1Connect shows similarity across tissues. For example, runs that originated from the temporal lobe and the substantia nigra, which are both parts of the brain, have high scores. On the other hand, it is unclear why liver samples would be similar to the ovary or the substantia nigra. Additional work needs to be done to determine if these high scores are reflective of biology. Finally, MS1Connect did not detect the expected similarity between blood plasma and blood serum samples.

Examining our three performance metrics, we found no method consistently outperformed any other method. For QAP, the best method was the # of m/z in common with a score of 0.7801; on the other hand, the best method for AAP was the M_4 baseline with a score of 0.7218 (Table 4.4). When considering the precision@k results, we saw that multiple methods had the best score for different values of k (Table 4.5). The number of m/z bins in common method had the best performance when k was two, three or four. The M_4 baseline had the best performance when k was one or two. Finally, MS1Connect tied with the two previous methods for best performance when k was two. One thing to note is that all the considered methods performed well on the tissue classification task with the lowest precision@k score being 0.9821. Comparing the lowest score to the scores from random chance illustrates that these methods are working. Overall, we find our method, which was optimized for species prediction, can be reused for human tissue classification without any modifications.

4.3.4 MS1Connect can differentiate between bacterial inactivation methods

In addition to differentiating between sample types, we investigated whether MS1Connect scores can differentiate between runs that underwent different sample handling methods. We reanalyzed data from a project that studied the effect of bacterial inactivation on protein abundances.[53] In this study, aliquots of *Y. pestis* and *E. coli* were subjected to four different inactivation methods: autoclaving, irradiation, ethanol treatment, and no treatment. By investigating the profile of protein abundances, this study showed samples that were inactivated by the same method could be distinguished from samples inactivated by different

method	tissue test set	
	QAP	AAP
MS1Connect	0.7747	0.7190
M_1 only	0.7679	0.6999
M_2 only	0.7040	0.5493
M_3 only	0.7686	0.7001
M_4 only	0.7777	**0.7218**
# m/z bins in common	**0.7801**	0.7083

Table 4.4: **QAP and AAP.** A table of the per-query average precision (QAP) and the aggregate average precision (AAP) for the tissue datasets. Bolded values denote the best performance for each column. MS1Connect performs the best on the species training set while M_4 performs the best on the species tissue set.

method	k=1	2	3	4
MS1Connect	0.9881	**0.9881**	0.9868	0.9841
M_1 only	0.9881	0.9861	0.9856	0.9828
M_2 only	0.9643	0.9623	0.9563	0.9504
M_3 only	0.9881	0.9861	0.9854	0.9821
M_4 only	**0.9921**	**0.9881**	0.9868	0.9841
random chance	0.1111	0.0122	0.0013	0.0001
# m/z bins in common	0.9881	**0.9881**	**0.9881**	**0.9881**

Table 4.5: **Table of precision at k values.** A table of the precision@k values for the tissue test data. Bolded values denote the best performance for each column.

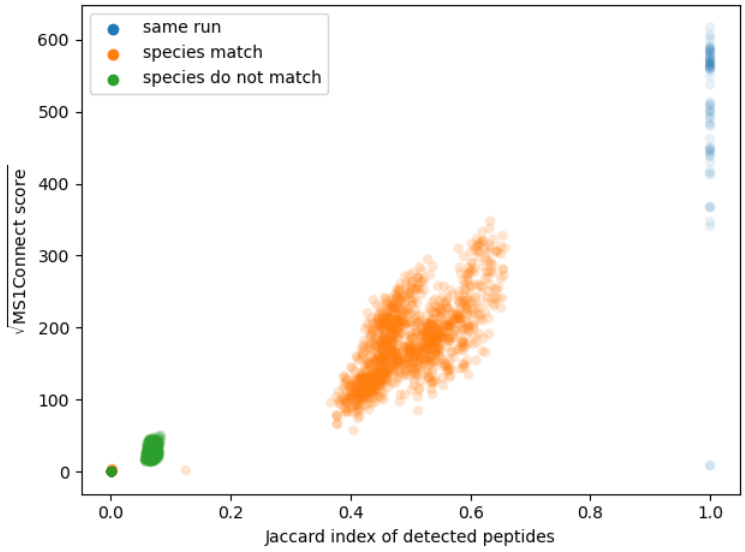


Figure 4.7: **Scatterplot of the MS1Connect scores against Jaccard index.** A scatterplot of the MS1Connect scores against the Jaccard index of the detected peptides, at 1% FDR, for the bacterial inactivation dataset. Overall, these two scores are highly correlated with each other with a Spearman rank correlation of 0.88. This suggests that MS1Connect scores are able to replicate results from proteomics analysis without conducting a database search.

methods.

Using MS1Connect scores, we were able to loosely recreate the finding from this previous study without having to conduct a database search (Figure 4.7). Specifically, the MS1Connect similarity scores were highly correlated with the Jaccard similarity of the 1% FDR controlled set of detected peptides with a Spearman's rank correlation of 0.88. In addition, a heatmap of the MS1Connect scores show a clear distinction between runs from the two different species (Figure 4.8). Looking specifically at the *E. coli* runs, we saw that runs subjected to irradiation, ethanol treatment, or no treatment were quite similar to each. This finding is replicated by the heatmap generated from the database search (Appendix B.6). On the other hand, the MS1Connect scores of the *Y. pestis* runs did not as closely mimic the

results from the database search. The database search suggested that the four inactivation methods are distinct from each other. While this phenomenon is present in the MS1Connect data, the distinction is not very apparent.

In addition to the overall structure that result from differences in species and sample handling, MS1Connect scores can detect additional fine grain structure related to injection batches. Within the *E. coli* runs that were inactivated by ethanol, there are two blocks. The first block corresponds to the first technical replicate injection of the three samples while the second block corresponds to the second technical replicate injection. This suggest that MS1Connect scores may be able to detect differences that result from technical artifacts.

A striking feature in the two heatmaps is the presence of two dark streaks (Figure 4.8 and Appendix Figure B.6). These two dark streaks suggested that two runs are not similar to any other run in the dataset. A closer investigation showed that these two runs had poor data quality. The first indication of poor data quality was that these two runs had few detectable MS1 features. The second indication was that a database search of the *Y. pestis* runs showed that these two runs accept very few PSMs (Appendix Figure B.7).

One of the differences between Figure 4.8 and and Appendix Figure B.6 is the self-similarity of the two *Y. pestis* runs with poor data quality. In the MS1Connect version, the self-similarity of these two runs are low. On the other hand, the Jaccard index is 1.0. Given the information that these two runs are of poor data quality, we argue that MS1Connect correctly shows low self-similarity since the data itself is of poor quality. Altogether, this indicates that MS1Connect has some data quality control aspects to it.

Finally, we note that the original study analyzed the two species separately while MS1Connect analyzed the species simultaneously. Our method was able to distinguish between runs of the two species as well as differences in sample handling. This indicates that MS1Connect scores are able to calculate similarities of mass spectrometry runs in the face of multiple orthogonal differences. Finally, we emphasize that we were able to recreate these findings without a database search and by only using MS1 data.

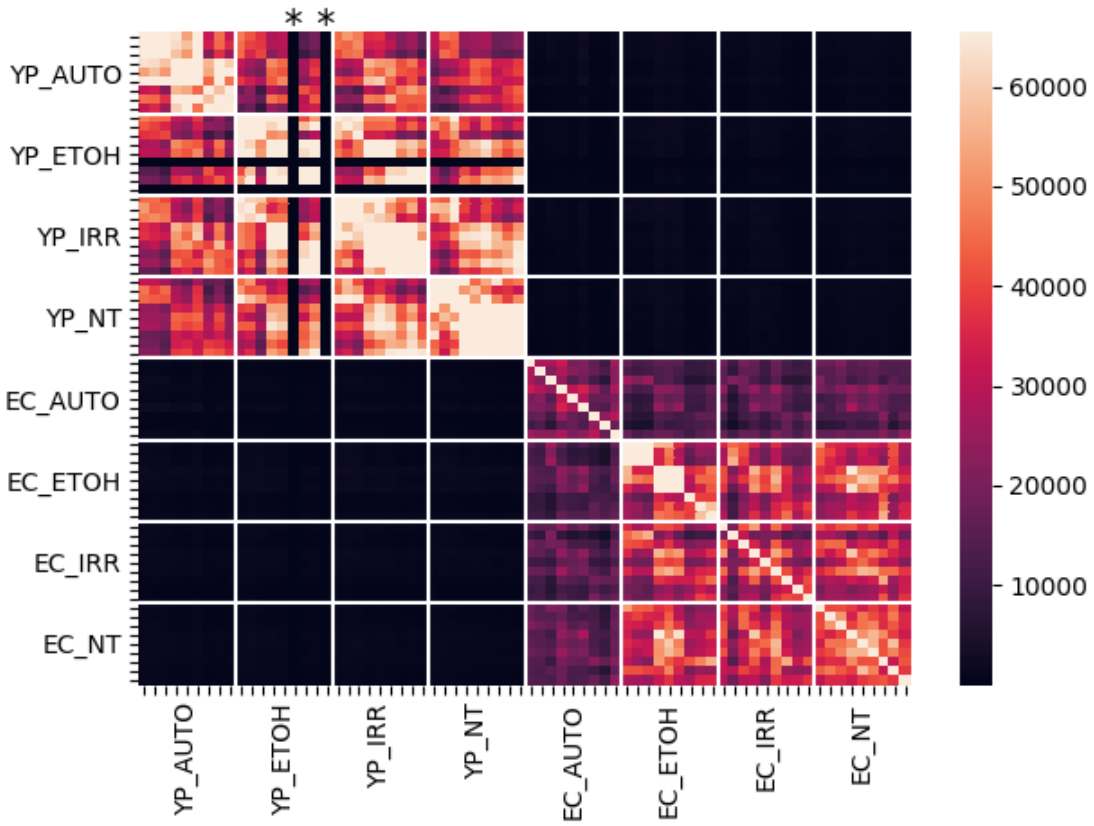


Figure 4.8: **Heatmaps of inactivation data.** A heatmap of the MS1Connect scores for all pairs of runs in the bacterial inactivation study. The solid white lines denote boundaries between groups of runs. This figure shows that MS1Connect is able to delineate between species and bacterial inactivation method. "YP" and "EC" correspond to *Y. pestis* and *E. coli*, respectively, while "AUTO", "ETOH", "IRR", and "NT" correspond to autoclave, ethanol, irradiation, and no treatment, respectively. Runs in this heatmap were first ordered by species then inactivation method. Within an inactivation method, the runs were order by technical injection blocks. * This black strip is due to two runs that have poor data quality and therefore are not similar to any other run.

4.4 Discussion

In this work, we introduce a new method, MS1Connect, that measures the similarity between pairs of proteomics runs. We show evidence that MS1Connect successfully measures proteomics run similarity by using MS1Connect scores for the classification tasks of species prediction and human tissue prediction. In addition, we show evidence that MS1Connect scores are able detect difference that result from differences in sample handling.

To achieve the best performance, the m/z tolerance hyperparameter was set to 0.0035 Da. Researchers more typically use ppm as a precursor threshold. For a 1000 Da and 500 Da peptide, 0.0035 Da equates to 35ppm and 7ppm, respectively. These two ppm numbers are well within the range of typical precursor tolerance researchers use during a database search.

Our work shows evidence that supermodular methods outperform modular methods. Here we speculate why modular terms may be insufficient for measuring similarities between mass spectrometry runs by discussing the modular M_3 term. A single edge will score highly with M_3 when the normalized retention time between the two features that make up an edge are very similar. At first glance, this seems reasonable as two peptide features with the same m/z and retention time should be the same ion. However, this fails to account for systematic differences in retention time that result from differing liquid chromatography. How liquid chromatography is conducted is not standardized across the community. Therefore, while we expect the order in which peptides elute off the column to approximately match between two runs, we can not expect such correspondence in retention time. As a result, an edge with a large retention time shift could be correct. Using supermodularity to consider pairs of edges overcomes this issue because it considers a neighborhood of edges. If a set of edges in the same region have the same retention time shift, it indicates that a systemic time shift has occurred and the retention time shifts are correct.

Since MS1Connect only uses MS1 data, it is agnostic to acquisition style. Therefore, it can calculate the similarity between runs that were collected by DIA or DDA. In fact, the data in this study contained a mixture of DDA and DIA runs. In the future, our method

could be used for jointly analyzing DDA and DIA data.

Finally, we have shown that MS1Connect scores can be used to predict metadata labels of runs. Future works needs to be conducted to determine the limits of our method. For example, none of the data used in this study was isobaric labeling. MS1Connect could be extended to include these types of sample. In addition, there is a wide variety of samples that MS1Connect has not been tested on. For example, fractionated sample or multi-species samples. Additional work could be pursued to test the applicability of MS1Connect in these scenarios.

Chapter 5

CONCLUSIONS

The field of computational proteomics has greatly advanced since the advent of the first database search engine, SEQUEST,[22] in 1994. Instead of having to annotate each spectrum at a time by hand, researchers are able to efficiently analyze thousands of spectra via these computational methods. In addition, advances in statistical methods have been instrumental for giving researchers confidence in their set of detected PSMs. Over the years, these methods have been widely adopted. For example, since the introduction of TDC to proteomics in 2011, using TDC to estimate the FDR of a set of PSMs is now standard practice across the field.[19]

Even though much has been achieved in computational proteomics, advances are still possible. Currently, the field has settled on a standard process for calculating the FDR of a set of PSMs using TDC. However, one size does not fit all, and this process is not applicable for all situations. For example, we have described in Chapter 3 a variant of the TDC process to account for the situations where researchers are only interested in a subset of the peptides in a sample. Additional variants of the TDC process are needed to better take account of the vagaries of proteomics data. For example, new variants of the TDC process are needed to allow for the presence of chimeric spectra, *i.e.*, spectra that are generated by more than one peptide. The current database search process assumes that only the best scoring peptide could have generated the observed spectrum. However, it is fairly common for an observed spectrum to be generated by multiple peptides.[28,30]

In order for proteomics data to be analyzed, a fair amount of metadata about the sample must be known. Without this information, it can be difficult to analyze proteomics data in a way to gain sensible results. However, a large majority of proteomics analysis tools assumes this information is known. In practice, this assumption is not always true. Metadata can be

lost or corrupted. In addition, in the field of forensics proteomics, very little is known about the sample. In such settings, there is a need for metadata free proteomics analysis methods.

In Chapter 4 we describe a method for inferring metadata associated with a proteomics run by using MS1Connect, a supermodular objective function to measure the similarity of pairs of runs. Additional work is needed to improve the functionality of MS1Connect. For example, one drawback of MS1Connect is its need to calculate the similarity of all pairwise runs in the dataset. This leads to a time complexity of $O(n^2)$, which is inefficient for large datasets. Using a selected subset of runs to represent the full database may allow this method to scale to larger datasets.[51,81,82,96]

Finally, MS1Connect and the more established method of *de novo* sequencing provides the foundation for metalabel free analysis of liquid chromatography-tandem mass spectrometry analysis of proteomics data. Development of these methods will allow the field to be robust to the loss of metadata. This, in turn, will ensure that all past and future data can be analyzed even if sample metadata has been lost.

BIBLIOGRAPHY

[1] G. Alves, A. Y. Ogurtsov, and Y. K. Yu. RAId_aPS: MS/MS analysis with multiple scoring functions and spectrum-specific statistics. *PLOS ONE*, 5(11):e15438, 2010.

[2] G. Alves, W. W. Wu, G. Wang, R. S. Shen, and Y. Yu. Enhancing peptide identification confidence by combining search methods. *Journal of Proteome Research*, 7(8):3102–3113, 2008.

[3] W. Bai, J. Bilmes, and W. S. Noble. Bipartite matching generalizations for peptide identification in tandem mass spectrometry. In *ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 327–336, Seattle, WA, 2016.

[4] Wenruo Bai and Jeff Bilmes. Greed is still good: Maximizing monotone submodular+supermodular (bp) functions. In *International Conference on Machine Learning (ICML)*, Stockholm, Sweden, July 2018.

[5] T. Bailey and M. Gribskov. Estimating and evaluating the statistics of gapped local-alignment scores. *J Comput Biol*, 9(3):575–593, 2002.

[6] T. L. Bailey and W. N. Grundy. Classifying proteins by family using the product of correlated p-values. In S. Istrail, P. Pevzner, and M. Waterman, editors, *Proceedings of the Third Annual International Conference on Computational Molecular Biology*, pages 10–14. ACM, April 1999.

[7] F. Boulund, R. Karlsson, L .Gonzales-Siles, A. Johnning, N. Karami, O. AL-Bayati, C. Åhrén, E. R.B. Moore, and E. Kristiansson. Typing and characterization of bacteria using bottom-up tandem mass spectrometry proteomics. *Molecular & Cellular Proteomics*, Jun 2017.

[8] M. Carrera, Benito Ca nas, and J. M. Gallardo. Chapter 19 - proteomic identification of commercial fish species. In Michelle L. Colgrave, editor, *Proteomics in Food Science*, pages 317 – 330. Academic Press, 2017.

[9] M. C. Chambers, B. Maclean, R. Burke, D. Amodei, D. L. Ruderman, S. Neumann, L. Gatto, B. Fischer, B. Pratt, J. Egertson, K. Hoff, D. Kessner, N. Tasman, N. Shulman, B. Frewen, T. A. Baker, M. Y. Brusniak, C. Paulse, D. Creasy, L. Flashner, K. Kani, C. Moulding, S. L. Seymour, L. M. Nuwaysir, B. Lefebvre, F. Kuhlmann,

J. Roark, P. Rainer, S. Detlev, T. Hemenway, A. Huhmer, J. Langridge, B. Connolly, T. Chadick, K. Holly, J. Eckels, E. W. Deutsch, R. L. Moritz, J. E. Katz, D. B. Agus, M. J. MacCoss, D. L. Tabb, and P. Mallick. A cross-platform toolkit for mass spectrometry and proteomics. *Nature Biotechnology*, 30(10):918–920, 2012.

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

[11] R. Craig and R. C. Beavis. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20:1466–1467, 2004.

[12] B. Diament and W. S. Noble. Faster SEQUEST searching for peptide identification from tandem mass spectra. *Journal of Proteome Research*, 10(9):3871–3879, 2011.

[13] J. Doellinger, A. Schneider, M. Hoeller, and P. Lasch. Sample preparation by easy extraction and digestion (speed) - a universal, rapid, and detergent-free protocol for proteomics based on acid extraction. *Molecular & Cellular Proteomics*, 19(1):209–222, 2020.

[14] V. Dorfer, P. Pichler, T. Stranzl, J. Stadlmann, T. Taus, S. Winkler, and K. Mechtler. MSAmanda, a universal identification algorithm optimized for high accuracy tandem mass spectra. *Journal of Proteome Research*, 13(8):3679–3684, 2014.

[15] J. Edmonds. Matroids, submodular functions, and certain polyhedra. *Combinatorial Structures and Their Applications*, pages 69–87, 1970.

[16] N. Edwards, X. Wu, and C. Tseng. An unsupervised, model-free, machine-learning combiner for peptide identifications from tandem mass spectra. *Clinical Proteomics*, 5(1):23, 2009.

[17] B. Efron. Microarrays, empirical bayes and the two-groups model. *Statistical Science*, 23(1):1–22, 2008.

[18] B. Efron. Simultaneous inference: When should hypothesis testing problems be combined? *The Annals of Applied Statistics*, pages 197–223, 2008.

[19] J. E. Elias and S. P. Gygi. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nature Methods*, 4(3):207–214, 2007.

[20] J. E. Elias and S. P. Gygi. Target-decoy search strategy for mass spectrometry-based proteomics. *Methods in Molecular Biology*, 604(55–71), 2010.

[21] J. K. Eng, T. A. Jahan, and M. R. Hoopmann. Comet: an open source tandem mass spectrometry sequence database search tool. *Proteomics*, 13(1):22–24, 2012.

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

[23] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 1998.

[24] Y. Fu. Bayesian false discovery ratesfor post-translational modification proteomics. *Statistics and Its Interface*, 5:47–59, 2012.

[25] Y. Fu and X. Qian. Transferred subgroup false discovery rate for rare post-translational modifications detected by mass spectrometry. *Molecular and Cellular Proteomics*, 13(5):1359–1368, 2014.

[26] H. Gőtzke, C. Muheim, A. F. Altelaar, A. J.R. Heck., G. Maddalo, and D. O. Daley. Identification of putative substrates for the periplasmic chaperone yfgm in escherichia coli using quantitative proteomics. *Molecular & Cellular Proteomics*, 14(1):216–226, 2015.

[27] K. He, Y. Fu, W.-F. Zeng, L. Luo, H. Chi, C. Liu, L.-Y. Qing, R.-X. Sun, and S.-M. He. A theoretical foundation of the target-decoy search strategy for false discovery rate control in proteomics. *arXiv*, 2015. https://arxiv.org/abs/1501.00537.

[28] M. R. Hoopmann, G. Finney, and M. J. MacCoss. High-speed data reduction, feature detection, and MS/MS spectrum quality assessment of shotgun proteomics datasets using high-resolution mass spectrometry. *Analytical Chemistry*, 79:5620–5632, 2007.

[29] T. Y. Hou, C. Chiang-Ni, and S. H. Teng. Current status of MALDI-TOF mass spectrometry in clinical microbiology. *J Food Drug Anal*, 27(2):404–414, 04 2019.

[30] S. Houel, R. Abernathy, K. Renganathan, K. Meyer-Arendt, N. G. Ahn, and W. M. Old. Quantifying the impact of chimera MS/MS spectra on peptide identification in large-scale proteomics studies. *Journal of Proteome Research*, 9(8):4152–4160, 2010.

[31] 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.

[32] P. Jagtap, J. Goslinga, J. A. Kooren, T. McGowan, M. S. Wroblewski, S. L. Seymour, and T. J. Griffin. A two-step database search method improves sensitivity in peptide sequence matches for metaproteomics and proteogenomics studies. *Proteomics*, 13(8):1352–1357, 2013.

[33] S. Ji, D. Xu, M. Li, Y. Wang, and D. Zhang. Stochastic greedy algorithm is still good: Maximizing submodular + supermodular functions. In H. Le Thi, H. Le, and T. Pham Dinh, editors, *Optimization of Complex Systems: Theory, Models, Algorithms and Applications*, pages 488–497, Cham, 2020. Springer International Publishing.

[34] L. Käll, J. Canterbury, J. Weston, W. S. Noble, and M. J. MacCoss. A semi-supervised machine learning technique for peptide identification from shotgun proteomics datasets. *Nature Methods*, 4:923–25, 2007.

[35] R. Karlsson, A. Thorsell, M. Gomila, F. Salvà-Serra, H. E. Jakobsson, L. Gonzales-Siles, D. Jaén-Luchoro, S. Skovbjerg, J. Fuchs, A. Karlsson, F. Boulund, A. Johnning, E. Kristiansson, and E. R. B. Moore. Discovery of Species-unique Peptide Biomarkers of Bacterial Pathogens by Tandem Mass Spectrometry-based Proteotyping. *Mol Cell Proteomics*, 19(3):518–528, Mar 2020.

[36] E. Katsevich and A. Ramdas. Simultaneous high-probability bounds on the false discovery proportion in structured, regression, and online settings. *arXiv preprint arXiv:1803.06790*, 2019.

[37] U. Keich and W. S. Noble. On the importance of well calibrated scores for identifying shotgun proteomics spectra. *Journal of Proteome Research*, 14(2):1147–1160, 2015.

[38] U. Keich, K. Tamura, and W. S. Noble. Averaging strategy to reduce variability in target-decoy estimates of false discovery rate. *Journal of Proteome Research*, 18(2):585–593, 2018.

[39] A. Keller, A. I. Nesvizhskii, E. Kolker, and R. Aebersold. Empirical statistical model to estimate the accuracy of peptide identification made by MS/MS and database search. *Analytical Chemistry*, 74:5383–5392, 2002.

[40] A. Kertesz-Farkas, U. Keich, and W. S. Noble. Tandem mass spectrum identification via cascaded search. *Journal of Proteome Research*, 14(8):3027–3038, 2015.

[41] D. Kessner, M. Chambers, R. Burke, D. Agnus, and P. Mallick. Proteowizard: open source software for rapid proteomics tools development. *Bioinformatics*, 24(21):2534–2536, 2008.

[42] Y. J. Kil, C. Becker, W. Sandoval, D. Goldberg, and M. Bern. Preview: a program for surveying shotgun proteomics tandem mass spectrometry data. *Analytical Chemistry*, 83(13):5259–5267, 2011.

[43] M. Kim, S. M. Pinto, D. Getnet, R. S. Nirujogi, S. S. Manda, R. Chaerkady, A. K. Madugundu, D. S. Kelkar, R. Isserlin, S. Jain, et al. A draft map of the human proteome. *Nature*, 509(7502):575–581, 2014.

[44] S. Kim, N. Gupta, and P. A. Pevzner. Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. *Journal of Proteome Research*, 7:3354–3363, 2008.

[45] S. Kim and P. A. Pevzner. MS-GF+ makes progress toward a universal database search tool for proteomics. *Nature Communications*, 5:5277, 2014.

[46] 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.

[47] P. Kumar, J. E. Johnson, C. Easterly, S. Mehta, R. Sajulga, B. Nunn, P. D. Jagtap, and T. J. Griffin. A sectioning and database enrichment approach for improved peptide spectrum matching in large, genome-guided protein sequence databases. *Journal of Proteome Research*, 19(7):2772–2785, 2020.

[48] I. K. Kushner, G. Clair, S. O. Purvine, J. Y. Lee, J. N. Adkins, and S. H. Payne. Individual Variability of Protein Expression in Human Tissues. *J Proteome Res*, 17(11):3914–3922, 11 2018.

[49] P. Lasch, A. Schneider, C. Blumenscheit, and J. Doellinger. Identification of Microorganisms by Liquid Chromatography-Mass Spectrometry (LC-MS1) and in Silico Peptide Mass Libraries. *Mol Cell Proteomics*, 19(12):2125–2138, 12 2020.

[50] L. I. Levitsky, M V. Ivanov, A. A. Lobas, and M. V. Gorshkov. Unbiased false discovery rate estimation for shotgun proteomics based on the target-decoy approach. *Journal of Proteome Research*, 16(2):393–397, 2017.

[51] M. W. Libbrecht, J. A. Bilmes, and W. S. Noble. Choosing non-redundant representative subsets of protein sequence data sets using submodular optimization. *Proteins*, 86(4):454–466, 2018.

[52] A. Lin, J. J. Howbert, and W. S. Noble. Combining high-resolution and exact calibration to boost statistical power: A well-calibrated score function for high-resolution ms2 data. *Journal of Proteome Research*, 17:3644–3656, 2018.

[53] A. Lin, E. D. Merkley, B. H. Clowers, J. R. Hutchison, and H. W. Kreuzer. Effects of bacterial inactivation methods on downstream proteomic analysis. *J Microbiol Methods*, 112:3–10, May 2015.

[54] Hui Lin and Jeff A. Bilmes. Optimal selection of limited vocabulary speech corpora. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Florence, Italy, August 2011.

[55] S. E. Lindner, K. E. Swearingen, M. J. Shears, M. P. Walker, E. N. Vrana, K. J. Hart, A. M. Minns, P. Sinnis, R. L. Moritz, and S. H. I. Kappe. Transcriptomics and proteomics reveal two waves of translational repression during the maturation of malaria parasite sporozoites. *Nature Communications*, 10(1):4964–4964, Oct 2019.

[56] D. Luo, Y. He, K. Emery, W. S. Noble, and U. Keich. Competition-based control of the false discovery proportion. *arXiv preprint arXiv:2011.11939*, 2020.

[57] L. Martens, M. Chambers, M. Sturm, D. Kessner, F. Levander, J. Shofstahl, W. H. Tang, A. Römpp, S. Neumann, A. D. Pizarro, L. Montecchi-Palazzi, N. Tasman, M. Coleman, F. Reisinger, P. Souda, H. Hermjakob, P. A. Binz, and E. W. Deutsch. mzML–a community standard for mass spectrometry data. *Mol Cell Proteomics*, 10(1):R110.000133, Jan 2011.

[58] D. H. May, K. Tamura, and W. S. Noble. Param-Medic: A tool for improving MS/MS database search yield by optimizing parameter settings. *Journal of Proteome Research*, 16(4):1817–1824, 2017.

[59] D. H. May, K. Tamura, and W. S. Noble. Detecting modifications in proteomics experiments with Param-Medic. *Journal of Proteome Research*, 18(4):1902–1906, 2019.

[60] Damon H. May, Emma Timmins-Schiffman, Molly P. Mikan, H. Rodger Harvey, Elhanan Borenstein, Brook L. Nunn, and William S. Noble. An alignment-free "metapeptide" strategy for metaproteomic characterization of microbiome samples using shotgun metagenomic sequencing. *Journal of Proteome Research*, 15(8):2697–2705, 2016.

[61] W. H. McDonald, D. L. Tabb, R. G. Sadygov, M. J. MacCoss, J. Venable, J. Graumann, J. R Johnson, D. Cociorva, and J. R. Yates, III. MS1, MS2, and SQT-three unified, compact, and easily parsed file formats for the storage of shotgun proteomic spectra and identifications. *Rapid Communications in Mass Spectrometry*, 18:2162–2168, 2004.

[62] S. McIlwain, K. Tamura, A. Kertesz-Farkas, C. E. Grant, B. Diament, B. Frewen, J. J. Howbert, M. R. Hoopmann, L. Käll, J. K. Eng, M. J. MacCoss, and W. S. Noble. Crux: rapid open source protein tandem mass spectrometry analysis. *Journal of Proteome Research*, 13(10):4488–4491, 2014.

[63] E. D. Merkley, S. C. Jenson, J. S. Arce, A. M. Melville, O. P. Leiser, D. S. Wunschel, and K. L. Wahl. Ricin-like proteins from the castor plant do not influence liquid chromatography-mass spectrometry detection of ricin in forensically relevant samples. *Toxicon*, 140:18–31, 2017.

[64] B. Mesuere, B. Devreese, G. Debyser, M. Aerts, P. Vandamme, and P. Dawyndt. Unipept: Tryptic peptide-based biodiversity analysis of metaproteome samples. *Journal of Proteome Research*, 11(12):5773–5780, 2012. PMID: 23153116.

[65] A. D. Mooradian, S. van der Post, K. M. Naegle, and J. M. Held. Proteoclade: A taxonomic toolkit for multi-species and metaproteomic analysis. *PLOS Computational Biology*, 16(3):1–12, 03 2020.

[66] 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.

[67] S. Nahnsen, A. Bertsch, J. Rahnenführer, A. Nordheim, and O. Kohlbacher. Probabilistic consensus scoring improves tandem mass spectrometry peptide identification. *Journal of Proteome Research*, 10(8):3332–3343, 2011.

[68] A. I. Nesvizhskii. A survey of computational methods and error rate estimation procedures for peptide and protein identification in shotgun proteomics. *Journal of Proteomics*, 73(11):2092 – 2123, 2010.

[69] W. S. Noble. Mass spectrometrists should only search for peptides they care about. *Nature Methods*, 12(7):605–608, 2015.

[70] W. S. Noble and U. Keich. Response to "Mass spectrometrists should search for all peptides, but assess only the ones they care about". *Nature Methods*, 14(7):644, 2017.

[71] J.G. Oxley. *Matroid Theory: Second Edition*. Oxford University Press, 2011.

[72] M. Palmblad and A. M. Deelder. Molecular phylogenetics by direct comparison of tandem mass spectra. *Rapid Commun Mass Spectrom*, 26(7):728–732, Apr 2012.

[73] C. Y. Park, A. A. Klammer, L. Käll, M. P. MacCoss, and W. S. Noble. Rapid and accurate peptide identification from tandem mass spectra. *Journal of Proteome Research*, 7(7):3022–3027, 2008.

[74] B. N. Pease, E. L. Huttlin, M. P. Jedrychowski, E. Talevich, J. Harmon, T. Dillman, N. Kannan, C. Doerig, R. Chakrabarti, S. P. Gygi, and D. Chakrabarti. Global analysis of protein expression and phosphorylation of three stages of *Plasmodium falciparum* intraerythrocytic development. *Journal of Proteome Research*, 12:4028–4045, 2013.

[75] Y. Perez-Riverol, A. Csordas, J. Bai, M. Bernal-Llinares, S. Hewapathirana, D. J. Kundu, A. Inuganti, J. Griss, G. Mayer, M. Eisenacher, E. Pérez, J. Uszkoreit, J. Pfeuffer, T. Sachsenberg, S. Yilmaz, S. Tiwary, J. Cox, E. Audain, M. Walzer, A. F. Jarnuczak, T. Ternent, A. Brazma, and J. A. Vizcaíno. The PRIDE database and related tools and resources in 2019: improving support for quantification data. *Nucleic Acids Res*, 47(D1):D442–D450, 01 2019.

[76] C. Ramus, A. Hovasse, M. Marcellin, A. Hesse, E. Mouton-Barbosa, D. Bouyssia, S. Vaca, C. Carapito, K. Chaoui, C. Bruley, J. Garin, S. Cianfaani, M. Ferro, A. V. Dorssaeler, O. Burlet-Schiltz, C. Schaeffer, Y. Coutaa, and A. Gonzalez de Peredo. Spiked proteomic standard dataset for testing label-free quantitative software and statistical methods. *Data in Brief*, 6, March 2016.

[77] V. Rieder, B. Blank-Landeshammer, M. Stuhr, T. Schell, K. Biß, L. Kollipara, A. Meyer, M. Pfenninger, H. Westphal, A. Sickmann, and J. Rahnenführer. DISMS2: A flexible algorithm for direct proteome- wide distance calculation of LC-MS/MS runs. *BMC Bioinformatics*, 18(1):148, Mar 2017.

[78] H. L. Röst, T. Sachsenberg, S. Aiche, C. Bielow, H. Weisser, F. Aicheler, S. Andreotti, H. Ehrlich, P. Gutenbrunner, E. Kenar, X. Liang, S. Nahnsen, L. Nilse, J. Pfeuffer, G. Rosenberger, M. Rurik, U. Schmitt, J. Veit, M. Walzer, D. Wojnar, W. E. Wolski, O. Schilling, J. S. Choudhary, L. Malmström, R. Aebersold, K. Reinert, and O. Kohlbache. OpenMS: a flexible open-source software platform for mass spectrometry data analysis. *Nature Methods*, 13(9):741, 2016.

[79] F. Roux-Dalvai, C. Gotti, M. Leclercq, M. Hélie, M. Boissinot, T. N. Arrey, C. Dauly, . Fournier, I. Kelly, J. Marcoux, J. Bestman-Smith, Mi. G. Bergeron, and A. Droit. Fast and accurate bacterial species identification in urine specimens using lc-ms/ms mass spectrometry and machine learning. *Molecular and Cellular Proteomics*, 18(12):2492 – 2505, 2019.

[80] H. L. Röst, U. Schmitt, R. Aebersold, and L. Malmström. pyOpenMS: a Python-based interface to the OpenMS mass-spectrometry algorithm library. *Proteomics*, 14(1):74–77, Jan 2014.

[81] J. M. Schreiber, J. Bilmes, and W. S. Noble. Prioritizing transcriptomic and epigenomic experiments by using an optimization strategy that leverages imputed data. *bioRxiv*, 2019. https://www.biorxiv.org/content/10.1101/708107v1.

[82] J. M. Schreiber, J. Bilmes, and W. S. Noble. apricot: Submodular selection for data summarization in python. *Journal of Machine Learning Research*, 21(161):1–6, 2020.

[83] B. C. Searle, M. Turner, and A. I. Nesvizhskii. Improving sensitivity by probabilistically combining results from multiple MS/MS search methodologies. *The Journal of Proteome Research*, 7(1):245–253, 2008.

[84] D. Shteynberg, E. W. Deutsch, H. Lam, J. K. Eng, Z. Sun, N. Tasman, L. Mendoza, R. L. Moritz, R. Aebersold, and A. I. Nesvizhskii. iProphet: multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates. *Molecular & Cellular Proteomics*, 10(12):M111–007690, 2011.

[85] M. Sielaff, J. Kuharev, T. Bohn, J. Hahlbrock, T. Bopp, S. Tenzer, and U. Distler. Evaluation of FASP, SP3, and iST protocols for proteomic sample preparation in the low microgram range. *Journal of Proteome Research*, 16(11):4060–4072, 2017.

[86] A. Sticker, L. Martens, and L. Clement. Mass spectrometrists should search for all peptides, but assess only the ones they care about. *Nature Methods*, 14(7):643–644, 2017.

[87] T. Sultana, R. Jordan, and J. Lyons-Weiler. Optimization of the use of consensus methods for the detection and putative identification of peptides via mass spectrometry using protein standard mixtures. *Journal of Proteomics and Bioinformatics*, 2(6):262, 2009.

[88] K. Taejoon, H. Choi, C. Vogel, A. I. Nesvizhskii, and E. M. Marcotte. MSblendera: A probabilistic approach for integrating peptide identifications from multiple database search engines. *Journal of Proteome Research*, 10(7):2949–2958, 2011.

[89] The UniProt Consortium. UniProt: a worldwide hub for protein knowledge. *Nucleic Acids Research*, pages D506–D515, 2019.

[90] A. Z. Tremp, S. Saeed, V. Sharma, E. Lasonder, and J. T. Dessens. Plasmodium berghei laps form an extended protein complex that facilitates crystalloid targeting and biogenesis. *Journal of Proteomics*, 227:103925, 2020.

[91] S. Tsuchida, H. Umemura, and T. Nakayama. Current Status of Matrix-Assisted Laser Desorption/Ionization-Time-of-Flight Mass Spectrometry (MALDI-TOF MS) in Clinical Diagnostic Microbiology. *Molecules*, 25(20), Oct 2020.

[92] S. J. van der Plas-Duivesteijn, T. Wulff, O. Klychnikov, D. Ohana, H. Dalebout, P. A. van Veelen, J. de Keijzer, M. A. Nessen, Y. E. van der Burgt, A. M. Deelder, and M. Palmblad. Differentiating samples and experimental protocols by direct comparison of tandem mass spectra. *Rapid Commun Mass Spectrom*, 30(6):731–738, Mar 2016.

[93] M. Vaudel, J. M. Burkhart, R. P. Zahedi, E. Oveland, F. S. Berven, A. Sickmann, L. Martens, and H. Barsnes. PeptideShaker enables reanalysis of MS-derived proteomics data sets. *Nature Biotechnology*, 33(1):22–24, 2015.

[94] K. Verheggen, H. Raeder, F. S. Berven, L. Martens, H. Barsnes, and M. Vaudel. Anatomy and evolution of database search engines—a central component of mass spectrometry based proteomic workflows. *Mass Spectrometry Reviews*, 2017. Epub ahead of print.

[95] D. C. Wedge, R. Krishna, P. Blackhurst, J. A. Siepen, A. R. Jones, and S. J. Hubbard. FDRAnalysis: a tool for the integrated analysis of tandem mass spectrometry identification results from multiple search engines. *Journal of Proteome Research*, 10(4):2088–2094, 2011.

[96] K. Wei, M. W. Libbrecht, J. A. Bilmes, and W. S. Noble. Choosing panels of genomics assays using submodular optimization. *Genome Biology*, 17(1):229, 2016.

[97] M. Welker. Proteomics for routine identification of microorganisms. *PROTEOMICS*, 11(15):3143–3153, 2011.

[98] C. D. Wenger and J. J. Coon. A proteomics search algorithm specifically designed for high-resolution tandem mass spectra. *Journal of Proteome Research*, 12(3):1377–1386, 2013.

[99] J. R. Wiśniewski, Alexandre Zougman, Nagarjuna Nagaraj, and Matthias Mann. Universal sample preparation method for proteome analysis. *Nature Methods*, 13:359–362, 2009.

[100] S. Woo, S. W. Cha, S. Bonissone, S. Na, D. L. Tabb, P. A. Pevzner, and V. Bafna. Advanced proteogenomic analysis reveals multiple peptide mutations and complex immunoglobulin peptides in colon cancer. *Journal of Proteome Research*, 14(9):3555–3567, 2015.

[101] X. Yi, F. Gong, and Y. Fu. Transfer posterior error probability estimation for peptide identification. *BMC Bioinformatics*, 21, May 2020.

Appendix A

APPENDIX TO "IMPROVING POWER WHILE CONTROLLING THE FALSE DISCOVERY RATE WHEN ONLY A SUBSET OF PEPTIDES ARE RELEVANT"

Algorithm 1 Search-then-select

procedure searchThenSelect($S, \mathcal{T}, \mathcal{D}, \mathcal{T}_r, \alpha$)
    $(M, P) :=$ Search($S, \mathcal{T} \cup \mathcal{D}$)
    $A :=$ ControlFDR(M, P, α)
    $R := \{(s_i, p_i, m_i) \mid a_i = 1, p_i \in \mathcal{T}_r\}$
    return R
end procedure

Algorithm 2 Subset-search

procedure subsetSearch($S, \mathcal{T}_r, \mathcal{D}_r, \alpha$)
    $(M, P) :=$ Search($S, \mathcal{T}_r \cup \mathcal{D}_r$)
    $A :=$ ControlFDR(M, α)
    $R := \{(s_i, p_i, m_i) \mid a_i = 1\}$
    return R
end procedure

Algorithm 3 Group-FDR

procedure GROUPFDR $(S, \mathcal{T}_r, \mathcal{D}_r, \mathcal{T}_i, \mathcal{D}_i, \alpha)$
  $(M, P) := \text{SEARCH}(S, \mathcal{T}_r \cup \mathcal{T}_i \cup \mathcal{D}_r \cup \mathcal{D}_i)$
  $(M^1, P^1) := \{(m_i, p_i) \mid p_i \in \mathcal{T}_r \cup \mathcal{D}_r\}$
  $A := \text{CONTROLFDR}(M^1, \alpha)$
  $R := \{(s_i^1, p_i^1, m_i^1) \mid a_i = 1\}$
  return R
end procedure

Algorithm 4 All-sub

procedure ALLSUB$(S, \mathcal{T}, \mathcal{D}, \mathcal{T}_r, \alpha)$
  $(M, P) := \text{SEARCH}(S, \mathcal{T} \cup \mathcal{D})$
  $A := \text{CONTROLFDR2}(M, \alpha)$
  $R := \{(s_i, p_i, m_i) \mid a_i = 1\}$
  return R
end procedure

Algorithm 5 Subset-neighbor search (SNS)

procedure SNS $(S, \mathcal{T}_r, \mathcal{D}_r, \mathcal{T}_n, \mathcal{D}_n, \alpha)$
  $(M, P) := \text{SEARCH}(S, \mathcal{T}_r \cup \mathcal{T}_n \cup \mathcal{D}_r \cup \mathcal{D}_n)$
  $(M^1, P^1) := \{(m_i, p_i) \mid p_i \in \mathcal{T}_r \cup \mathcal{D}_r\}$
  $A := \text{CONTROLFDR}(M^1, \alpha)$
  $R := \{(s_i^1, p_i^1, m_i^1) \mid a_i = 1\}$
  return R
end procedure

file name	# scans
UPS1_12500amol_R1.ms2	39718
UPS1_12500amol_R2.ms2	39682
UPS1_12500amol_R3.ms2	39782
UPS1_25000amol_R1.ms2	40856
UPS1_25000amol_R2.ms2	40512
UPS1_25000amol_R3.ms2	40439
UPS1_50000amol_R1.ms2	41833
UPS1_50000amol_R2.ms2	41653
UPS1_50000amol_R3.ms2	41665
UPS1_5000amol_R1.ms2	37918
UPS1_5000amol_R2.ms2	38046
UPS1_5000amol_R3.ms2	38052

Table A.1: **yeast/UPS1 data.** The number of scans found in each run. This is for the samples where UPS1 was spiked into yeast.

*file name	cultivar	preparation method	# scans	*file name	cultivar	preparation method	# scans
Zanz_1_1_03Jun16	*Zanzibarensis*	M0	15735	GCH4_1_1_03Jun16	GCH4	M0	21634
Zanz_1_2_03Jun16	*Zanzibarensis*	M0	17138	GCH4_1_2_03Jun16	GCH4	M0	16765
Zanz_1_3_03Jun16	*Zanzibarensis*	M0	18834	GCH4_1_3_03Jun16	GCH4	M0	18663
Zanz_2_1_03Jun16	*Zanzibarensis*	M0	19760	GCH4_2_1_03Jun16	GCH4	M0	19830
Zanz_2_2_03Jun16	*Zanzibarensis*	M0	18986	GCH4_2_2_03Jun16	GCH4	M0	24944
Zanz_2_3_03Jun16	*Zanzibarensis*	M0	16222	GCH4_2_3_03Jun16	GCH4	M0	21667
Zanz_3_1_03Jun16	*Zanzibarensis*	M0	18900	GCH4_3_1_03Jun16	GCH4	M0	20267
Zanz_3_2_03Jun16	*Zanzibarensis*	M0	21368	GCH4_3_2_03Jun16	GCH4	M0	22246
Zanz_3_3_03Jun16	*Zanzibarensis*	M0	20315	GCH4_3_3_03Jun16	GCH4	M0	25330
TMVCH1_1_1_03Jun16	TMVCH1	M0	14893	200_1_03Jun16	200	M0	13742
TMVCH1_1_2_03Jun16	TMVCH1	M0	20086	200_2_03Jun16	200	M0	17822
TMVCH1_1_3_03Jun16	TMVCH1	M0	23470	200_3_03Jun16	200	M0	6158
TMVCH1_2_1_03Jun16	TMVCH1	M0	19125	5952_1_03Jun16	592	M0	14951
TMVCH1_2_2_03Jun16	TMVCH1	M0	23804	592_2_03Jun16	592	M0	16157
TMVCH1_2_3_03Jun16	TMVCH1	M0	20689	592_3_03Jun16	592	M0	14125
TMVCH1_3_1_03Jun16	TMVCH1	M0	16456	611_1_03Jun16	611	M0	10724
TMVCH1_3_2_03Jun16	TMVCH1	M0	25938	611_2__03Jun16	611	M0	13621
TMVCH1_3_3_03Jun16	TMVCH1	M0	16414	611_3_03Jun16	611	M0	8454
1_M0_AM_R1_7Mar16	PNNL	M0	5076	829_1_03Jun16	829	M0	17343
2_M2_JA_R3_7Mar16	PNNL	M2	16784	3_M0_JA_R3_7Mar16	829	M0	18592
3_M0_JA_R3_7Mar16	PNNL	M0	15859	829_3_03Jun16	829	M0	9475
3_M0_JA_R3_B_03Jun16	PNNL	M0	22015	8_M1_JA_R2_7Mar16	PNNL	M1	15669
4_M1_AM_R1_7Mar16	PNNL	M1	13188	9_M4_AM_R1_7Mar16	PNNL	M4	14892
5_M2_AM_R1_7Mar16	PNNL	M2	15358	9_M4_JA_R2_7Mar16	PNNL	M4	16140
6_M0_JA_R2_7Mar16	PNNL	M0	16728	9_M4_JA_R3_7Mar16	PNNL	M4	15472
6_M0_JA_R2_B_03Jun16	PNNL	M0	21544	10_M2_JA_R2_7Mar16	PNNL	M2	18358
7_M1_JA_R3_7Mar16	PNNL	M1	17973	16_M3_03Jun16	PNNL	M3	20583

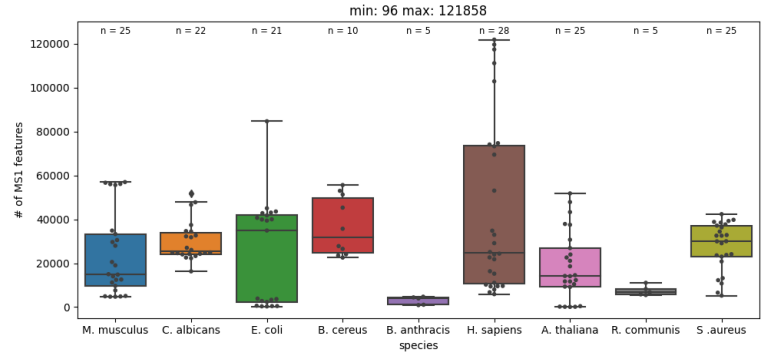
Table A.2: **Ricin data.** The number of scans found in each ricin run. *All file names start with "Rcom_" and end with either "_Samwise_16-03-32.ms2" or "_Samwise_15-08-55.ms2".

file name	# scans
217_2018_ZBS6_HeLa_ISD_1.ms2	89162
220_2018_ZBS6_HeLa_SPEED_1.ms2	87981
222_2018_ZBS6_HeLa_FASP_2.ms2	87398
223_2018_ZBS6_HeLa_ISD_2.ms2	86869
225_2018_ZBS6_HeLa_SP3_2.ms2	90238
226_2018_ZBS6_HeLa_SPEED_2.ms2	88086
228_2018_ZBS6_HeLa_FASP_3.ms2	87255
229_2018_ZBS6_HeLa_ISD_3.ms2	87835
231_2018_ZBS6_HeLa_SP3_3.ms2	90546
232_2018_ZBS6_HeLa_SPEED_3.ms2	87703
234_2018_ZBS6_HeLa_FASP_1.ms2	87797
235_2018_ZBS6_HeLa_SP3_1.ms2	89253

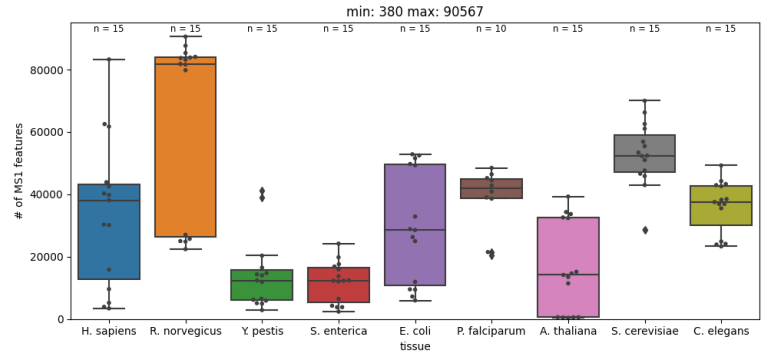
Table A.3: **Human data.** The number of scans found in each run.

Appendix B

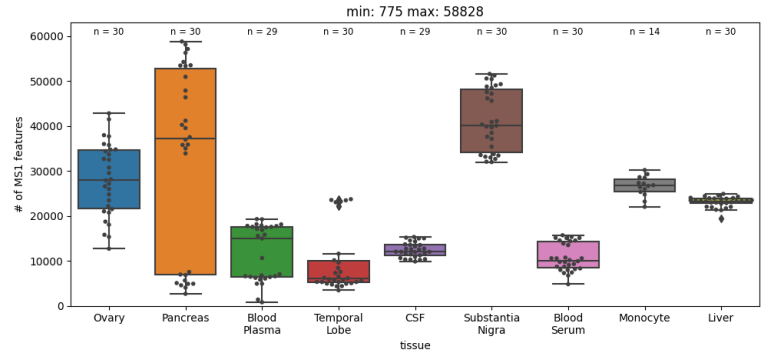
APPENDIX TO "MS1CONNECT: A MASS SPECTROMETRY RUN SIMILARITY MEASURE"



A



B



C

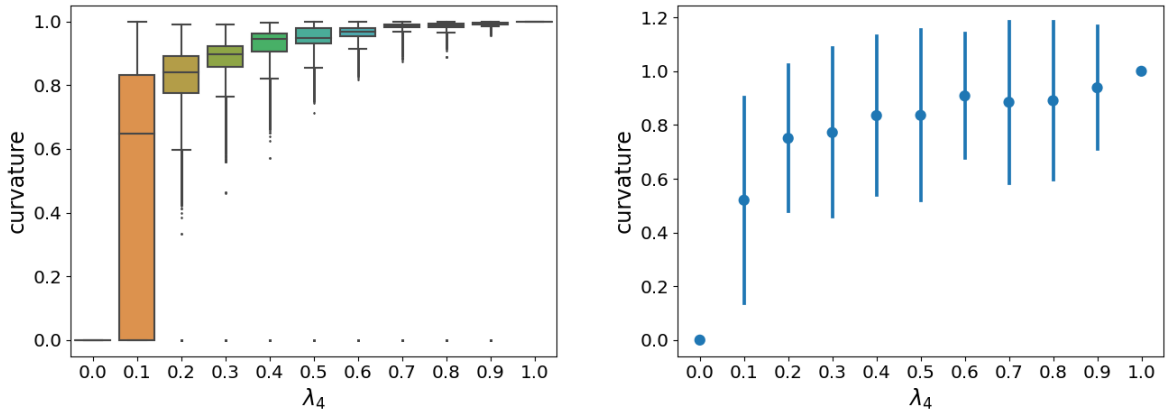
Figure B.1: **Number of MS1 features per run.** A series of boxplots of the number of MS1 features per run split by species or tissue. In addition to the boxplots, each point shows the number of MS1 features per run. The text near the top of the plot indicates the number of runs that each label has. A) The number of MS1 features per run for the species training data. B) The number of MS1 features per run for the species test data. C) The number of MS1 features per run for the tissue test data.

Variable	Definition
$G = (U, V, E)$	Bipartite graph containing two vertex sets U and V and edge set E
U	vertex set of MS1 features
V	vertex set of MS1 features
E	set of valid edges
u	node in U
v	node in V
e	instance of valid edge
$s(e)$	time shift of edge e
$u(e)$	MS1 feature u that is a part of edge e
$v(e)$	MS1 feature v that is a part of edge e
$t(u)$	normalized retention time of MS1 feature u
$m(u)$	m/z of MS1 feature u
$I(u)$	normalized intensity of u

Table B.1: **Notation** Notation used in this chapter.

Method	# MS1 Features	m/z tolerance	λ_1	λ_2	λ_3	λ_4	α	β	γ
MS1Connect	4000	0.0035	0.0	0.0	0.1	0.9	1.0	0.01	1×10^{-6}
M_1 only	1000	0.0035	1.0	0.0	0.0	0.0	NA	NA	NA
M_2 only	500	0.0035	0.0	1.0	0.0	0.0	NA	NA	NA
M_3 only	1000	0.0035	0.0	0.0	1.0	0.0	1×10^{-5}	NA	NA
M_4 only	4000	0.0035	0.0	0.0	0.0	1.0	NA	1.0	1e-05
# m/z bins in common	1000	0.0071	NA	NA	NA	NA	NA	NA	NA

Table B.2: **Hyperparameters.** The best set of hyperparameters for each method.



A B

Figure B.2: **Curvature as a function of** λ_4 These plots show the curvature value as a function of λ_4. The best performance of MS1Connect occurs when $\lambda_4 = 0.9$ A) A boxplot of the curvature values showing mean and inter-quartile range. B) A plot of the mean and standard deviation of the curvature values.

method	training data				test data			
	k=1	2	3	4	k=1	2	3	4
MS1Connect	**0.994**	**0.997**	**0.988**	0.9804	**1.000**	0.996	0.990	**0.989**
M_1 only	**0.994**	0.991	0.986	0.980	**1.000**	**1.000**	0.9897	0.9788
M_2 only	0.976	0.961	0.954	0.947	0.962	0.946	0.931	0.917
M_3 only	**0.994**	0.991	0.986	0.982	**1.000**	**1.000**	0.990	0.979
M_4 only	**0.994**	0.994	0.986	0.982	**1.000**	0.996	0.990	**0.986**
random chance	0.130	0.017	0.002	0.0003	0.077	0.005	0.0003	2.0×10^{-5}
# m/z in common	**0.994**	0.994	0.988	**0.982**	**1.000**	**1.000**	**0.992**	0.981

Table B.3: **Precision at** k**.** A table of the precision at k values for the species training and test data where runs from the same PRIDE ID are not removed from the repository. Bolded values denote the best performance for each column.

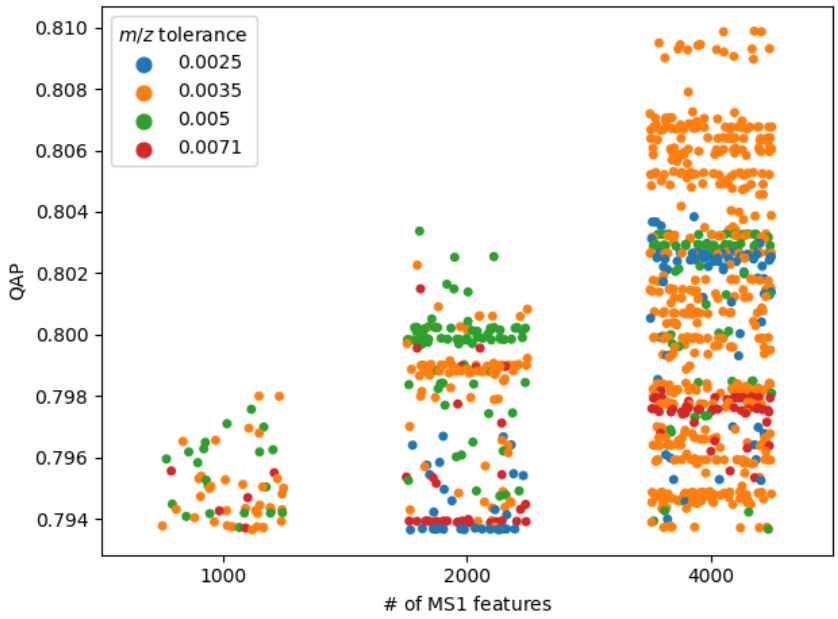


Figure B.3: **Top scoring hyperparameters** A strip plot of the top 1000 scoring sets of hyperparameters. Each point is the performance of a specific set of hyperparameters and is colored by m/z tolerance. Note that most of the high-scoring hyperparameter sets used 4000 MS1 features and has a m/z tolerance of 0.0035.

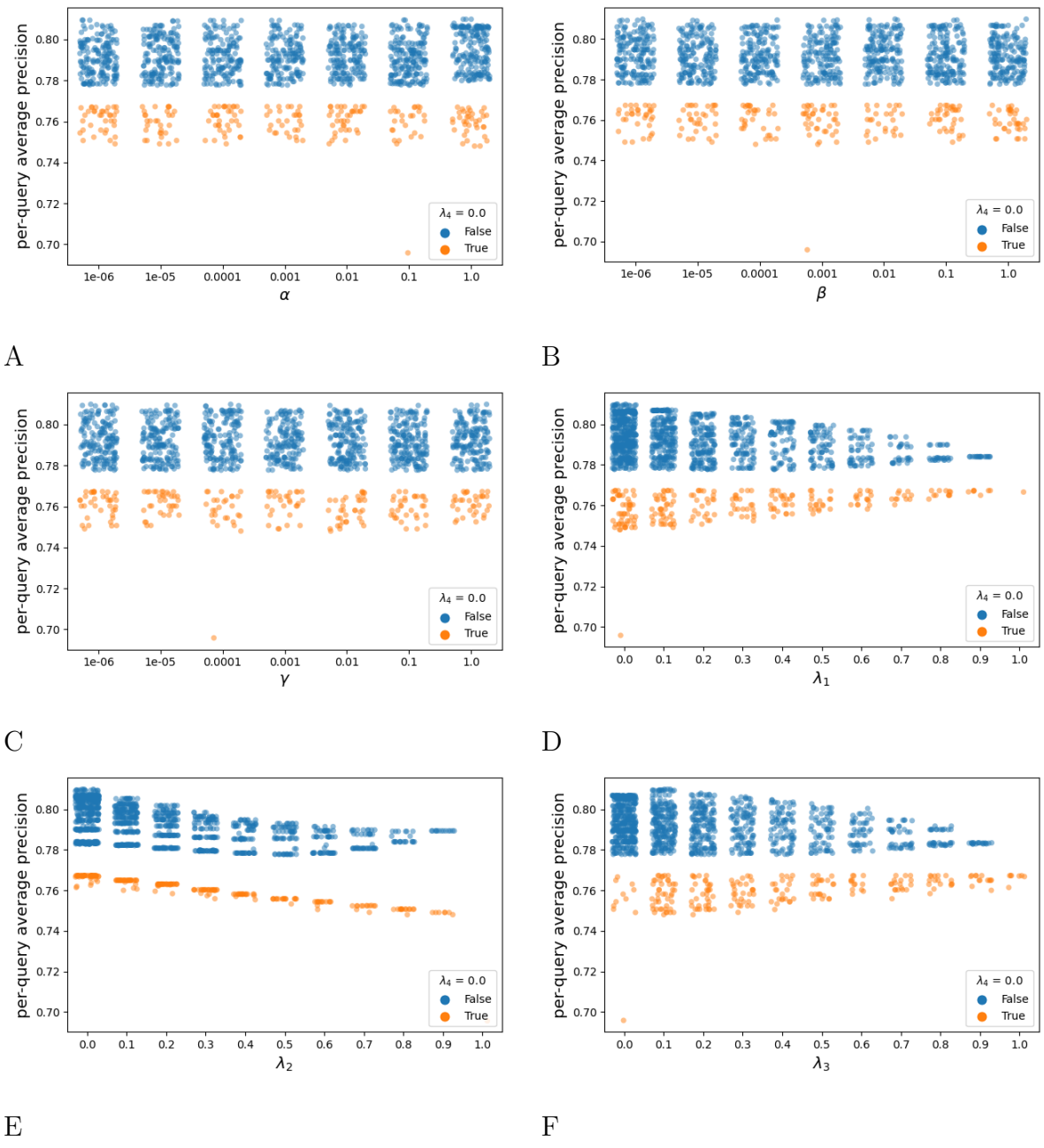


Figure B.4: **Performance split by hyperparameter value.** A strip plot of the QAP split by the possible values of each hyperparameter on the species training dataset when the number of MS1 features is fixed to 4000 and the m/z tolerance is set to 0.0035 Da.

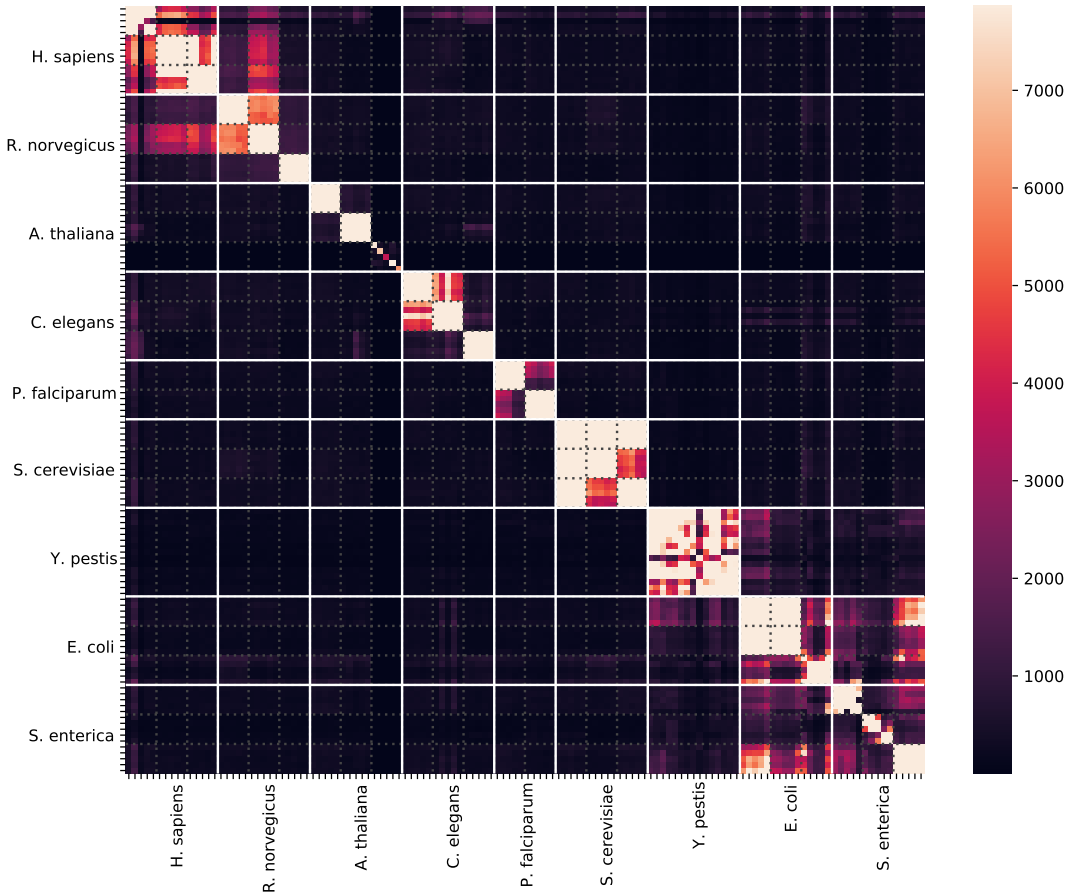


Figure B.5: **Heatmap of MS1Connect scores for species test data.** Each cell is colored by the MS1Connect score between a pair of runs. The solid white lines denote the border between different species while the dotted gray lines delineate the border between different experiments (PRIDE ID).

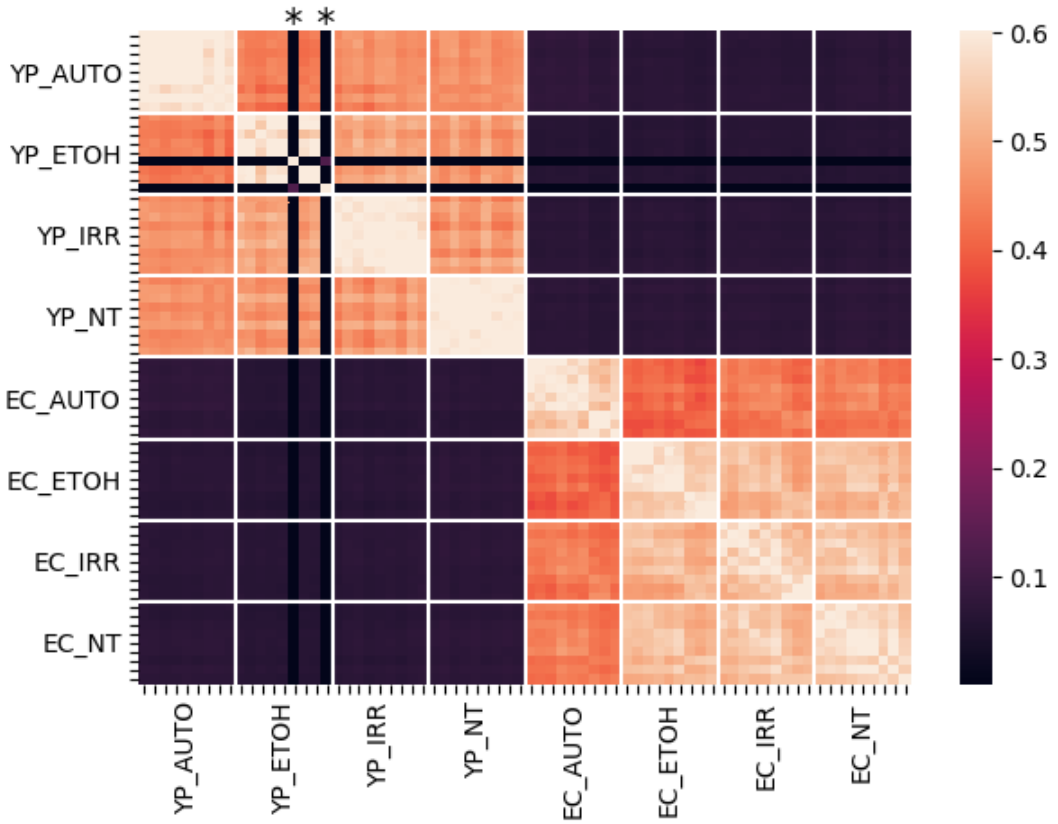


Figure B.6: **Heatmaps of inactivation data.** A heatmap of the Jaccard index of the detected peptides in the bacterial inactivation study. The peptide list for each run was generated from the set of PSMs accepted at a 1% FDR threshold. The solid white lines denote boundaries between groups of runs. "YP" and "EC" correspond to *Y. pestis* and *E. coli*, respectively, while "AUTO", "ETOH", "IRR", and "NT" correspond to autoclave, ethanol, irradiation, and no treatment, respectively. * This black strip is due to two runs that have poor data quality and therefore are not similar to any other run.

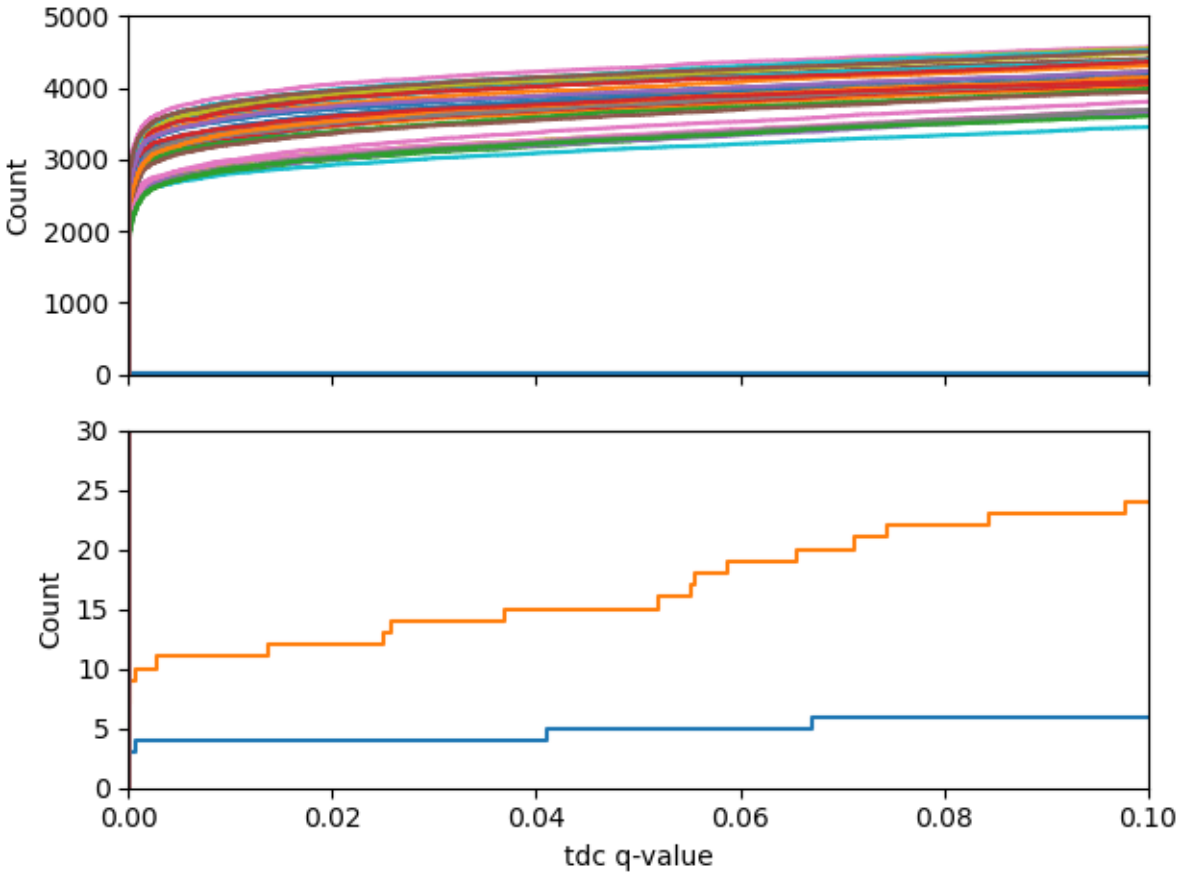


Figure B.7: **Database search.** Each line shows the number of accepted PSMs as a function of q-value threshold for the *Y. pestis* runs in the bacterial inactivation dataset. We see that two of the runs accept a minuscule number of PSMs compared to the remaining datasets. The lower panel is a zoomed in version of the upper panel.