Enabling Community-Driven Proteomics

Damon May

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

Doctor of Philosophy

University of Washington

2018

Reading Committee:

William S. Noble, Chair

Michael J. MacCoss

Brook L. Nunn

Program Authorized to Offer Degree:
Genome Sciences

University of Washington

Abstract

Enabling Community-Driven Proteomics

Damon May

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

As the field of proteomics matures, it faces several computational challenges. This dissertation describes three new computational methods to address some of these challenges: Metapeptides, Param-Medic and GLEAMS.

Metapeptides constructs a database from site-specific metagenomic sequencing of microbial community samples to facilitate identification of mass spectra. Even massive public databases offer incomplete coverage of a given microbial community sample. Metaproteomes assembled from site-specific metagenomic sequencing offer better coverage but fail to include all the variability present in sequencing data. Metapeptides constructs a small, sample-targeted peptide database optimized for database search, offering superior sequence coverage and providing a dramatic boost to metaproteomic database search sensitivity at a controlled false discovery rate (FDR).

Param-Medic infers optimal database search parameters directly from mass spectrometry data. Tight precursor and fragment mass tolerances can increase database search sensitivity at a given FDR. However, too-tight tolerances reduce sensitivity by improperly excluding match candidates and lowering match scores. Param-Medic infers optimal precursor and fragment tolerances by analyzing pairs of acquired spectra that are likely to have been generated by the same peptide ion, yielding more high confidence identifications at a given FDR than tolerances based on per-instrument best practice or even determined by experts.

GLEAMS embeds mass spectra into a low-dimensional space in which spectra generated by the same peptide are close together, enabling rapid propagation of sequence identifications among communities of nearby spectra. Public proteomics repositories contain billions of spectra from researchers around the world, but traditional data analysis workflows fail to take advantage of those data. GLEAMS detects communities of spectra that represent the same peptide. Identifications can be propagated from identified to unidentified spectra, and unidentified communities can then be characterized by targeted downstream analysis. GLEAMS enables identification of 8% more spectra in a sample repository of five million spectra at low computational expense. Scaled up to an entire public repository, GLEAMS offers an efficient, community-driven approach to proteomics data analysis.

TABLE OF CONTENTS

LIST OF FIGURES

GLOSSARY

DE NOVO SEQUENCING: A method for peptide sequencing performed without prior knowledge of the amino acid sequence

FDR: False discovery rate

LC-MS/MS: Liquid chromatography tandem mass spectrometry

MS/MS: Tandem mass spectrometry

M/Z: Mass-to-charge ratio

PSM: peptide-spectrum match

ACKNOWLEDGMENTS

This work could not have been completed, or even begun, without unwavering support and material sacrifice from Laura and Elena May. Thank you for believing in me and sitting through my practice talks.

Bill Noble contributed several central ideas to this work and greatly helped with developing most of the rest. He's been an great advisor, giving guidance and free rein as needed.

Brook Nunn has been a fantastic mentor and collaborator. She taught me everything I know about metaproteomics, and many of the ideas I've developed here spun out of conversations with her.

I've learned a great deal about machine learning from Jeff Bilmes and Jacob Schreiber, two top-notch machine learning researchers who were kind enough to help me start blundering around in their domain.

Neither I nor my work would be possible without my parents, Doug and Ritha May. They raised me with a love of learning and access to every opportunity.

And my graduate program would have been far poorer without Alex Hu, Mike Riffle, Emma Timmins-Schiffman, Lindsay Pino, Wout Bittremieux and the supportive communities of the Noble lab and the UW Department of Genome Sciences.

To our good friends who bought us happy hour beers during our Grand Belt-Tightening: we'll get the next round.

DEDICATION

To my fierce, brilliant, terrifying daughter, Elena.

Never stop learning, changing and growing.

Or we will have *words*.

Chapter 1

INTRODUCTION

Proteomics is the study of the proteins expressed by an organism or community of organisms. Since proteins are the molecules directly responsible for nearly every task of cellular life, proteomics is our most direct molecular method of understanding the activity of organisms and their responses to their environments. By analyzing tandem mass spectra collected by a mass spectrometer, each representing one or more peptides, proteomics researchers make the qualitative and quantitative assessments that underpin all further analyses of proteomes. The sensitivity and accuracy of these assessments, then, is vital to our understanding of the molecular workings of life.

The work described here helps expand the field of computational proteomics in several ways. I describe new methods for determining the identity of peptides present in microbial communities; for estimating the measurement error in proteomics experiments; and for using the vast collections of proteomics data deposited in public data repositories to learn more about some of the cryptic peptide mass spectra found in those repositories.

1.1 Metaproteomics

Metaproteomics is the study of the proteins expressed by communities of microorganisms. This relatively new subfield has made important contributions to our understanding of the human microbiome, environmental microbiomes like soil and ocean, and the effects of environmental contaminants on wastewater.

The backbone of modern proteomics is searching databases of peptides for matches to the mass spectra present in a given experiment. In metaproteomics, the choice of database presents two unique issues. First, metaproteomics samples may contain organisms whose proteomes are not present in publicly available databases. Second, including the proteome of every organism that might be present in the sample in the database to search would explode the search space, requiring correction for false positives due to multiple testing that drastically drives down detection sensitivity.

To address these two issues, many metaproteomics researchers construct protein databases based on metagenomic sequencing of the samples they are studying. The main

approach has been to assemble a metagenome from the metagenomic sequencing reads. However, metagenome assembly is optimized to produce long contigs of high confidence, rather than to represent the sample's proteome comprehensively. Accordingly, it misses many peptide sequences that may be present in the sample because they failed to be included in retained contigs. In Chapter 2, I describe a simple method for building 'metapeptide' databases directly from metagenomic sequencing reads, with the goal of maximizing coverage of the peptides that may be present in the sample. I show that searching metaproteomic mass spectra against this database provides far higher sensitivity at a controlled false discovery rate (FDR) than searching against public databases or against assembled metaproteomes.

1.2 Proteomics data introspection

Many mass spectrometrists pride themselves on maintaining their instruments assiduously and ensuring the smallest possible error in their measurements of the mass-to-charge ratio (m/z) of each intact peptide and peptide fragment. However, despite this caution, the 'precursor' and 'fragment' mass accuracy of each instrument run may vary significantly from analysis to analysis, and estimates of those two quantities are extremely important parameters for database search. Precursor accuracy determines the list of candidates to be considered for each spectrum, and fragment mass accuracy influences the degree of separation between a correct peptide match and spurious matches.

Furthermore, many important analyses are performed on proteomics data acquired directly from other laboratories or from public repositories. In these cases, there may be no way to obtain a reliable assessment of the precursor and fragment accuracy in order to perform an optimal database search. The practical effect of using suboptimal tolerances for database search can be a reduction in identification sensitivity at a given FDR.

To address this issue in my own analyses, and to allow others to address it in theirs, I developed a tool called Param-Medic that estimates precursor and fragment mass accuracy directly from data and suggests optimal parameter values to use for database search. On a variety of proteomics experiments acquired from public repositories, I demonstrated that

using Param-Medic can result in a dramatic increase in identification sensitivity as compared with using parameter values that are supposedly appropriate for each instrument or even using the values originally used by the researchers who generated the data.

1.3 Deep learning for proteomics

Public proteomics data repositories have been widely used for over a decade, and many journals now require researchers to submit their data to a public repository in order to publish their work. The largest repositories now contain tens of millions of mass spectra from thousands of different experiments on a wide variety of different organisms, tissues and microbial communities.

All these publicly available data, however, have had a relatively small impact on how proteomics research is done: researchers typically do their own database searches against public databases, making no use of the fact that other researchers may have previously acquired and analyzed many very similar spectra representing the same peptides.

To share information among similar mass spectra, I trained a neural network to embed mass spectra into a 32-dimensional space in which spectra generated by the same peptide are close in Euclidean distance. I then embedded millions of mass spectra and detected communities of close-together spectra likely to represent a single peptide. By propagating peptide identifications among the spectra in each community, and by focusing computational resources on identifying representative spectra from each unidentified community, I demonstrated the potential of this embedding to greatly expand the proportion of identified spectra across a public data repository. This approach can provide substantial analytical benefits to every researcher who deposits data in the repository, immediately upon submission, to increase the sensitivity of their proteomics analysis.

Chapter 2

METAPEPTIDES

2.1 Introduction

As the ocean microbial community is the dominant driver of ocean biogeochemical processes such as the carbon cycle, a quantitative understanding of the ocean microbial taxa performing important functions is essential.[1,3,71] Due to culture technique limitations on mixed microbial communities, methods for examining whole microbiomes *in situ* are needed. Metaproteomic analysis of ocean samples has the power to detect peptides from thousands of proteins over a wide range of taxonomic groups within a single analysis.[20,27,57,94] Accordingly, metaproteomics has been used to investigate the functional roles of ocean microbes in a variety of ecological contexts.[20,27,81] However, the success of high-throughput proteomics on ocean samples has been limited by a lack of detection sensitivity.[60]

The majority of organisms active in the ocean microbiome do not have assembled genomes.[73] Public databases can provide partial metaproteome coverage, but without a precise guide to which organisms are present in the sample those databases must be extremely large in order to accommodate as much sequence variation as possible. Searching against such very large databases severely and negatively impacts search sensitivity.[6,63,64] Because of the difficulty of constructing a protein database that accurately reflects an ocean bacterial microbiome, ocean metaproteomics experiments typically only detect a small proportion of the potentially detectable peptides in a sample.[20,34]

As sequencing technologies have become more accessible, "meta-omics" studies have integrated metagenomic, metatranscriptomic and metaproteomic data. For example, databases for metaproteomic search can be constructed using genes predicted from an assembled metagenome.[81] However, this approach can lead to low peptide detection sensitivity for two reasons. First, since many gene fragments present in sequencing reads cannot be reliably assembled into longer contigs, they will be missing from the gene prediction. Second, the process of optimal metagenome assembly requires expertise not necessarily shared by all researchers wishing to do metaproteomics analysis, and if not done optimally the metagenome may fail to contain much of the variation present in the sequencing data. For both of these

reasons, even metaproteomic databases based on site-specific assembled metagenomes tend to provide substantially incomplete coverage of the sample metaproteome.[6,29,93]

An alternative approach takes advantage of the fact that most of the organisms present in many microbiome samples are prokaryotes, and therefore high proportions of their genomes are protein-coding. Tools such as MetaGeneAnnotator[65,66], Orphelia[29] and FragGeneScan[74] predict gene fragments directly from sequencing reads, without assembling the reads into contigs. These approaches can be used to construct metaproteomic databases suitable for database search. As Cantarel et al.[6] demonstrated, these databases enable a greater peptide yield via database search than other methods, with sensitivity greatly dependent on the specifics of the approach to database construction. However, the goal of these tools is sensitive gene prediction rather than peptide detection, and so databases containing translations of their raw gene fragment output can be extremely large. This can lead to impractically long running times for database searches and, more importantly, reduced peptide detection sensitivity.

In the approach described here, we begin with the gene fragments predicted by MetaGeneAnnotator or with six-frame translations of raw reads. We trim and filter these sequences to build a database of "metapeptides": short amino acid sequences derived from open reading frame fragments found in individual reads that are more likely to be identifiable via LC-MS/MS (Figure 2.1A). This approach exploits more of the metagenomic data than an approach based on an assembled metagenome, incorporating reads that fail to be integrated into a contig as well as all of the sample variation for each gene sequence, while avoiding a loss of sensitivity due to over-inclusivity. It is both more complete and more focused on the sample at hand than a strategy based on public databases, potentially including sequences never before observed in any organism and excluding sequences from species not present in the sample.

To evaluate the utility of our metapeptide approach, we compared the sets of peptide sequences detected in two Arctic Ocean microbiome samples at a 1% false discovery rate (FDR) via database search against three different databases (Figure 2.1B): the NCBI non-

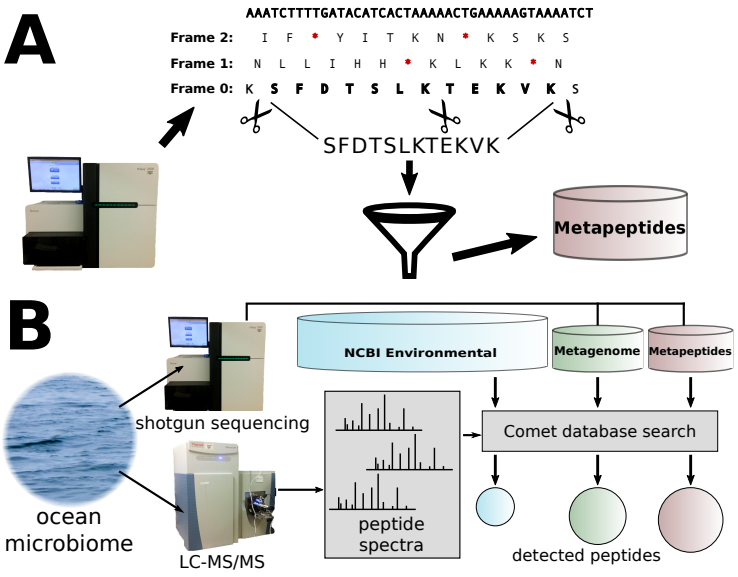

Figure 2.1: **Multiple approaches for metaproteomics of microbiome samples.** A. After high-throughput sequencing, metapeptide database construction begins with six-frame translations of raw sequencing reads, or with gene fragments predicted from reads. Amino acid sequences are trimmed to their outermost tryptic sites to yield metapeptide sequences. Candidate metapeptides are filtered on per-base quality scores, sequence length and other features. Passing candidates are added to the database. B. Alternative proteomics workflows. Microbiome samples are subjected to shotgun metagenomic sequencing and LC-MS/MS analysis. MS/MS spectra are searched with Comet against the NCBI environmental database, against predicted genes from an assembled metagenome, or against a metapeptide database, resulting in peptide yields of different size.

redundant database of environmental protein sequences (env_nr), which is commonly used to interrogate ocean and soil microbiome samples,[55,57] a database derived from a metagenome assembled from shotgun metagenomic sequencing of the two Arctic Ocean samples, and metapeptide databases constructed from the same sequencing reads.

Two microbiome samples were collected from the Arctic Ocean, one sample from the surface chlorophyll maximum layer in the Bering Strait (BSt) and one from bottom waters in the Chuckchi Sea (CS). A total of 2,050 peptides were detected in the BSt sample by searching the environmental database. A metagenome-derived database search yielded 2.12 times as many peptides, and a metapeptide search yielded 3.37 times as many peptides. Results were similar in the CS sample, though with many fewer peptides detected in each search. Integrating the results from all three databases further increased peptide yield.

This substantial advantage in peptide yield contributes greatly to the taxonomic and potential functional classification of the sample proteomes. We used Unipept[52,53] to infer the lowest common ancestor taxon for peptides detected in each search, as well as the list of Gene Ontology (GO) 'biological process' categories associated with proteins containing each peptide. Comparison of the results revealed a much richer taxonomic characterization of the proteins present in the samples from the metapeptide search than from either of the other methods, and a much higher number of detected peptides with the potential for functional annotation. Thus, in addition to dramatically increasing the number of peptides detected in a given ocean sample, the metapeptide-based approach can significantly expand our understanding of the organisms producing the biochemically active molecules in a microbiome. This understanding is crucial to developing a functional model of the microbiome.

2.2 Methods

The data described in the following sections may be downloaded at http://noble.gs.washington.edu/proj/metapeptide.

2.2.1 Experimental methods

Sample collection. Water samples were collected in August of 2013 from the Bering Strait (BSt) chlorophyll maximum layer (7m depth, 65° 43.44" N, 168° 57.42" W) and from the more northern Chukchi Sea (CS) bottom waters (55.5 m depth, 72° 47.624" N, 16° 53.89" W) using a 24-bottle CTD (conductivity, temperature and depth) rosette (10 L General Oceanics Niskin X). The integrated water column Chlorophyll-a measurement was 226.88 mg/m^2 at station BSt and 2.64 mg/m^2 at station CS. As our previous work has shown, to examine bacterial contributions it is essential to remove the very high background contribution from algal inhabitants.[56] Also, oceanic marine bacteria are typically smaller than bacteria in gut biomes or freshwater systems, with the majority passing a 1.0 μm filter.[41,61] Accordingly, a 15-liter water sample was prefiltered through 10 μm and then 1 μm high-volume cartridges to remove larger eukaryotes, and the filtrate comprising the bacterial microbiome was then collected on a glass fiber filter (GF/F) with nominal pore size of 0.7 μm. Filters were flash frozen and stored at -80°C until extraction.

Metagenome DNA extraction, library preparation and sequencing. Filters were sliced and DNA extraction was accomplished using the protocol developed for planktonic biomass on Sterivex filters, as described in Wright et al.[90] Briefly, DNA was extracted from the collected cells using phenol:chloroform and chloroform extractions. DNA was then purified using a cesium chloride density gradient. Extracted DNA was sheared to $<$ 1 kb and excess salts were cleaned up using Agencourt AMPure XP purification (Beckman Coulter, Brea, CA). Library preparation was done with the Kapa Hyper Kit, following the manufacturer's instructions (Kapa Biosystems, Wilmington, MA), and library quality was confirmed using the Bioanalyzer (Agilent, Santa Clara, CA). Libraries were sequenced in one lane on an Illumina HiSeq. The resulting 100 base pair, paired-end sequencing reads were trimmed and filtered using SolexaQA,[9] with a minimum Phred quality score[16] of 20 on any base.

Protein sample preparation and tandem mass spectrometry (LC-MS/MS). GF/F filters with the bacterial fraction were placed in 1.5 mL tubes with 100 μl of 0.5mm glass beads, 100 μl 6M urea and 500 μl nanopure water. Filters were shaken on a bead beater for one minute and then placed in ice for five minutes. This process was repeated 10 times to ensure cell lysis and filter breakup. A needle was then heated by flame and used to create a <0.5mm hole at the bottom of the 1.5mL sample tube. The sample tubes were then placed atop an open 1.5mL tube and centrifuged (3000 x g, 10 minutes). This process was completed to isolate protein lysate from extracted particles and glass beads. Protein concentrations were determined using BCA colormetric assay; 100 μg of total protein was used for digestion. Each 100 μg protein sample received 300 ng purified human ApoA1 to monitor protein digesion. Samples were reduced, alkylated, enzymatically digested with trypsin and desalted following Nunn et al.[67] Prior to MS injections, 50 fmol of the Pierce Peptide Retention Time Standard (ThermoFisher Scientific) was added to each autosample vial at 50 fmol per 2 μg total protein. Peptides were separated using an inline NanoAquity HPLC with a 4 cm pre-column (5 μm; 200A; Magic C18) and 30 cm Reprosil-Pur Basic 3 μm C18 analytical column (Dr. Maisch GmbH, Germany). Peptides were eluted using a 2-30% ACN, 0.1% formic acid non-linear gradient in 120 minutes at 300 nl/min. LC-MS/MS was performed with a Q-Exactive-HF (ThermoScientific) on technical triplicates for each sample. Instrument was operated in Top 20 data-dependent acquisition mode, collecting data on 400-1600 m/z range with a 5 s dynamic exclusion.

2.2.2 Computational methods

All computation was performed on a Univa Grid Engine cluster with 1.90GHz AMD Opteron processors.

Gene prediction from shotgun sequencing with existing methods. The MOCAT pipeline[42] was used to assemble a metagenome and predict genes as follows. Trimmed and filtered reads from both BSt and CS samples were aligned to the human hg19 reference using

SOAPaligner v2.21, and aligned reads were removed. The remaining reads were assembled into contigs and scaftigs with SOAPdenovo v1.06. The assembly was revised, correcting for indels and chimeric regions, with SOAPdenovo v1.06 and BWA v0.7.5a-r16. Genes were predicted using Prodigal v2.60.

We used three well-established gene fragment prediction tools, MetaGeneAnnotator (in multiple species mode), FragGeneScan version 1.2.0 (illumina_10 model parameters) and Orphelia (with Net300 prediction model) to predict gene fragments directly from shotgun metagenomic sequencing reads from each sample.

Metapeptide databases. Separate metapeptide databases were constructed from the BSt and CS sequencing runs, either from predicted gene fragments or from raw read sequences. When starting from raw read sequences, each read was translated in all six reading frames, and reading frames containing a stop codon were discarded. The results described in section 2.3 were obtained by starting with predicted gene fragments from MetaGeneAnnotator.

Whether starting from gene fragments or from raw read sequences, amino acid sequences from each nucleotide sequence were trimmed to the first and last tryptic cleavage site (or discarded if fewer than two sites), and the remaining ends discarded (Figure 2.1A). This was done in order to remove partial tryptic peptide sequences that are unlikely to be detected by LC-MS/MS of a trypsinized metaproteome. The resulting candidate sequences were discarded if they were less than 10 amino acids long, if they contained no tryptic peptides with seven or more amino acids, or if the minimum Phred quality score over the length of the sequence was less than 30. Finally, metapeptide candidates meeting all the above criteria were discarded if they were represented by fewer than two reads. A FASTA database was constructed from the remaining metapeptides.

For purposes of comparison, we also made use of a metagenome-derived database of translated genes from the metagenome described above and the NCBI non-redundant database of protein sequences from large environmental sequencing projects ('env_nr', downloaded from ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/env_nr.gz on December 1, 2015).

Database search. All database searches were performed using Comet[14] version 2015.01 rev. 2, using a concatenated decoy database in which peptide sequences were reversed but C-terminal amino acids left in place. Search parameters included a static modification for cysteine carbamidomethylation (57.021464) and a variable modification for methionine oxidation (15.9949). Enzyme specificity was trypsin, with one missed cleavage allowed. Parent ion mass tolerance was set to 10ppm around five isotopic peaks, and fragment ion binning was 0.02, with offset 0.0. Peptide-spectrum matches (PSMs) from all technical replicates were combined into a single dataset. As described previously,[22] after each unique peptide was associated with its top-scoring spectrum, irrespective of charge state, we used the widely-used target-decoy search strategy of estimating the false discovery rate (FDR) associated with a given set of accepted peptides.[13] In this context, the FDR is defined as the proportion of the accepted peptides that are not responsible for generating observed spectra. We then empirically examined the trade-off between FDR and the number of accepted peptides, since in practice the mass spectrometrist is typically interested in accepting as many peptides as possible while maintaining an acceptable FDR. Note that this trade-off is similar to the distinction between precision (1 - FDR) and recall or sensitivity.

Results of searches of individual samples against multiple databases were integrated as follows. PSMs from searches against all databases were combined into a single tab-delimited file of features for input to Percolator.[32] For each database, a new binary feature was added to the combined feature file indicating whether the PSM was derived from a search against that database. Percolator was then used to analyze the combined set, thereby computing a discriminant score for each PSM. For each scan with multiple PSMs (from multiple databases), all but the highest-scoring PSM were removed. Peptide-level FDR was then calculated as described above.

Taxonomic and functional inference. Detected peptides were given taxonomic assignments by Unipept version 1.1.0. For all tryptic peptides with no missed cleavages present in UniProtKB, Unipept assigns a lowest common ancestor (LCA) taxon from the NCBI Tax-

onomy Database, the most-granular taxon common to all organisms containing the peptide. For peptides with missed tryptic cleavages, Unipept calculates an LCA based on the LCAs associated with all completely-cleaved peptide sequences contained in the peptide.

Since no such standard methods currently exist for assigning functional annotations to detected peptide sequences, we estimated the maximum number of peptides that could be assigned functional annotations. We used Unipept to retrieve all of the proteins containing each detected peptide, along with their GO category annotations. GO annotations are divided into three namespaces: 'biological process', 'molecular function' and 'cellular compartment'. We declared a peptide to be potentially functionally informative if at least one protein containing it was annotated with at least one GO category in the 'biological process' namespace.

2.3 Results

2.3.1 Gene fragment predictions from deep shotgun metagenomics sequencing are not directly usable for proteomics database search.

Shotgun sequencing of the BSt and CS samples generated 171 million and 245 million reads, respectively. We evaluated three different gene prediction tools, FragGeneScan, Orphelia and MetaGeneAnnotator. None of these tools were originally developed for this high depth of coverage, nor have they been updated to accommodate high-depth sequencing. On the BSt reads, MetaGeneAnnotator ran to completion in 3.5 hours, producing 133 million fragments. Orphelia quickly exceeded 100GB of memory usage; as its output on smaller inputs was 33 times the size of the output of MetaGene, with no scoring mechanism to use for filtering, we decided not to pursue it further. After five days of running time, FragGeneScan had not yet completed, and its output on smaller inputs was 32 times the size of the output of MetaGene, so we decided not to pursue it further.

The 133 million fragments produced by MetaGeneAnnotator contained 222 million unique peptides and require more than four days to search one replicate against. However, 177

million of the peptides in the database represented ragged ends of peptides terminating at the beginning or end of a metagenomic sequencing read, and did not represent a detectable tryptic peptide.

2.3.2 Environmental and assembled metagenome databases provide incomplete coverage of peptides in ocean samples.

Next, we quantified the extent to which a public database and a metagenome-derived database could be used to detect the peptides present in the two ocean microbiome samples. The environmental and metagenome databases contained 119 million and 11 million peptides, respectively. All three replicates of each sample were searched against both databases, and the set of peptides detected with FDR < 0.01 in searches against each database was determined with Percolator, as described above (Figure 2.2).

editFor the BSt sample, of the 4,344 peptides present in the metagenome database and detected in the metagenome search, 61.2% did not occur in the environmental database; similarly, 46.4% of the 2,050 peptides present in the environmental database and detected in the environmental search were absent from the metapeptide database. This high complementarity indicates that large numbers of peptides present in the sample are absent from each database. Furthermore, of the 1,708 peptides present in both databases (and therefore potentially detectable by either search) and detected in one or both searches, only 1.3% were detected in the environmental database search but not in the metapeptide database search. By contrast, 35.7% were detected in the metapeptide search but not in the environmental search. Since those peptides were present in the environmental database, we conclude that the failure to detect them is due to a loss of statistical power stemming from the much larger size of the environmental database.

2.3.3 Searching metapeptide databases increases peptide yield, enriches taxonomic and functional characterization.

Next, we evaluated the ability of our metapeptide databases to increase peptide detection sensitivity relative to the environmental and metagenome databases. The metapeptide databases constructed from the shotgun metagenomic sequencing reads from the BSt and CS samples contained 12 million and 14 million peptides, respectively. The BSt metapeptide database (12 million peptides) contained 2 million peptides in common with the environmental database (129 million peptides) and 4 million in common with the metagenome database (11 million peptides). All MS/MS replicates of the BSt and CS ocean microbiome samples were searched against the metapeptide database constructed from the sample being searched, and the set of peptides detected with FDR < 0.01 was derived with Percolator. In the BSt and CS samples, the numbers of peptides detected were 1.59 and 1.92 times the number detected by searching against the metagenome-derived database, and 3.37 times and 2.74 times the number detected by searching against the environmental database, respectively (Figure 2.2).

To determine the reasons for this larger peptide yield, we compared the sets of peptides detected by searching the BSt spectra against the metapeptide and environmental databases (Figure 2.3). Of the 6,918 peptides present in the metapeptide database and detected in the metapeptide search, 67.8% did not occur in the environmental database; by contrast, only 27.5% of the 2,050 peptides present in the environmental database and detected in the environmental search were absent from the metapeptide database. This discrepancy suggests that the metapeptide database contains more of the peptides present in the sample. Furthermore, of the 2,261 peptides present in both databases and detected in one or both searches, only 1.5% were detected in the environmental database search but not in the metapeptide database search. By contrast, 34.2% were detected in the metapeptide search but not in the environmental search. Since those peptides were present in the environmental database, we conclude that the failure to detect them is due to a loss of statistical power stemming from

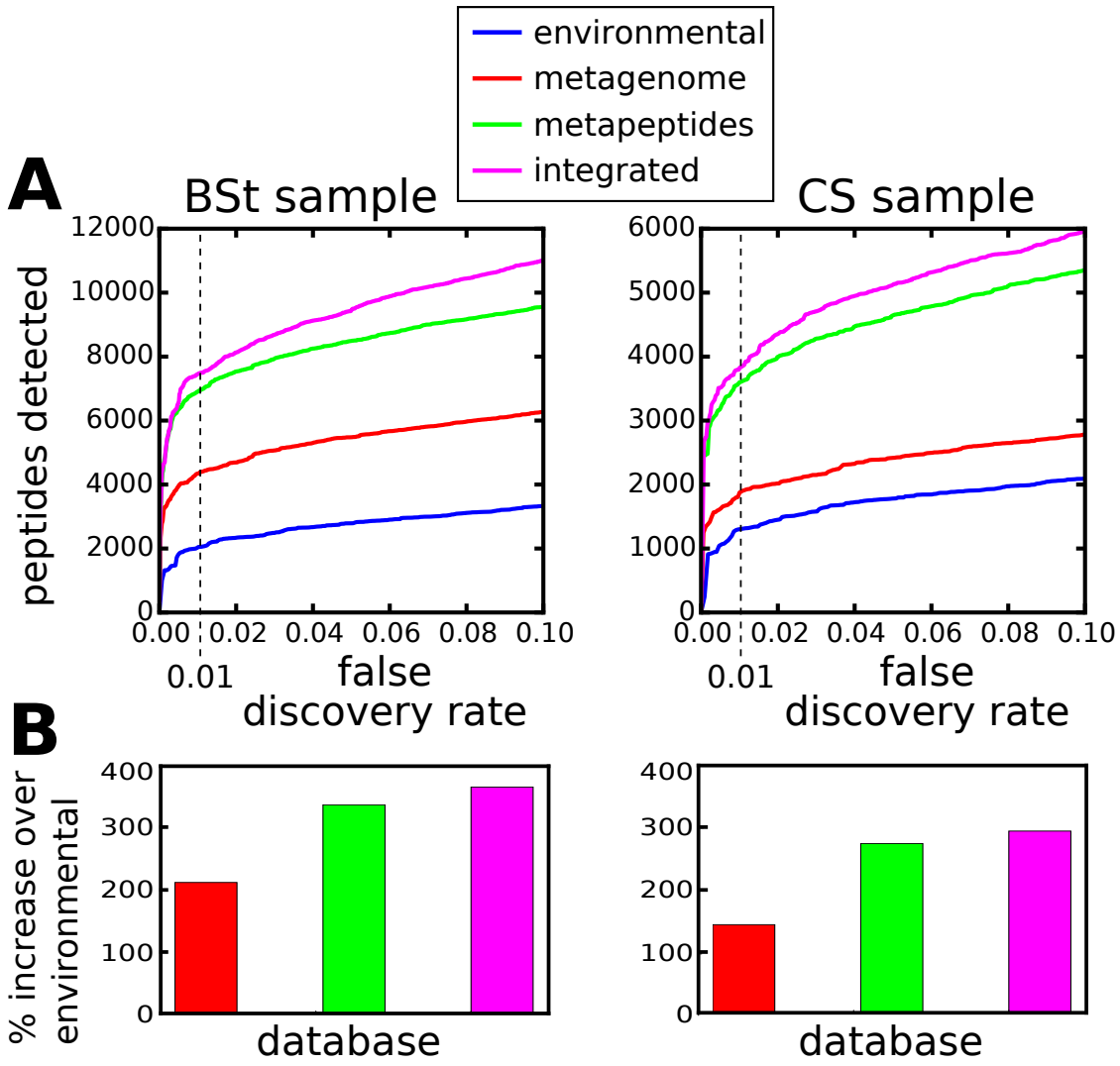

Figure 2.2: **Peptides detected in different searches.** A. Line plots of peptide FDR (horizontal axis) *vs.* number of peptide sequences detected at that FDR in the Bering Strait (BSt) and Chukchi Sea (CS) samples (vertical axis), when searched four different ways. B. Detected peptide counts at FDR < 0.01 in the metagenome, metapeptide and integrated database searches, as a percentage of the counts detected in environmental search.

the much larger size of the environmental database. We also compared the sets of peptides detected by searching the BSt spectra against the metapeptide and metagenome databases, which are of much more similar size. Of the 6,918 peptides present in the metapeptide database and detected in the metapeptide search, 41.9% did not occur in the environmental database; by contrast, only 7.9% of the 4,344 peptides present in the metagenome database and detected in the metagenome search were absent from the metapeptide database, suggesting more complete sample coverage in the metapeptide database. Furthermore, of the 4,250 peptides present in both databases and detected in one or both searches, 5.3% were detected in the metagenome search but not in the metapeptide database search, while 5.8% were detected in the metapeptide search but not in the metagenome search. This suggests that the metapeptide database advantage is essentially due to greater coverage, while the searches of the two similarly-sized databases are roughly equally sensitive.

By themselves, detected peptide sequences provide limited information about a sample. However, the peptides can be used to provide important insight into the sample's community composition. Accordingly, we assessed the extent to which the additional peptides detected using the metapeptide database can enrich the taxonomic and functional classification of the metaproteome.

For taxonomic inference, we used the Unipept tool to assign a least common ancestor (LCA) taxon to all possible peptides detected in a search of the BSt sample replicates against a given database. The metapeptide search detected 1.28 times as many peptides that were assigned LCAs as the metagenome-derived database search, and 1.76 times as many as the environmental database search. At every taxonomic rank more granular than class, the highest number of taxa were detected by the integrated search, followed by the metapeptide, metagenome and then environmental searches (Figure 2.4). The same order was observed when examining the number of peptides with an LCA at each taxonomic rank. As a side note, the number of peptides detected by a given database search with an LCA at a given rank decreases monotonically with the granularity of the rank, which is a reflection of the LCA ranks of detectable peptides, as a whole, as determined by Unipept based on

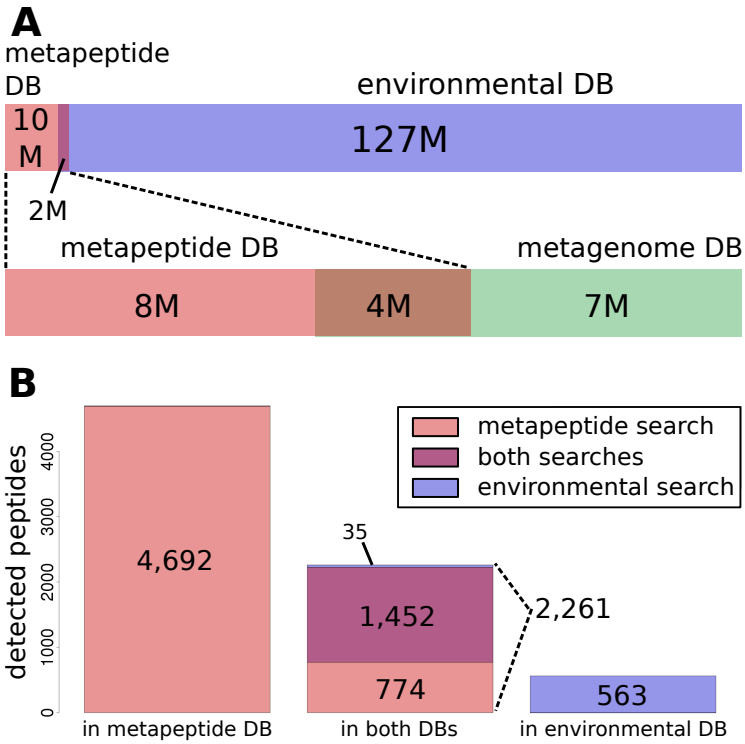

Figure 2.3: **Database and detected peptide comparisons.** A. The BSt metapeptide database contains roughly 12 million tryptic peptides. The environmental database contains 129 million, with an intersection between the two databases of 2 million peptides. The metagenome database contains 11 million peptides, with 4 million peptides in common with the BSt metapeptide database. B. Searching against the BSt metapeptide database detects 6,918 unique peptides at FDR < 0.01, *vs.* 2,050 when searching against the environmental database, with 1,452 in common. Of the 2,261 peptides detected in either search that were present in both databases, 1,452 were detected in both searches, 774 were only detected in the BSt metapeptide database search and 35 were only detected in the environmental database search.

UniProt annotations. The number of taxa detected increases with rank granularity until the rank of species, which shows a modest decline from the rank of genus. This relationship is also a function of the LCA ranks of detectible peptides and does not reflect any particular characteristics of the various searches.

Because many metapeptides are likely from unsequenced microbes not present in public protein databases (and therefore uninformative to Unipept), an important question is whether the detected peptides that were present in the metapeptide database but absent from the environmental database conferred any taxonomic information via this method. Considering the peptides detected in the metapeptide database search, the percentage of peptides that are assignable to an LCA by Unipept is much greater for the subset of those peptides that are present in the environmental database (70.8%) than for the subset that are absent (16.3%). However, because 67.8% of the peptides detected in the metapeptide database search are absent from the environmental database, in absolute terms 32.6% of the peptides assignable to LCAs come from that latter group. Thus, both the greater peptide detection sensitivity and the greater peptide coverage afforded by the metapeptide database contribute to its increased potential for metaproteome taxonomic classification.

To estimate the number of peptides with potential functional annotations, we used Unipept to locate all of the proteins containing each detected peptide, along with their associated GO categories. The metapeptide search detected 1.50 times as many peptides associated with one or more GO categories in the 'biological process' namespace as the metagenome-derived database search, and 2.12 as many as the environmental database search.

2.3.4 Combining results from multiple databases further increases peptide coverage

Although the metapeptide databases are the most valuable individual databases for searching these samples, a higher overall peptide yield can be obtained by combining results from multiple databases. PSMs from searches against the environmental, metagenome and metapeptide databases were integrated as described above. In the BSt and CS samples, respectively, 1.09 times and 1.07 times as many peptides were detected by this method as by searching against

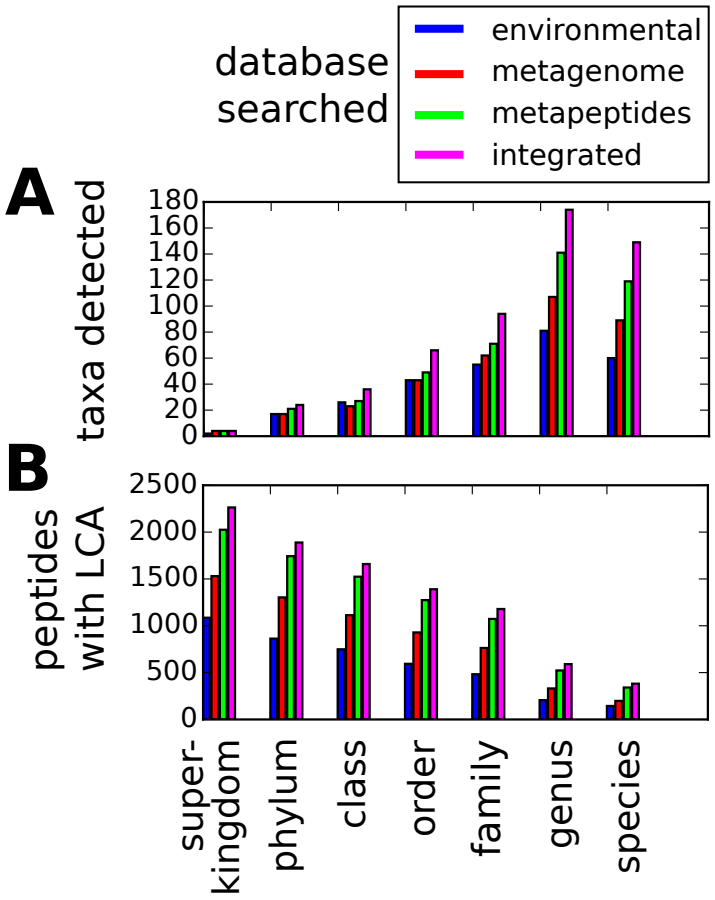

Figure 2.4: **Taxonomic inference summary.** Bar charts comparing the taxonomic information derived from four different searches of the BSt sample: against the NCBI environmental database, against the metagenome-derived database, against site-specific metapeptides, and integrating results from all three databases. A. Counts of taxa detected, by rank from superkingdom to species. B. Counts of peptides associated with an LCA at each rank.

the individual metapeptide databases (Figure 2.2). For context, the largest set of peptides that could possibly be detected at FDR 0.01 by any method combining these three searches is the union of all peptides detected at FDR 0.01 in each of the three separate database searches of each sample. This number was 1.16 times and 1.18 times the number of peptides detected via metapeptide searches of the two samples, respectively. We determined that this modest increase was statistically significant via an analysis of the three technical replicates for each sample. In the metapeptide database searches, technical replicates of the BSt and CS samples detected a mean of and 4,691.3 and 2,991.6 peptides, respectively, with a mean pairwise intersection of 3,075.0 and 2,377.7 peptides between replicates, respectively. The per-replicate means for the BSt and CS samples in the integrated searches were 5,090.0 and 3,193.0, respectively, and for each sample the increase was statistically significant by both one-tailed and two-tailed paired t-tests, at $p < 0.05$.

In terms of taxonomic inference, the integrated searches of the BSt and CS samples detected 1.13 and 1.12 times as many peptides assignable to LCAs compared with a metapeptide search (Figure 2.4 for BSt comparison), with more taxa observed at every taxonomic rank lower than superkingdom.

This method of integrating database search results yielded 14.2% more peptides at FDR < 0.01 as searching a concatenated database combining the environmental, metagenome and metapeptide databases. Due to the reduced statistical power of a search against a larger database, searching the concatenated database yielded 5.0% fewer peptides than searching the metapeptide database alone. The extra Percolator features representing the database against which each PSM was made were of modest benefit, increasing peptide yield by 4.3% *vs.* an integrated search with those features removed.

2.3.5 Metapeptide databases from two microbiome samples can be used to interrogate each other.

Constructing a metapeptide database is a relatively expensive endeavor, requiring library preparation, short read sequencing and computational time, and so it would be convenient

to use a single database to interrogate the metaproteome from multiple samples. Our two samples are from two different locations and from two very different positions in the water column (chlorophyll maximum layer and bottom water). In each case, overall peptide yield from a database search against the metapeptide database derived from the other sample was a large improvement over the yield from a search against the environmental database (2.17 and 1.95 times as many peptides, respectively). In each case, however, searching a sample against its site-specific metapeptide database detected many more peptides than searching against the database derived from the other sample. Notably, the BSt sample appeared to benefit greatly from a search against the BSt database rather than against the CS database (1.55 times as many peptides), while the effect in the opposite direction was not as pronounced (1.40 times as many). A potential explanation for this difference lies in the depth from which the two samples were taken: the BSt sample, from the upper water column, is expected to contain more biodiversity than the CS sample taken from the bottom layer, which has no light.

2.3.6 Filtering protocols are critical to resulting metapeptide database size.

Prior to filtering, the trimmed MetaGene output contained 51 million tryptic peptides. To investigate the effects of the filtering criteria, we systematically varied each parameter while leaving the remaining parameters set to the values described in Section 2.2.2. The results (Figure 2.5) demonstrate that filtering metapeptides based on the support of two or more reads and the use of MetaGeneAnnotator fragments rather than six-frame translations of raw reads had particularly large effects on database size, reducing the number of unique tryptic peptides by 74.0% and 51.8%, respectively, and decreasing search time by similar proportions.

To investigate the effect of database filtering parameters on peptide detection sensitivity, we generated a small sample set of 24,000 MS/MS spectra from the BSt sample (8,000 random spectra from each replicate run) to compare the number of peptides detected at FDR < 0.01 by searching each database. Beginning with MetaGeneAnnotator fragments rather than with

a six-frame translation of raw reads increased detected peptides by 9.0%, demonstrating that the MetaGeneAnnotator is valuable but not crucial to the metapeptide strategy. The MetaGeneAnnotator score was not useful as a filtering criterion: higher score thresholds resulted in monotonically lower peptide yield. Requiring two or more reads increased detected peptides by 8.4%. Higher read count thresholds monotonically reduced yield. Sufficiently restrictive values for each parameter reduce peptide yield much more severely (data not shown). However, in general, within the range of values shown in Figure 2.5 the reduction in yield was minor, suggesting a relative robustness of the parameter settings.

2.4 Discussion

In this work, we have demonstrated the value of interrogating microbial metaproteomes by constructing metapeptide databases from site-specific shotgun metagenomic sequencing reads. These databases afford much greater peptide detection sensitivity than the NCBI environmental database or a database of genes predicted from an assembled metagenome. Furthermore, we have shown that a database derived from one sample can be used to interrogate another sample from a different location and position in the water column. By combining metapeptide databases from a variety of samples, sequencing efforts could potentially be centralized to an extent, and metapeptide databases integrated into existing metaproteomics workflows such as the MetaProteomeAnalyzer.[59] In principle, these methods should be applicable to other microbiomes, such as riverine and soil-derived microbial communities, in which prokaryotes dominate the microbiome and the great majority of organisms are unsequenced. These methods may also provide additional sensitivity in a better-understood environment such as the human gut microbiome.

From a practical standpoint, the larger environmental database required more computational time for database search than the metapeptide databases, making searching it a bit of a boondoggle. On our hardware, the three BSt sample replicates, with an average of 104,000 MS/MS scans, took us an average of 1.15 hours to search against the BSt database and an average of 14 hours to search against the environmental database. The much larger

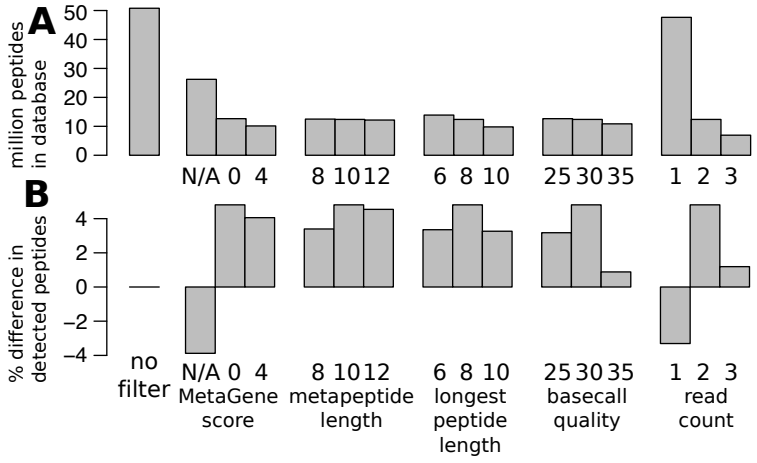

Figure 2.5: **Metapeptide parameter comparison.** Comparisons between metapeptide databases constructed with different filtering parameters. The bars on the far left represent a database of MetaGene fragments trimmed to outermost tryptic ends but otherwise unfiltered. Each group of three bars represents three different values for a single parameter, with all other parameters set as described in Section 2.2.2. In each group, the middle value represents the value used to generate the results described in Figures 2.2-2.4. The N/A value for MetaGene score represents the use of raw six-frame translations of reads instead of MetaGene output. A. Millions of tryptic peptides in each database. B. Percent difference in counts of peptides detected in a search of 24,000 scans from the three BSt sample replicates at FDR < 0.01 against each database, as compared with search against the unfiltered, trimmed MetaGene database.

raw database of MetaGeneAnnotator fragments took more than four days to search. The strategy of trimming reads to the outermost tryptic sites and the filtering criteria applied are responsible for the much smaller metapeptide database size, making the metapeptide database easier to integrate into a proteomics pipeline.

De novo sequencing is another strategy for increasing peptide detection sensitivity. Although this method can yield many confident partial peptide sequences, it is less effective at confidently detecting full-length peptides. Furthermore, due to codon degeneracy, *de novo* sequencing also cannot easily link detected peptides with their corresponding nucleotide sequences for taxonomic annotation. As others have noted, the space of peptides likely to be present in a metaproteomics sample should remain tractable to a database search approach if search databases are constructed with an eye toward detection sensitivity.[75] However, *de novo* sequencing remains a viable approach for assignment of spectra that cannot be assigned with database search.

Some of the peptide sequences detected by metapeptide database search are present in organisms with publicly available genomes, enabling putative taxonomic assignment using existing peptide-based tools and enriching taxonomic characterization. However, a large proportion of peptides detected by searching against metapeptide databases have never been reported in an assembled genome. In future work, we will place those peptides within a taxonomic hierarchy using sequence homology. This may be accomplished using all of the nucleotide sequences of the reads that contributed to the inclusion of each metapeptide in the database.

Sequence homology could also be used to infer the putative function of proteins containing these detected peptides. With both taxonomic and functional assignments, a large number of detected peptides could be used in comparisons of the activity of various microbes between samples. This research will quantify the protein functions responsible for chemical transformations at meaningful taxonomic levels, thereby exposing microbial ecosystems at the molecular level to improve our understanding of their interactions and biological roles. Applying this approach in conjunction with recent advances in quantitative proteomics can

bring about a fundamental change in how we view, analyze, and model microbial ecosystems.

An important area for future research lies in the development of improved methods for combining search results from multiple databases. The approach we have adopted here relies upon the machine learning algorithm Percolator to calibrate scores between the different database searches. A more powerful approach might be to adopt a strategy similar to cascade search,[37] searching against, in order, the metapeptide, metagenome and environmental databases. In future work, we plan to develop and validate a statistical method for combining cascade search with a machine learning post-processing step like Percolator.

The software tools described here have been implemented in Python 2.7. The software (including source code) and data described in this manuscript may be downloaded at http://noble.gs.washington.edu/proj/metapeptide.

Chapter 3

PARAM-MEDIC

3.1 Introduction

Database search algorithms such as Sequest[15] serve as the core of many shotgun analysis pipelines. Most search engines require a long list of user-supplied parameters, including cleavage enzyme, number of missed cleavages to allow, static and variable peptide modifications, and tolerances to use in matching observed precursor and fragment masses to their theoretical counterparts. Appropriate values for these parameters depend on the instrument used, the instrument settings used for a particular analysis, instrument performance at the time of acquisition, and other factors.

In this work, we focus on two of the most important search algorithm parameters. Precursor mass tolerance defines the peptide candidates considered for each spectrum. A narrower setting reduces the running time of the search algorithm by requiring it to perform fewer comparisons between peptides and spectra, but a too-narrow setting can exclude true matches. Too wide a setting can reduce sensitivity in a different way: as more candidates are considered for each spectrum, the chance of a false match randomly generating a higher score than a true match increases[62]. Similarly, fragment mass tolerance or bin size determines how small the absolute value of the difference between a pair of observed and theoretical fragment masses must be in order to consider them a match. A tighter setting can exclude true matches between fragments, while a loose setting can lead to false matches between fragments, leading to more high-scoring false matches.

An important goal of many proteomics workflows is to achieve high statistical power for peptide detection. A commonly-used proxy for the peptide detection power of a database search is the number, or "yield," of peptide-spectrum matches (PSMs)at a set false discovery rate (FDR) such as 0.01, as estimated by target-decoy procedure[13]. We define the optimal value for precursor or fragment mass tolerance as the value that yields the most PSMs at FDR 0.01. The optimal value for either parameter may vary widely from experiment to experiment. This sensitivity to parameter settings has a real impact on experimental results, because the measurement of yield can vary greatly between the best and the worst parameter

settings.

Researchers adopt different strategies to arrive at the settings they use for a given analysis. Some labs fine-tune the optimal settings for a particular instrument by performing searches on acquired data with many different settings. Because instrument performance can change over time to cause drift in both mass accuracy and calibration, researchers most concerned with using the proper settings will periodically perform measurements solely to reassess performance. On the other extreme, database searches are often performed by researchers other than those who ran the instrument, as when labs share data or when spectra are reanalyzed after being deposited in a public repository. In the absence of detailed information about how the instrument was run or how well it was performing at that time, researchers typically rely on instrument settings reported by the lab that ran the instrument or on the advertised capabilities of the instrument that was used,

Several tools have been developed to aid researchers in selecting optimal search parameter values. Many of these tools infer instrument calibration from experimental data by analyzing the observed m/z values of known ions: either spiked-in peptides or peaks confidently identified by database search.[38,47,58,70]. One such tool for the Windows platform, Preview,[38] additionally assesses precursor and fragment mass error, nonspecific digestion, and sample modifications using a fast database search. However, neither Preview nor any of the other tools we surveyed provides a well-defined method for translating assessed m/z error into parameter settings for database search.

Here we describe Param-Medic: an open-source, cross-platform tool for assessing experimental m/z error and deriving parameters to search an LC-MS/MS experiment. We have trained Param-Medic to produce parameters appropriate for the Comet[14] search engine, but the same strategy could be extended to work for any algorithm. At the heart of Param-Medic is a key assumption that despite the use of so-called "dynamic exclusion" rules, LC-MS/MS experiments typically make multiple observations of many individual peptide ions. Param-Medic exploits these repeated observations to enable estimation of m/z error. Specifically, the algorithm assesses measurement precision (but not calibration) by identifying pairs of

spectra likely to represent the same peptide and then analyzing the distribution of differences between those pairs' precursor and matched fragment ion *m/z* values. We trained Param-Medic on eight datasets from public repositories from a variety of organisms and instruments. We evaluated its performance on three additional public datasets from the same instruments, as well as on a dataset generated using a very different Q-TOF instrument. Param-Medic is available as a standalone tool and as a part of the Crux proteomics toolkit, providing an open, integrated platform for parameter inference and database search.

3.2 Methods

3.2.1 Mass-to-charge error estimation

Param-Medic infers both precursor and fragment m/z search parameters in a four-step procedure (Figure 3.1). First, it pairs closely-eluting MS/MS spectra that have similar precursor and fragment m/z values. Then, it calculates the mass differences of both the paired precursors and the paired fragments. Next, it fits a separate mixed Gaussian-Uniform distribution to the error values for precursors and for fragments. Finally, it maps the standard deviation of each estimated Gaussian distribution to a value usable as a precursor tolerance or fragment bin size for database search.

Param-Medic begins by assembling pairs of measurements from spectra with an inferred charge of 2 that appear likely to represent the same precursor ion or fragment ion (Figure 3.1). Spectra are paired permissively in order to generate distributions of pairwise measurement differences with sufficient numbers of both correctly and incorrectly paired spectra so that the two component distributions can be estimated. Precursor and fragment masses are calculated from their observed *m/z* values and are each binned coarsely with bin size 1.0005079, corresponding to the distance between the centers of two adjacent peptide mass clusters.[89] One list of paired measurements is initialized for precursor values, and another for fragments.

As Param-Medic processes each sequential MS/MS scan, the algorithm identifies the previous MS/MS scan within the last 1000 scans whose precursor falls in the same bin (if

any). It then checks whether the associated precursor m/z is within 50 parts per million (ppm) of the precursor m/z of the new scan and whether at least 20 of the 40 most-intense binned fragments are unambiguously shared between the two spectra. If both conditions are met, then the two spectra are considered to represent the same peptide ion. In this case, the two precursor m/z values and the paired values for the five most-intense pairs of fragment m/z values are added to their respective lists. No single spectrum is included in more than one such pair, and additional measurements of the same ion are paired rather than being assembled into higher-order tuples. If Param-Medic detects fewer than 200 such pairs (an arbitrary threshold that may be adjusted as desired), then the program will terminate without estimating parameter settings.

In the second step, the ppm differences in measurement pairs are calculated from the pairs of measurements. This step and the following steps are performed separately but identically for precursor pairs and for fragment pairs. The output of this step is an empirical list of ppm differences in paired peak measurements. In practice, this list represents a mixture of differences between two correctly-paired measurements of the same peak and differences between two incorrectly-paired measurements of peaks that represent different ions. Below, we refer to these as "true" and "false" pairs, respectively.

In the third step, Param-Medic fits a theoretical distribution to the empirical distribution of errors from step two. Param-Medic assumes that ppm measurement error for true pairs is normally distributed. Therefore, the difference between two values drawn from the distribution of ppm measurement error is also normally distributed, with variance twice that of the measurement error. Param-Medic also assumes that differences between false pairs are uniformly distributed over the range considered. Accordingly, it models the distribution of measurement differences as a mixed Gaussian ($\mathcal{N}(y)$ for observed differences y) and uniform distribution. Expectation-maximization (EM) is used to estimate three parameters: the mean and standard deviation of the Gaussian distribution component ($\hat{\mu}_\delta$ and $\hat{\sigma}_\delta$), and the probability of membership in the Gaussian distribution (p_G). EM maximizes the

log-likelihood of the observed data:

$$\sum_{i=1}^{n} \ln \left(p_G \mathcal{N}(y_i; \mu_\delta, \sigma_\delta) + (1 - p_G) \frac{1}{b - a} \right) \tag{3.1}$$

The algorithm alternates between an E step, which estimates expectation of the log-likelihood using the current parameter estimates, and an M step, which computes new parameter values maximizing the expected log-likelihood. Once $\hat{\sigma}_\delta$ is estimated, the standard deviation of the measurement error, $\hat{\sigma}_\epsilon$, is estimated as $\hat{\sigma}_\epsilon = \frac{\hat{\sigma}_\delta}{\sqrt{2}}$.

In the final step, having estimated the standard deviation of the ppm error distributions, Param-Medic applies a scaling factor to $\hat{\sigma}_\epsilon$ to calculate the estimated optimal search parameter (either precursor tolerance or fragment bin size). This scaling factor is empirically estimated on an analysis of data from a wide variety of mass spectrometry experiments, as described in the following sections.

Many of Param-Medic's parameters are adjustable. The values mentioned above for the charge state (2), wide ppm tolerance (50 ppm), number of peaks that must be shared between spectrum pairs (20 of the most-intense 40), number of fragments per pair used for estimation (5), number of difference measurements required for estimation (200), and maximum scan distance between spectrum pairs (1000) are defaults that should be widely applicable but may be adjusted for unusual datasets. For example, a user may wish to choose a higher charge state when analyzing an experiment on tryptic peptides known to contain a very high proportion of missed tryptic cleavages, or to remove the maximum scan distance constraint altogether for very long gradients.

3.2.2 Search of public datasets with different parameter values

For use in learning the scaling factors mapping $\hat{\sigma}_\delta$ to search parameter values, we collected eight training and three test datasets from the PRIDE[86] and Chorus Project (http://chorusproject.org) proteomics data repositories, representing a variety of organisms and instruments (Table 4.1). All database searches were performed using Comet[14] version 2015.01 rev. 2. Samples were searched against the appropriate UniProt databases for

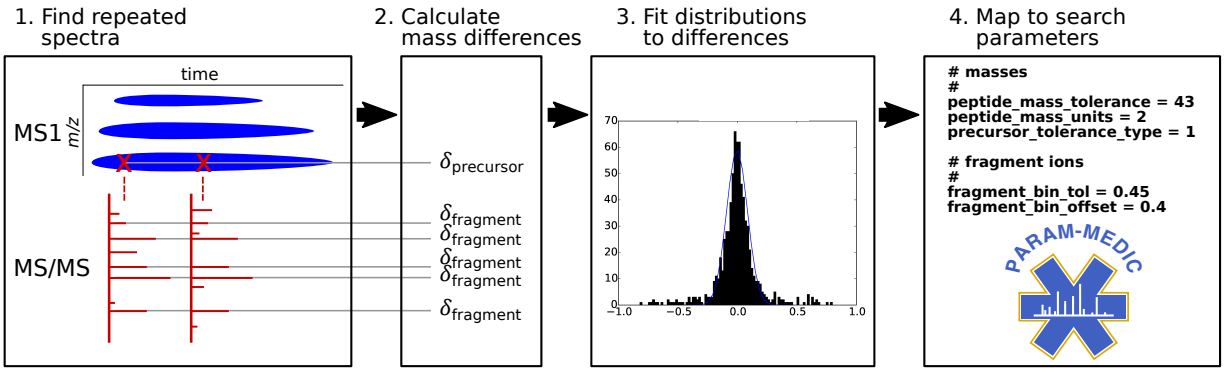

Figure 3.1: **Param-Medic workflow.** Param-Medic collects pairs of closely-eluting MS/MS spectra and assembles their pairwise precursor and most-intense five fragment mass differences. Precursor and fragment error are inferred by fitting a mixed Gaussian/uniform distribution to pairwise differences. Search parameter values are chosen by multiplying estimated error standard deviation by a multiplier associated with highest mean PSM yield in training datasets.

single organisms, Human Microbiome Project stool database for gut microbiome[30], or a site-specific sequencing-derived database for ocean microbiome[49]. We used a concatenated decoy database in which peptide sequences were reversed but C-terminal amino acids left in place. Search parameters included a static modification for cysteine carbamidomethylation (57.021464) and a variable modification for methionine oxidation (15.9949). Enzyme specificity was trypsin with proline cleavage suppression, with one missed cleavage allowed. Parent ion mass tolerance was defined around five isotopic peaks. False discovery rate (FDR) was calculated by target-decoy competition using Percolator,[33] and PSMs were accepted at FDR 0.01.

The most basic method of choosing parameters is to use settings associated with the typical performance of the instrument. This method is often used when the experimental details related to a dataset are unknown.In characterizing the instruments used to generate the training and test datasets, we deliberately used only the information available in the repository metadata, as would most researchers downloading the dataset from the repository. In several datasets, more detailed information, such as the mass analyzer used, was not publicly available. We defined "instrument default" settings for precursor ppm error and fragment bin tolerance for each instrument represented by the training and test datasets (Table 3.2), based on advertised instrument capabilities and literature search. We then held fragment bin tolerance for each experiment at the instrument default and performed ten separate searches, with settings for precursor ppm error varying uniformly over the range 5–50 ppm. Similarly, we held precursor ppm error at the instrument default and performed ten additional searches with settings for fragment bin tolerance varying uniformly over the range 0.02–1.0005 Da. A related parameter, fragment bin offset, should be set to roughly 0.4 when fragment bin size is near 1.0005 to ensure that the highest proportion possible of peaks associated with the same nominal mass are included in the same bin, but has little effect for other bin size values. This parameter was set to 0.4 in all searches. PSM yield for each search was defined as the number of PSMs at FDR 0.01.

3.2.3 Mapping estimated error to search parameter values

The final outputs of Param-Medic are precursor and fragment m/z tolerance values for use in a database search. To produce these estimates, we used the search results from our eight training data sets over a wide range of parameter settings, along with the empirical error standard deviations $\hat{\sigma}_\epsilon$, to estimate a multiplier that converts $\hat{\sigma}_\epsilon$ values into database search parameters that maximize PSM yield for a wide range of datasets. To this end, we normalized for differences in measurement error across the eight training datasets as follows. Separately for each parameter (precursor m/z tolerance and fragment bin size), we divided each parameter value vraw_i by the corresponding measurement error standard deviation $\hat{\sigma}_\epsilon$ for that sample and then calculated a normalized value $\hat{v}_i$ as the natural log of the result:

$$\hat{v}_i = \ln\left(\frac{v_i}{\hat{\sigma}_\epsilon}\right) \tag{3.2}$$

We then normalized the PSM yield y_{e_i} associated with the search of an experiment e with the ith value for the parameter, by dividing by the highest PSM yield observed for experiment e under any parmeter setting:

$$\hat{y}_{e_i} = \frac{y_{e_i}}{\max_{1\leq j\leq n} y_{e_j}} \tag{3.3}$$

For each experiment, this process yielded a different set of normalized parameter setting values, each associated with a different normalized PSM yield. In order to estimate the value associated with the highest mean normalized PSM yield over all experiments, we segmented the range from the minimum to the maximum values of the normalized parameter setting into 200 bins. We defined the yield of experiment e in bin b, $\hat{y}_{e_b}$, as the normalized PSM yield in that experiment associated with that bin, interpolating linearly between adjacent observed measurements $\hat{y}_{e_i}$ and using the yields for the bins with highest and lowest normalized parameter values for each dataset to stand in for all higher-value or lower-value bins not searched for that dataset (Figure 3.3). We then chose the bin b' associated with the highest mean normalized yield over the n experiments:

$$b' = \arg\max_b \frac{1}{n}\sum_{e=1}^{n} \hat{y}_{e_b} \tag{3.4}$$

Experiment	Instrument	Organism	Precursor tolerance (ppm)	Fragment bin size (Th)
		Training Datasets		
2014kim-kidney [39]	Orbitrap Velos	human	10	0.05
2014kim-lung [39]	Orbitrap Elite	human	10	0.05
2015clark-redefining [8]	LTQ Orbitrap	human	50	1
2015radoshevich-isg15 [72]	QExactive	human	4.5	0.02
2015tanca-impact [80]	Orbitrap Velos	human gut microbiome	10	0.02
2015uszkoreit-intuitive [84]	Orbitrap Elite	mouse	5	0.4
2016mann-unpublished	QExactive	human	10	0.02
2016schittmayer-cleaning [77]	Orbitrap Velos	yeast	10	0.8
		Test Datasets		
2016may-metapeptides [49]	Qexactive	ocean microbiome	10	0.02
2016audain-in-depth [2]	LTQ Orbitrap	yeast	25	0.5
2016zhong-quantitative [95]	Orbitrap Velos	human	20	0.5

Table 3.1: Experiments used in the training and testing of Param-Medic and their associated search parameters as adapted from their publications.

The center of bin b', $\bar{b}'$, is the natural log of Param-Medic's estimate of the optimal multiplier relating one of the two $\hat{\sigma}_\epsilon$ values to its corresponding search parameter value. Therefore, to calculate the optimal precursor tolerance or fragment bin size, Param-Medic multiplies the appropriate $\hat{\sigma}_\epsilon$ estimate by its associated $\exp(\bar{b}')$.

Param-Medic will refuse to estimate precursor error or fragment bin tolerance if there are fewer than 200 pairs of values that make up the mixed distribution, a situation representing a bit of a quagmire for tolerance estimation. It will also fail if, as was the case in one of our training datasets, at least half of the values in the mixed distribution are exactly 0. This situation occurs when the values are rounded, and it is incompatible with the Param-Medic approach.

3.2.4 Alternative parameter-setting strategies.

We compared search PSM yield from settings determined by Param-Medic with PSM yield from searches using other means of determining search parameters. In addition to the in-

Instrument	precursor (ppm)	fragment bin (Th)
LTQ Orbitrap	20	1.005
Orbitrap Velos	20	0.05
Orbitrap Elite	20	0.02
QExactive	20	0.02

Table 3.2: Settings used in "instrument default" searches.

strument defaults described above, we also derived parameter settings from the publications describing the datasets (or, in the case of one as-yet-unpublished training dataset, from the experimental metadata provided in the PRIDE repository for project ID PXD002854). Because the datasets were originally searched with a variety of search algorithms, the published parameter values may not map directly to Comet precursor tolerance and fragment bin size; ours is a good faith effort to represent the original searches as accurately as possible within the Comet/Percolator framework. We also used Preview to assess precursor and fragment median *m/z* error. To map these Preview-estimated error values to Comet search parameters, we used five times the median error, which is the the "rule of thumb" suggested in the Preview user manual.

3.3 Results

3.3.1 Param-Medic's performance.

We evaluated Param-Medic's performance in terms of PSM yield, comparing it with the settings used in the original papers describing our datasets, with instrument default settings, and with Preview. On seven training datasets (Figure 3.2), Param-Medic parameter settings yielded 96% to 152% as many PSMs as settings from the original papers (median: 105%), and 99% to 334% as many as defaults based on instrument type (median: 103%). Param-Medic failed to find a sufficient number of repeated ions for parameter estimation on one training

dataset because of a large proportion of exactly identical sequential values for precursor m/z, which we speculate was due to rounding of the precursor m/z values. Preview failed on the same training dataset as Param-Medic due to insufficient search results for error estimation. On the remaining seven datasets, Param-Medic yielded 99% to 135% as many PSMs as Preview (median: 101%).

On three test datasets, Param-Medic parameter settings yielded 99–104% as many PSMs as settings from the original papers describing the experiments (median: 100%), and 103% to 212% as many PSMs as defaults based on instrument type (median: 104%). Preview failed on one test dataset due to insufficient search results for error estimation. On the other two, Param-Medic yielded 95% and 99% as many PSMs as Preview (Figure 3.2).

To assess the suitability of Param-Medic for evaluating a different kind of mass spectrometry data, we used it to evaluate a human dataset from a SCIEX TripleTOF 5600 (PRIDE accession number PXD000307). When we searched this dataset using the parameters specified in the PRIDE submission (10 ppm precursor tolerance, 0.4 Th fragment bin size), the PSM yield was 2738. Yield with parameter values estimated by Param-Medic (10.45 ppm precursor tolerance, 0.03 Th fragment bin size) was 2949, an increase of 7.7%.

Any method for automatically estimating m/z search parameters should be fast as well as effective at optimizing PSM yield. On a 3.0GHz Intel Core Duo processor, Param-Medic ran in a few seconds to just over a minute on all training and test datasets, while Preview ran in a few minutes to nearly an hour and a half (Table 3.3). Param-Medic's running time scaled with the number of spectra per experiment, while Preview's scaled with both both the number of spectra and the size of the database. Preview took 88 minutes to run on the human gut microbiome sample, which it searched against a large gut microbiome database, even though that sample had just 10% more spectra than a human sample on which Preview ran in 14 minutes. The Preview running times are dominated by the database search, but also include some time spent performing activities not required for inferring mass error (e.g., inferring peptide digestion and variable modifications).

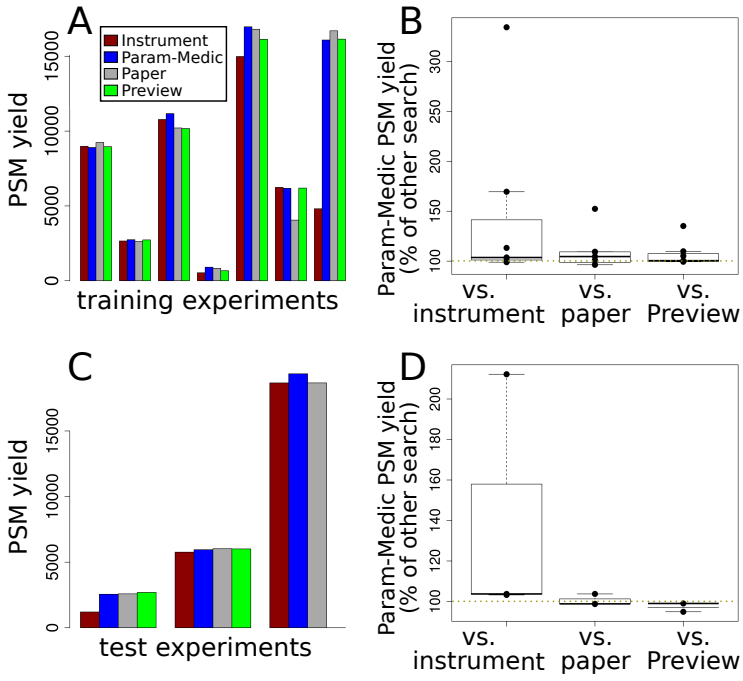

Figure 3.2: **Comparing Param-Medic with other methods.** A. PSM yield at FDR 0.01 using parameters determined by four different methods: instrument defaults, Param-Medic, original paper settings, or Preview. Each cluster of bars represents one of the seven training experiments for which Param-Medic and Preview returned error estimates. Results are reported for the seven training data sets. B. Box plots showing the distribution PSM yield of searches with Param-Medic parameters as a percentage of the PSM yield using instrument defaults, original paper settings, and Preview, over the same seven training experiments. C and D. As A and B, but showing data from the three test experiments.

Experiment	Organism	Spectra	Preview	Param-Medic
	Training Datasets			
2014kim-kidney[39]	human	9,072	2	0.07
2014kim-lung[39]	human	17,612	3	0.13
2015clark-redefining[8]	human	38,570	N/A	N/A
2015radoshevich-isg15[72]	human	63,185	14	1.03
2015tanca-impact[80]	human gut microbiome	69,685	88	0.48
2015uszkoreit-intuitive[84]	mouse	26,992	6	0.67
2016mann-unpublished	human	41,157	7	0.12
2016schittmayer-cleaning[77]	yeast	9,297	1	0.19
	Test Datasets			
2016may-metapeptides[49]	ocean microbiome	98,317	N/A	0.68
2016audain-in-depth[2]	yeast	18,175	2	0.35
2016zhong-quantitative[95]	human	14,962	3	0.27

Table 3.3: Wall-clock running times for Preview and Param-Medic on each experiment, in minutes. "N/A" indicates that a tool did not run successfully on a given experiment.

3.3.2 PSM yield variation between parameter settings.

Some of our training experiments were much more sensitive to parameter settings than others. The extremes in difference in PSM yield between optimal and suboptimal settings for either parameter were quite high, with the worst and best parameter settings for precursor error yielding between 9% and 117% as many peptides as the instrument default settings, and for fragment bin size yielding between 0% and 339% (Figure 3.3A and 3.3B). The relationship between parameter settings and PSM yield was not consistent within an instrument type, with, for instance, the two QExactive experiments having opposite trends in yield as a function of precursor error tolerance. These results further demonstrate that the values specified for precursor and fragment tolerances can have a sizeable impact on PSM yield, and that knowledge of instrument type alone is not sufficient to set those parameters optimally.

For fragment bin size, there was very close agreement between the experiments as to the optimal multiple of estimated error standard deviation (0.005). For precursor tolerance, the agreement was not as complete, with two experiments holding the most influence over the derived optimal multiple (37.40) due to their high sensitivity to changes in this parameter (Figure 3.3C and 3.3D). The lower level of agreement for precursor tolerance may reflect differences in the density of candidate precursor matches in the target and databases being searched against.

3.4 Discussion

We have demonstrated that Param-Medic optimizes precursor error and fragment bin size parameter settings for LC-MS/MS search based on characteristics of the dataset being searched. Param-Medic assumes that LC-MS/MS experiments are likely to make multiple observations of many peptide ions. Ironically, this phenomenon is often perceived as a chronic problem plaguing data-dependent acquisition proteomics: high-abundance peptides, in particular, will tend to trigger multiple MS/MS scans, leading to fewer acquisitions of other peptides. Accordingly, instrument makers and researchers often adjust a dynamic exclusion window to

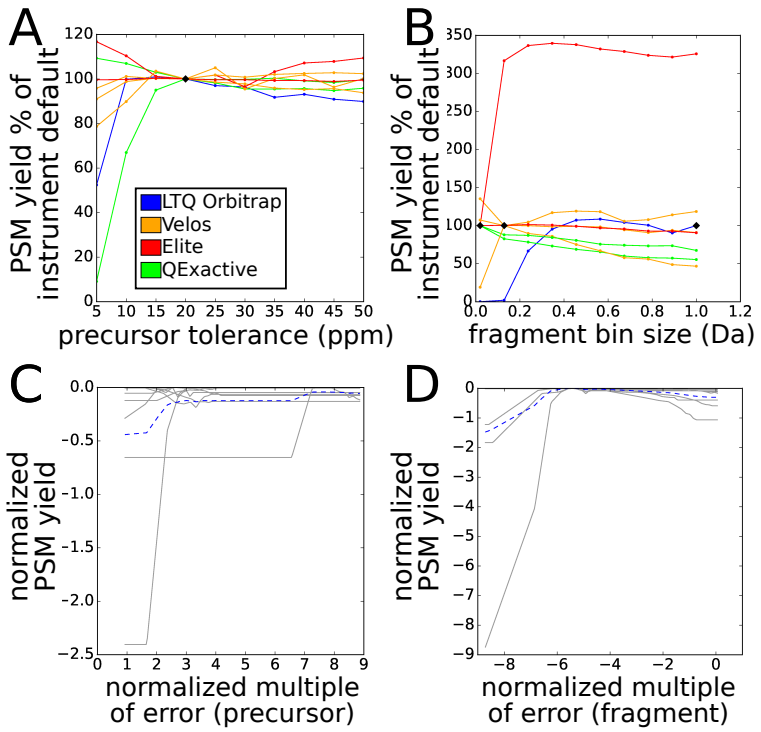

Figure 3.3: **PSM yield *vs.* parameter settings in training datasets.** Panels A and B show PSM yield at FDR 0.01 as a function of the percentage of the PSM yield for that dataset when searched with instrument default settings. Each line represents a different training dataset, colored by instrument type. Black diamonds indicate instrument default settings. A: varying precursor tolerance from 5 ppm to 50 ppm. B: varying fragment bin size from 0.02 Da to 1.005 Da. Panels C and D show normalized PSM yield as a function of normalized error. Vertical axis measures normalized PSM yield at FDR 0.01: the natural log of the ratio of the yield at a given setting to the maximum yield at any setting.Horizontal axis measures parameter setting as the natural log of a multiple of the estimated standard deviation of measurement error. Gray lines represent individual experiments; blue line represents mean across all experiments. C: varying precursor tolerance. D: varying fragment bin size.

minimize these repeated measurements, but such measurements are nonetheless a constant feature of most proteomics experiments. Param-Medic exploits these repeated measurements to provide valuable information about the m/z tolerance characteristics of the experiment.

On several of our training and test datasets, Param-Medic increased PSM yield greatly over parameter settings chosen based on instrument type. Many researchers will spend time iteratively fine-tuning their search settings for a particular instrument over multiple experiments in order to maximize yield, a process that Param-Medic can assist with. In other circumstances, instrument-based parameter settings are used often, as when searching experimental data provided by collaborators or downloaded from a public repository, with minimal description. Param-Medic showed particularly large improvement over instrument defaults for one of the Orbitrap Elite training datasets. Neither the paper describing the dataset nor the experimental metadata from PRIDE indicated whether the Orbitrap Elite was run in FT-FT mode (i.e., fragments analyzed in the orbitrap) or in FT-IT mode (i.e., fragments analyzed in the ion trap). Accordingly, we naïvely assumed the more-common (and higher-accuracy) FT-FT settings in our "instrument default" parameter settings. Upon further inspection, however, metadata in the mzML file for the acquisition indicated that the instrument was run in FT-IT mode. This setting likely accounts for the much higher yield at wider fragment bin settings and demonstrates that Param-Medic's error estimation can infer properties of the analysis that differ greatly from what might be expected from experimental metadata alone.

In our training and test datasets, Param-Medic settings yielded modestly more PSMs than settings chosen by experts for searching their own data for publication (52% more in one training dataset). We do not know what criteria these authors used to choose the settings, and the settings may have behaved quite differently in their hands, using different search engines or values for parameters other than the two considered here. However, the consistency of the trend indicates that many labs may benefit from an approach to parameter-setting that is based on the characteristics of the individual experiment being searched. The applicability of Param-Medic to Q-TOF data has particular potential to aid a subset of

proteomics researchers. Some Q-TOF manufacturers write mass-corrected values in the raw data files, while others do not, leading many researchers to use a very wide and potentially suboptimal precursor tolerance in searching Q-TOF data.

In terms of PSM yield, Param-Medic performs very similarly to Preview on most datasets evaluated, with a large advantage in PSM yield in a single training experiment and nearly identical performance in our test experiments (Figure 3.4 compares the parameter estimates derived from Param-Medic and Preview on the training and test datasets). Param-Medic and Preview each fail to assess error in different circumstances: Preview when its database search fails, Param-Medic when there are insufficient or suspicious differences in measurements available for error estimation. In our training and test datasets, Param-Medic refused to estimate error once, whereas Preview refused to estimate error on that same experiment and on one other experiment. An important difference between the tools is that Preview infers instrument calibration error in addition to measurement precision, and so would presumably provide superior guidance for acquisitions with large calibration errors. On the other hand, Preview is proprietary software and runs only on Windows. Param-Medic is implemented in Python as a standalone tool and is also integrated into the Crux toolkit for streamlined parameter estimation and search with Comet and Tide search engines. In both incarnations, Param-Medic is open source and can be run on Windows, Linux and Mac. Furthermore, the Param-Medic running time is much shorter than that of Preview. Preview's running time scales with both the number of MS/MS spectra and the database size, whereas Param-Medic's running time scales only with the number of spectra. In practice, neither tool's running time likely to be onerous, except possibly for Preview when the search database is large. This occurs often, for instance, in a metaproteomics context.

Although Param-Medic provides an estimate of ppm fragment error that could be used with any search engine, it currently only provides guidance for mapping this value to an appropriate fragment tolerance for search engines such as Comet, Sequest and Tide that use fragment binning. Future work will include a reanalysis of the training datasets in order to provide such guidance for search engines that use fragment tolerances rather than fragment

bins.

Param-Medic has been implemented as a standalone Python 2.7 tool which may be downloaded (including source code) at https://github.com/dhmay/param-medic or simply added to a Python installation with the 'pip' tool. It has also been incorporated into version 3.1 of the Crux Toolkit, available at http://crux.ms. Within Crux, Param-Medic is available as a standalone tool and is also integrated into the Tide and Comet search algorithms for automatic detection of optimal parameter settings. All proteomics datasets described here, and links to all software, may be found at http://noble.gs.washington.edu/proj/param-medic/.

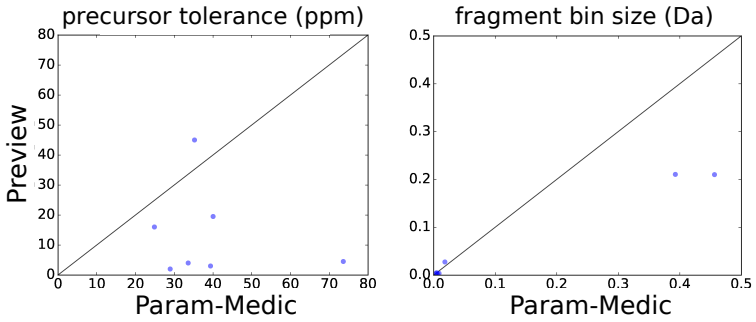

Figure 3.4: **Comparing parameter estimates from Param-Medic and Preview.** Scatter plots of parameter values estimated by Param-Medic (horizontal axis) and Preview (vertical axis) on all nine training and test experiments for which both tools returned values. Solid line represents 1:1 correspondence. Left: precursor error, $r = -0.20$. Right: fragment bin size, correlation coefficient $r = 0.99$. The fragment bin size estimates show high correlation ($r = 0.99$), though this value is largely driven by two outlier datasets.

Chapter 4

GLEAMS

4.1 Introduction

Since the publication of the SEQUEST[15] search algorithm in 1994, the dominant approach to assigning peptide sequences to tandem mass spectra has been to derive a list of candidates for each spectrum from a database, generate a theoretical spectrum for each candidate, and then score each putative peptide-spectrum match based on the similarity between the observed and theoretical spectrum fragments. Major advances have been made in search algorithm development and various downstream analyses,[79] but an individual lab searching their spectra against a sequence database remains the dominant paradigm for tandem mass spectrum identification.

Over the last decade, public proteomics repositories such as PRIDE[46] and MassIVE[54] have grown to include hundreds of millions of tandem mass spectra from tens of thousands of assays. As these repositories have become more comprehensive, efforts have been undertaken to make these spectra useful to researchers analyzing new datasets. Some approaches, such as PeptideAtlas,[17] the Global Proteome Machine Database,[18] the NIST spectral libraries,[43] and MassIVE,[88] involve re-searching the spectra using a common workflow. In each case, the output of the analysis is a spectral library, in which sets of spectra corresponding to the same peptide sequence are condensed into a single spectrum either by averaging or selecting a single, representative spectrum per peptide. The spectral library can then be used to analyze new data sets using a search algorithm. A drawback to any method that relies on standard database search is that each spectrum is, fundamentally, treated as an independent observation. By failing to jointly consider all of the spectra together, these pipelines miss out on the opportunity to exploit valuable structure in the data.

An alternative to simple re-searching of the data is to employ clustering algorithms, several of which have been developed specifically for clustering mass spectra.[5,19] In practice, we are aware of only one such method that has been applied to a large proportion of the peptide mass spectra in a repository: PRIDE Cluster[23,24] clustered all of the publicly available spectra in the PRIDE Archive in 2015, producing one spectral library for each of

several commonly-studied organisms and producing a consensus spectrum for each cluster. Any clustering approach, however, is limited in its ability to incorporate new data sets. In practice, new spectra that correspond to previously detected peptides can easily be added to the corresponding clusters, but entirely new clusters cannot be added without running the entire clustering algorithm from scratch. This is an extremely expensive operation that becomes more expensive as the repository grows, and presumably for this reason PRIDE Cluster has not been updated since 2015.

More fundamentally, clustering is problematic because it is an *unsupervised* approach. The input to a clustering algorithm is an unlabeled set of spectra. In practice, the labels (i.e., the associated peptide sequences) are used only in a *post hoc* fashion, to choose how many clusters to produce or to split up large clusters associated with multiple peptides.

In recent years, a revolution has occurred in machine learning, with deep neural networks proving to have applicability across a wide array of problems. Accordingly, within the field of proteomics, deep neural networks have been applied recently to the problems of *de novo* peptide sequencing,[83] predicting MS/MS spectra[96] and chromatographic retention time[45] for peptides, and protein inference.[82] However, to our knowledge no one has yet applied deep neural networks to the problem of making public repository spectra contribute to the analysis of new mass spectrometry experiments. We hypothesize that we can obtain more accurate and useful information about a large collection of spectra by using a supervised deep learning method that directly exploits peptide-spectrum assignments during joint analysis.

Accordingly, we propose GLEAMS (GLEAMS is a Learned Embedding for Annotating Mass Spectra), which is a deep neural network that has been trained to embed tandem mass spectra into a 32-dimensional space in such a way that spectra generated by the same peptide, with the same post-translational modifications and charge, are close together. Our work builds upon methods that have been used successfully to embed various types of data items, from text documents[36] and images[12] to protein sequences and structures[51] to "all the things".[91] The learned spectral embedding offers the advantages that new spectra can efficiently be embedded into the space and automatically associated with existing or new

clusters of spectra as needed. Our approach is fundamentally different from previous large-scale analyses, in the sense that, as the repository grows, previously unclustered spectra have the opportunity to join nascent clusters without requiring computationally onerous re-clustering.

We validate the embedding by demonstrating its ability to place spectra assigned the same label by database search close together, and we describe a method for using this embedding to assign peptide labels to new spectra. We also describe a method for detecting spectrum "communities": groups of spectra that all represent the same peptide. We use these communities to help identify systematic sources of error in annotations produced by database search and to help characterize the so-called "dark matter" of proteomics. We provide a software implementation of our method, as well as a pre-trained model that can be used to efficiently embed spectra for joint analysis (https://bitbucket.org/noblelab/gleams).

4.2 Results

4.2.1 A deep neural network learns to embed millions of spectra into a common latent space

The learned model consists of a "Siamese network,"[25] in which two copies of an embedding network operate side by side (Figure 4.1a). During training, the network is provided with pairs of spectra s_a and s_b and an associated label Y, where $Y = 1$ indicates that the spectra were generated by the same peptide, and $Y = 0$ indicates that they were generated from different peptides. Each embedder transforms a spectrum s_a into its embedded representation E_a. The key to the learning process is the contrastive loss function adapted from Hadsell et al.,[25] defined as

$$L(W, Y, E_a, E_b) = \frac{Y}{2}||E_a - E_b||_2 + \frac{1-Y}{2}(\max(0, 1 - ||E_a - E_b||_2))^2, \tag{4.1}$$

where W represents the learned network weights. Intuitively, this loss function pulls the two spectra together if they are associated with the same peptide ($Y = 1$) and pushes them apart if they are associated with different peptides ($Y = 0$). Backpropagation from this loss function is used to update the weights in the network.

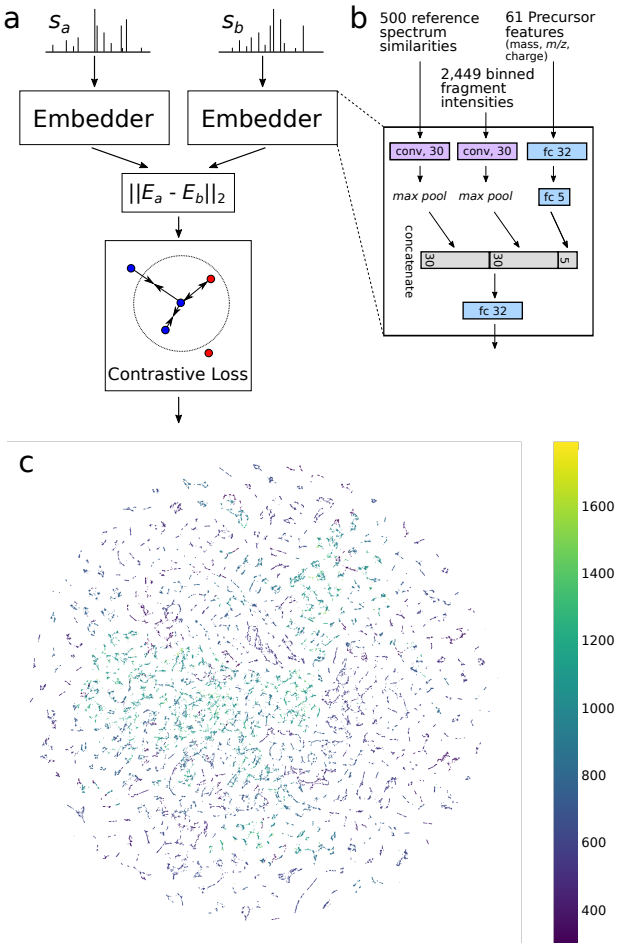

Figure 4.1: **Learning to embed spectra.**

(A) Two spectra, s_a and s_b, are encoded and passed as input to two instances of the embedder network, with tied weights. The Euclidean distance between the two resulting embedded spectra, E_a and E_b, is passed to a contrastive loss function that penalizes far-apart same-label spectra and nearby different-label spectra, up to a margin of 1. (B) The embedder processes each of the three types of inputs separately. (C) A t-SNE projection of 70,000 randomly chosen spectra into two dimensions indicating a large influence of precursor m/z on the embedded space.

The heart of the model itself is the embedder network (Figure 4.1b) which takes as input a spectrum and embeds it into a 32-dimensional space. The model contains two copies of the embedder, with weights tied so that updates to one embedder are always reflected in the other. For input to the embedder, each spectrum is encoded using three sets of features representing, respectively, attributes of the precursor ion, binned fragments, and similarities to an invariant set of reference spectra (see Methods for details). The precursor features are processed through a two-layer fully-connected network, and the binned fragment and reference spectrum similarity features are each passed through a separate, single-layer convolutional neural network (CNN). Finally, the outputs of the three networks are concatenated and passed to a final, fully-connected layer with dimension 32.

To train and validate our model, we constructed a repository containing 5,462,275 mass spectra of charge state 2 or higher from 22 experiments (Table 1). Among these spectra, 1,650,587 (30.2%) were identified by database search at a 1% PSM-level FDR threshold, representing 170,946 distinct peptide sequences (see Methods for details of the search procedure and FDR assignment). We then randomly split the dataset by experiment, using 1,125,586 labeled spectra from 16 experiments for training and reserving 525,001 labeled spectra from six experiments for validation. Training the network on the training set required four hours and 45 minutes on a machine with an Intel Xeon(R) E5-2650 CPU, a Tesla K40c GPU and 90GB memory.

The design of the Siamese network required selection of a variety of hyperparameters. Some of these hyperparameters, such as the output dimensions, the number of layers, and the number of nodes per layer in each component of the network, are explicit. Other hyperparameters are implicit, such as how to encode precursor mass information or what type of learning rate schedule to employ in training the model. During development of the model, we partially explored this hyperparameter space (see Methods), leading us to the particular model described here.

We embedded all 5,462,275 repository spectra in 19 minutes and four seconds, at 4,775 spectra per second. As an initial evaluation of the learned embedding, we performed a

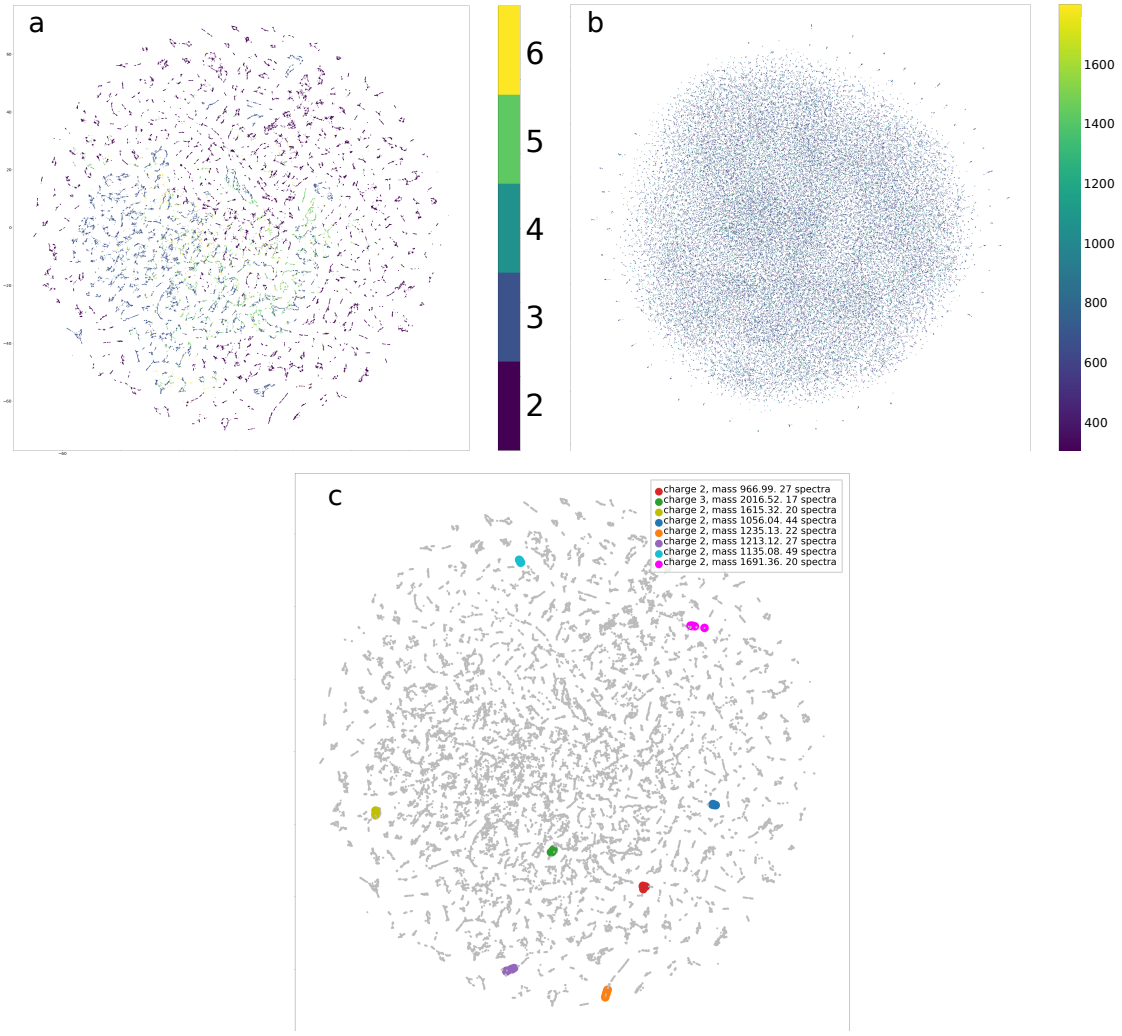

Figure 4.2: **Additional t-SNE projections.** Each point represents a spectrum. (A) colored by charge state. (B) With the per-spectrum values for each of the 32 dimensions of the embedded spectrum matrix shuffled prior to running t-SNE. The lack of "clumpy" structure in this plot demonstrates that the structure observed in the unpermuted plots is not an artifact of the t-SNE algorithm. (C) Unpermuted spectra, with all spectra within a single randomly chosen charge state and 1.000507 Da mass bin (158 mass spectra across all eight bins) each given a different color and larger dot size. The spectra from each mass bin all occur within the same globular structure. Legend indicates charges and centers of each mass bin.

further projection down to two dimensions. We embedded 70,000 randomly-chosen repository spectra, projected into 2D with t-SNE,[85] and colored the resulting points by precursor m/z (Figure 4.1c) and charge (Figure 4.2a). The observed "clumpy" structure does not occur when the values for each of the 32 embedded dimensions are randomly permuted (Figure 4.2b), demonstrating that this structure is not an artifact of the t-SNE embedding. The visualizations suggest that precursor m/z, mass, and charge strongly influence the structure of the embedded space, roughly determining the location of each spectrum. The small, globular structures that comprise the visualization each tend to contain spectra of a single charge state and nominal precursor mass (Figure 4.2c).

4.2.2 Spectrum communities contain spectra generated by the same peptide

If our training worked well, then spectra generated by the same peptide should lie close together, according to a Euclidean metric, in the embedded space. Accordingly, we investigated, for 200,000 randomly chosen embedded spectra, the relationship between neighbor distance and the proportion of labeled neighbors that have the same peptide label. The results (Figure 4.3a) show that neighbors at small distances overwhelmingly represent the same peptide. Furthermore, the few different-peptide labels at very small distances almost entirely represent single amino acid substitutions in which the masses of the two amino acids differ by less than 1 Da. We demonstrate below that these apparent single amino acid substitutions nearby in embedded space largely represent false peptide labels from database search.

Based on this relationship, we aimed to develop an efficient method for finding dense clusters of spectra, which we refer to as "spectrum communities." We considered three potential community detection algorithms. The first, simplest method is a "hub-and-spoke" procedure in which "hub" spectra are associated with their neighbors ("spokes") within a specified Euclidean distance threshold τ (see Methods and Algorithm 1 for details). The second method involves running the k-means clustering algorithm in the embedded space and then calling each of the resulting clusters a community. The third method is based on

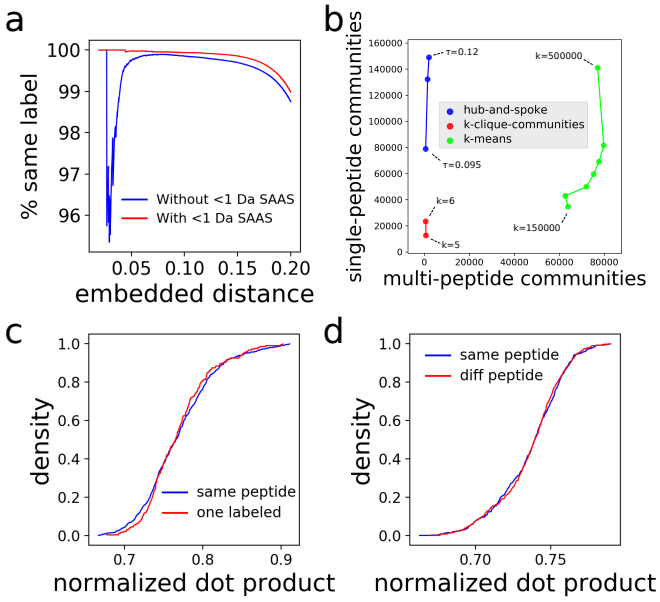

Figure 4.3: **Validation of the learned embedding**

(a) Relationship between embedded distance and proportion of same-peptide labels. Blue line: considering isobaric leucine-isoleucine substitutions to represent the same peptide. Red line: considering single amino acid substitutions (SAASs) for amino acid pairs with masses within 1 Da to represent the same peptide. **(b)** Comparisons of the numbers of single-peptide and multi-peptide communities detected among the 3,390,759 charge-2 repository spectra by the hub-and-spoke method with τ ranging from 0.095 to 0.12, the k-clique-communities method with $k = 5$ and $k = 6$, and k-means clustering with values of k: ranging from 150,000 to 500,000. **(c)** Cumulative density plot of normalized dot products between pairs of spectra in a single community, for pairs of spectra labeled with the same peptide (blue line) and with one spectrum identified and the other unidentified (red line), from a single-peptide community. **(d)** Same as panel (C), but using a two-peptide community. Pairs of spectra are labeled with the same peptide (blue line) and with two different peptides varying by an E to K substitution (red line).

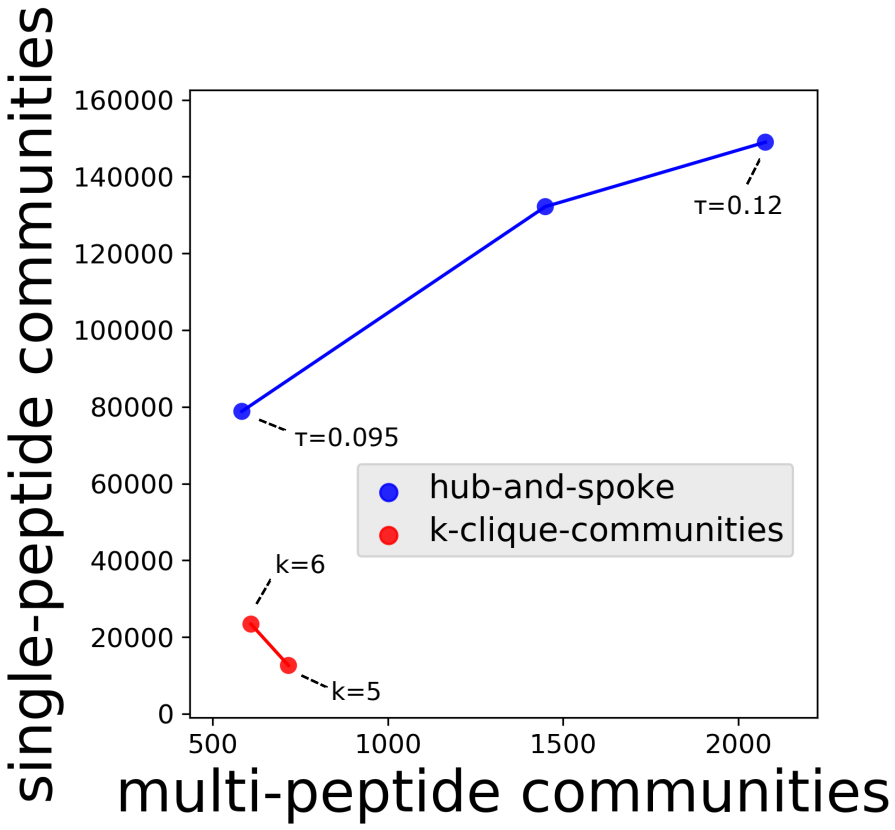

Figure 4.4: **Comparing hub-and-spoke and k-clique-communities methods for community detection.** Comparisons of the numbers of single-peptide and multi-peptide communities detected among the 3,390,759 charge-2 repository spectra by the hub-and-spoke method with τ ranging from 0.095 to 0.12 and the k-clique-communities method with $k = 5$ and $k = 6$.

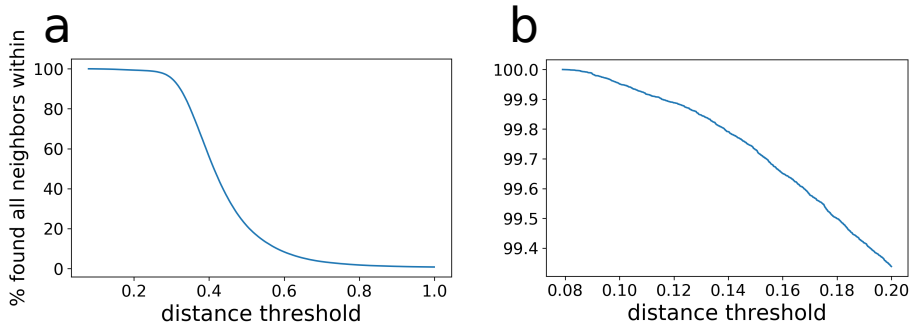

Figure 4.5: **Percentage of spectra with 1000 nearest neighbors within distance thresholds.** The percentage of embedded spectra (vertical axis) having all 1000 of their nearest 1000 neighbors within a given Euclidean distance threshold (horizontal axis). (A) Distance thresholds < 1 (B) Distance thresholds < 0.2

the notion of a "k-clique community."[69] In this setting, a "k-clique" is a set of k spectra that are completely connected to one another, subject to a distance threshold τ, and a k-clique community is the maximal union of k-cliques that can be reached from each other through a series of adjacent k-cliques with $k-1$ members in common. The k-clique community method was developed for the purpose of finding the most highly-overlapping cohesive groups of nodes in biological networks, and was designed to focus on the local structure of the network rather than the values of the underlying distances. For the hub-and-spoke and k-clique community methods, we selected value of τ such that 1% of the resulting spectrum communities with identified spectra contained spectra identified with more than one unique peptide sequence. Note that, for both of these methods, we also employed an approximate k-nearest neighbor criterion, with $k = 1000$, for computational efficiency (see Methods for details). The 1000-neighbor threshold has only a minimal impact on neighbor detection at relevant distance thresholds (Figure 4.5). Constructing an index for efficient nearest-neighbor search took 81 minutes and 23 seconds, and finding the 1000 nearest neighbors of all 5,462,275 spectra took seven hours and 17 minutes on a single GPU, at 205.5 spectra per second.

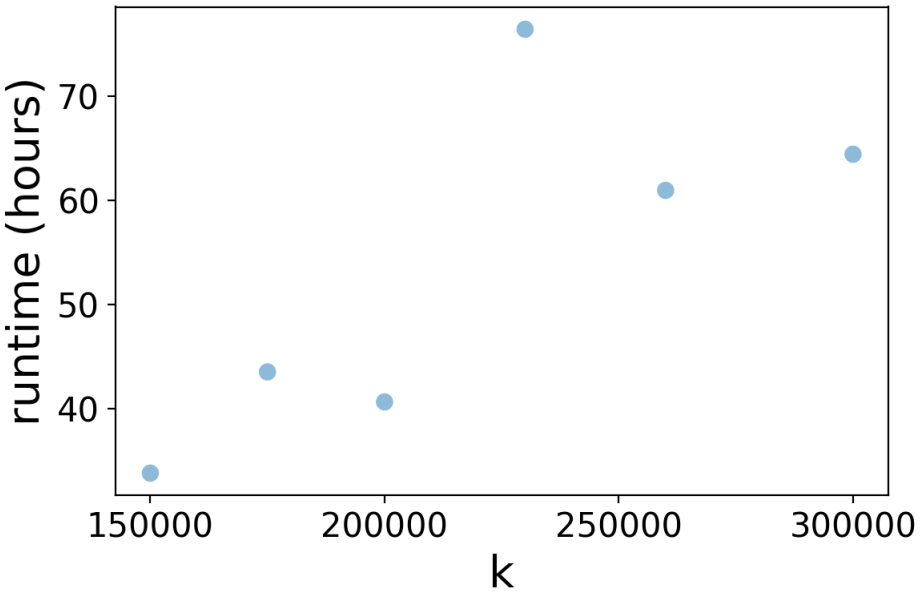

Figure 4.6: **Running time for k-means clustering on the charge-2 spectra as a function of k.** k-means clustering was performed with an Intel Xeon(R) E5-2650 CPU and 90GB memory available.

Comparison of the results for these three methods strongly suggests that the hub-and-spoke model yields the most useful clusters. To evaluate the methods, we applied them to the entire repository and then plotted each set of detected spectrum communities in terms of the number of single-peptide versus multi-peptide communities (Figure 4.3b, detail in Figure 4.4), with the aim of producing many single-peptide communities and few multi-peptide communities. The k-means algorithm, for values of k in the range 150,000 to 300,000, produced a large proportion of communities containing multiple peptides. It is possible that scaling to a very large value of k would yield better performance, but this turned out to be computationally infeasible (Figure 4.6). The k-clique community method, on the other hand, produced very few communities overall.

The hub-and-spoke method detected a large number of spectrum communities representing a single peptide. The method identified 229,167 spectrum communities ranging in size from 2 to 2,387 (mean: 3.9; median: 2). Among the communities, 127,788 were "no-peptide" communities containing no spectra identified by database search; 100,364 were "single-peptide" communities containing identified spectra representing only one peptide, and the remaining 1,015 were "multi-peptide" communities containing identified spectra representing two or more peptides. Most (78.7%) communities contained spectra originating from a single experiment, but some larger communities contained spectra from up to 13 different experiments (mean: 1.4). Every community contained spectra with only a single charge state. Communities contained spectra spanning mass ranges of size 0.000 to 3,971.844 Da (mean: 0.053; median: 0.001).

As a further demonstration that the hub-and-spoke communities typically contain spectra generated by a single peptide, we randomly chose a single-peptide community containing 42 labeled and 6 unlabeled spectra, and we calculated normalized dot products between all pairs of labeled spectra and between all labeled-unlabeled spectrum pairs. The normalized dot products are distributed nearly identically for the two types of pairs (Figure 4.3c), suggesting that the unidentified spectra were generated by the same peptide and represent false negatives from database search. Furthermore, we randomly chose a two-peptide commu-

nity containing spectra labeled with two different peptides (EVLGAFSDGLAHLDNLK and KVLGAFSDGLAHLDNLK) differing by a single E-to-K substitution representing a mass difference of 0.94763 Da, and we calculated normalized dot products between all pairs of same-labeled and different-labeled spectra. The distributions were again nearly identical (Figure 4.3d), and the precursor masses of all the spectra spanned only 0.004 Da, suggesting that the spectra were in fact generated by the same peptide and that some of the labels from database search were false positives due to necessarily loose precursor and fragment tolerances. We therefore investigated in more detail the spectra that, according to their GLEAMS community membership, appear to be mis-identified.

4.2.3 Embedding detects mis-identified spectra

As a first step in this investigation, we propagated identifications among spectra in single-peptide communities. Among the collection of 229,167 such communities, 17,948 (7.8%) communities contained a mixture of unidentified and identified spectra. We assigned peptide identifications to 40,053 unidentified spectra by propagating identifications from the other spectra in the these communities.

Next, we investigated the multi-peptide communities and determined that many of them contained only spectra that appear to have been generated by the same peptide despite having different peptide labels from database search. As is typical in proteomics analysis, our database searches were performed with a 1% FDR threshold. Hence, we expect at most 1% of the repository labels to be incorrect. Among these false discoveries, certain types of mis-identifications are particularly likely, including single amino acid substitutions. Not counting leucine-isoleucine, an isobaric substitution, we found 256 communities that contained two or more unique peptide labels that differed by only a substitution between amino acids that differ by less than 1 Da: E-K, E-Q, D-N, K-Q, L-N and I-N. We considered two possible explanations for these communities: either our approach generated communities containing spectra generated by multiple peptides, or some of the original PSMs from database search were incorrect. In these specific cases, we hypothesized that a false positive match with a

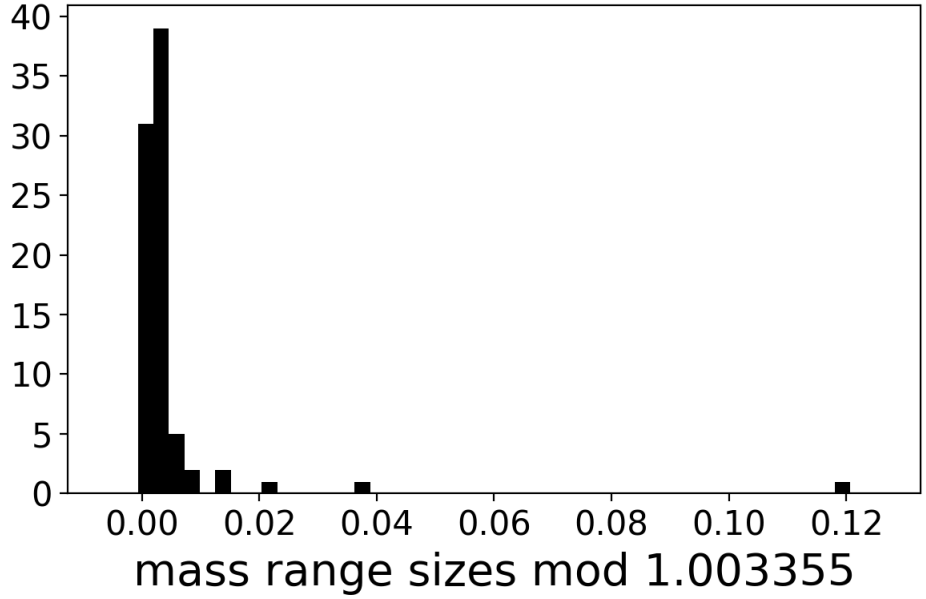

Figure 4.7: **Communities with single amino acid substitutions appear to be generated by a single peptide.** Mass ranges, modulo 1.003355 (the mass difference between ^{13}C and ^{12}C), of all 82 spectrum communities containing only spectrum identifications representing a single E-to-K amino acid substitution.

single amino acid substitution from the true generating peptide could arise due to the "isotope error" parameter setting we used in database search, which allows for identification of the isotopic peaks with greater mass than the monoisotope. Although this common parameter setting results in greater overall sensitivity at a given FDR, when combined with relatively loose tolerances for precursor mass error and fragment bin size (which are appropriate for many of the runs in our repository) it can produce false positives of this specific type.

Further analysis supports the conclusion that the spectrum communities including multiple peptides primarily arise due to incorrect original PSMs. For several such communities, we calculated pairwise normalized dot products between each pair of (unembedded) spectra. In each case, the normalized dot products between spectrum pairs with different peptide

labels were no lower than the products between pairs with the same label (p=0.45 by one-tailed Mann-Whitney U test for the example shown in Figure 4.3b). Furthermore, for each amino acid substitution listed above, the 95th percentile of the sizes of the mass ranges for all communities containing the substitution was far smaller than the fractional mass difference (mass difference modulo 1.003355, the difference between ^{12}C and ^{13}C) between the two amino acids (e.g., for E-K, 0.014 *vs.* 0.052). See Figure 4.7 for details.

These single-amino-acid-substitution communities, then, present opportunities for correcting false identifications from database search. The 256 such communities contained 1,957 spectra. For each community, we calculated the median precursor mass, chose the peptide sequence with the smaller precursor mass difference and assigned that sequence to the unidentified and mis-identified spectra. This approach allowed us to correct 717 false identifications and propagate those identifications to an additional 75 originally unidentified spectra.

4.2.4 Elucidating the "dark matter": targeted analysis of unidentified spectra in communities

A key outstanding question in protein mass spectrometry analysis concerns the source of spectral "dark matter," i.e., the spectra that are observed repeatedly across many experiments but consistently remain unidentified. To characterize such spectra, we focused on the 127,788 spectrum communities that contain no identified spectra. We then applied a cascade search strategy to the hub spectra, in which we search each spectrum against a series of increasingly large databases, performing FDR control after each search and passing only the unidentified spectra to the next stage.[37]

First, we performed a "tight" target-decoy database search (with appropriate precursor and fragment tolerances and only acetylated cysteine and potentially oxidized methionine as modifications) of the 127,788 hub spectra. This search differed from the initial database searches of the mini-repository runs chiefly in the choice of database: each hub spectrum was searched against all databases associated with one or more members of its spectrum

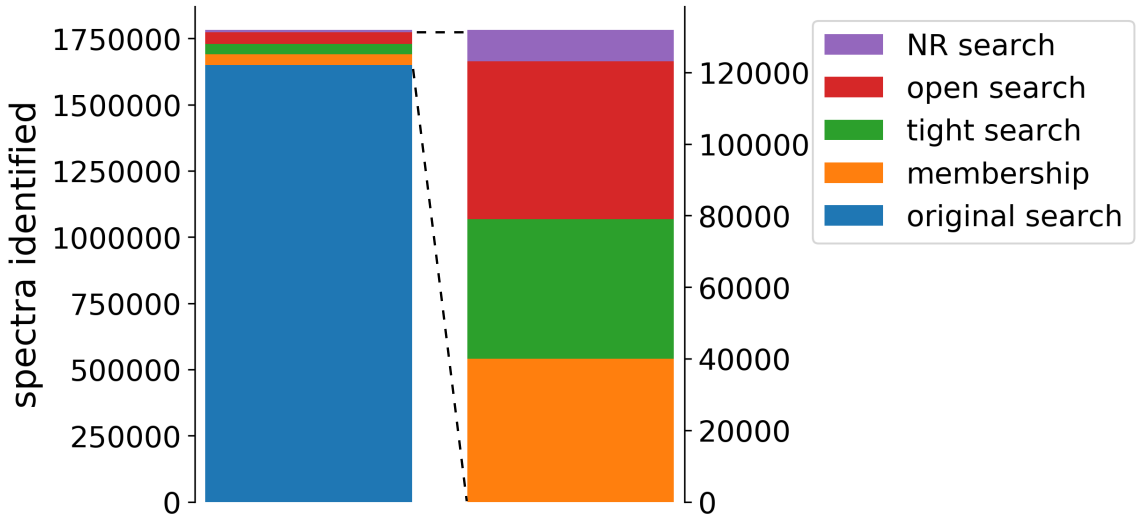

Figure 4.8: **Spectra newly identified using the embedding.** The original database searches of the repository spectra identified 1,650,587 spectra. The various methods of identification using the embedding identified a total of an additional 165,070 spectra, broken down by method as shown.

community, plus a contaminant database. This step identified 9,407 spectra, and we assigned peptide identifications to a total of 38,974 unidentified spectra by propagating these new identifications from hubs to spokes.

Second, we performed an "open" target-decoy database search (with a 500 Da precursor mass tolerance and only acetylated cysteine and potentially oxidized methionine as modifications) on the remaining 118,381 hub spectra. This step identified 12,501 hub spectra and, via propagation, an additional 44,113 spokes.

Finally, the remaining 74,268 unidentified hub spectra were searched against the full non-redundant (NR) database with "tight" tolerances. Due to the immense size of NR (101 GB), instead of searching a decoy database we used PeptideProphet[35] to estimate an identification probability for each peptide-spectrum match (PSM). At probability 95% or greater, 1,388 hub spectra and 7,361 spokes were identified.

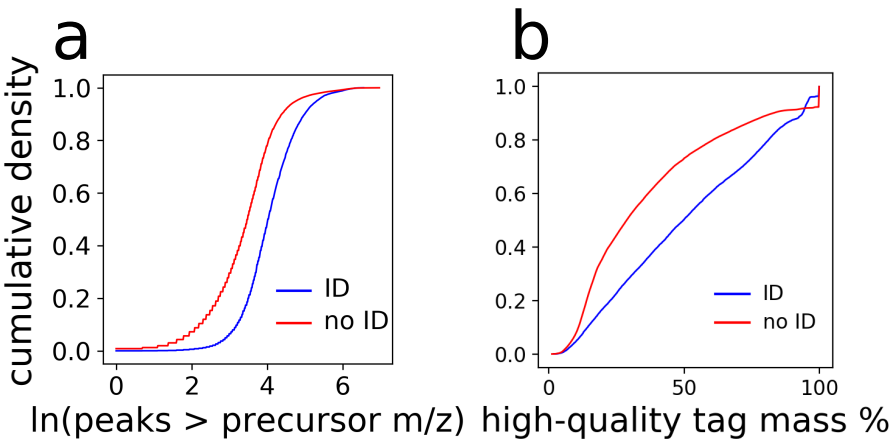

Figure 4.9: **Quality of identified and unidentified spectra.** Cumulative density functions for two proxies for spectrum quality, for spectra that were identified (blue line) or were not identified (red line) by database search. (A) The natural log of the one-padded count of fragment peaks higher than the precursor m/z. (B) The mass of the longest high quality sequence tag found by Novor as a percentage of the mass of the precursor ion.

Considering the additional identifications made by propagation of identifications in single-peptide communities and the two FDR-controlled searches ("tight" and "open"), we assigned identifications to 21,908 hub spectra representing communities containing 123,140 total spectra that were previously unidentified (7.46% of the originally unidentified spectra in the repository). Including the NR search results assigned confidence via PeptideProphet, we assigned identifications to 23,296 hub spectra representing communities containing 131,889 total spectra that were previously unidentified, or 7.99% of the number of originally identified spectra in the repository (Figure 4.8).

The remaining 104,495 unidentified hub spectra are of broadly lower quality than the hub spectra that were successfully identified, suggesting that a large proportion of these spectra may be fundamentally unidentifiable and may not have been generated by peptides. We used the count of fragment peaks of higher m/z as a rough proxy for spectrum quality, following a

common practice in choosing transitions for single reaction monitoring experiments.[50,87] By this metric, 26% of the hub spectra that remained unidentified were of lower quality than the 5th percentile by quality among the identified spectra, and unidentified spectra had lower quality overall ($p < 0.0001$ by one-tailed Mann-Whitney U test, see Figure 4.9a). We also searched the remaining unidentified hub spectra with the *de novo* search engine Novor[44], which detects and assigns quality scores to sequence 'tags': sequences of amino acids that match to a series of peaks within a spectrum. Compared with the identified spectra, high-quality tags from the unidentified spectra with quality score 36 or higher for each amino acid tended to account for a smaller proportion of the full peptide mass ($p < 0.0001$ by one-tailed Mann-Whitney U test, see Figure 4.9b). Novor also found high-quality full-length sequence tags, with a quality score of 36 or higher for each amino acid, for 8,953 hub spectra representing communities containing 32,643 spectra.

4.2.5 Exploring the benefits of scaling up

A key driver for this work is the idea that, as we scale from small data sets to big data sets, many analysis techniques undergo a "phase transition" in which questions that were previously hard to answer become significantly easier.[4,26] Specifically, we hypothesize that as the number of spectra embedded in our learned latent space grows, the cluster structure of the spectra in that space will help us to focus on and identify interesting spectra. For example, a "singleton" spectrum with no nearby neighbors in a small repository may be alone simply because the repository is small. But if we scale the repository up by an order of magnitude or more, then the remaining singleton spectra become relatively rare and much more likely to correspond to noise. Similarly, unlabeled spectral communities will become much less prevalent as we scale up the size of the repository, allowing us to focus on the most densely populated clusters of uncharacterized spectra.

To begin exploring the scaling behavior of our network, we randomly downsampled the number of spectra embedded into the space and investigated properties of the embedding and the spectral communities. Specifically, among the 790 mass spectrometry runs in our

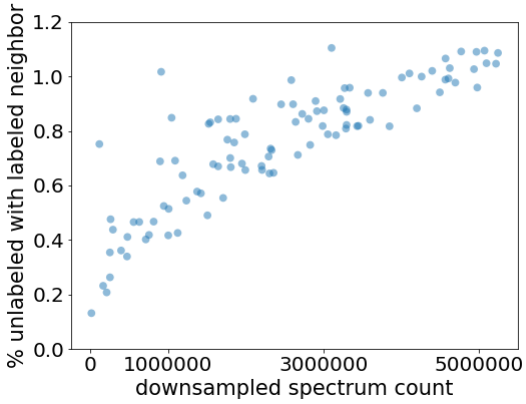

Figure 4.10: **A larger proportion of unidentified spectra have identified neighbors as repository size increases.** A scatter plot of the number of spectra retained (horizontal axis) versus the percentage of unidentified spectra that have an identified neighbor within τ, for 100 random downsamplings of our repository spectra.

repository, we randomly sampled 100 subsets of sizes uniformly distributed between 1 and 790. For each of these downsampled repositories, we counted the percentage of unidentified spectra that have a labeled neighbor within a fixed spectrum community distance threshold $\tau = 0.095$ (Figure 4.10). The results show a systematic increase in this percentage, without flattening as it approaches the full size of our repository.

4.3 Discussion

We have demonstrated the utility of the 32-dimensional embedding learned by GLEAMS. By mapping spectra from diverse experiments into a common latent space, we can rapidly add another additional 8.0% to the identifications derived from database search. Furthermore, the embedding can be used to detect mis-identified spectra and suggest corrections.

Notably, our embedder is able to learn from spectra labeled by database search even though those labels have a 1% FDR. We detected spectrum communities that contained spectra with different peptide identifications that appeared in fact to be generated by only

a single peptide. So-called "weak teacher" training is a well-established practice in machine learning,[92] including machine learning in proteomics.[21]

The embedding has potential utility beyond simply transferring identifications among nearby spectra. For example, it may be that the latent space encodes semantic relationships among spectra generated by related molecular species. If such relationships could be mapped, then it might be possible to, for instance, predict where in embedded space a spectrum generated by a peptide with a particular post-translational modification would be found, based on the known location of the unmodified species. Such semantic relationships have been uncovered previously using latent embeddings based on natural language.[68] It may also be possible to develop a joint embedding of peptide sequences and spectra, allowing arbitrary peptide sequences to be embedded. The embedded space could then be used like a search engine, assigning peptide identifications to spectra based on closeness to an embedded peptide sequence.

The learned embedding opens up possibilities for transfer learning. For example, it may be possible to train a separate neural network to predict a spectrum's quality, or potential for being identified, from its location in embedded space, or to classify spectra as "chimeric" (generated by more than one peptide) or not. Furthermore, the GLEAMS embedding may have potential for applications at the level of the mass spectrometry runs or entire experiments, using each experiment's embedded spectra to predict its tissue of origin or, in the case of metaproteomics experiments, taxonomic makeup.

A clear direction for future work is the development of statistical confidence estimation procedures suitable for this type of learned embedding. On the one hand, propagating peptide annotations between proximal pairs of spectra may risk introducing false positive assignments. On the other hand, when multiple identified spectra lie close to an unidentified spectrum, our confidence in such a propagation should intuitively increase relative to propagation with respect to a single spectrum-spectrum pair. Target-decoy methods for confidence estimation are widely used and provably correct under reasonable assumptions about the database search procedure,[28] but these methods do not generalize in a straightforward

fashion to a method based on propagation in the GLEAMS embedded space.

Compared with k-means and other similar clustering methods, the embedding approach is far more computationally efficient, enabling it to be scaled up to the size of an entire proteomics repository. Once the embedder is trained, new spectra representing previously unobserved peptides can be embedded and used for analysis without performing any expensive operations as long as they have sufficiently similar characteristics to the distribution of training spectra. The computational power required to find the nearest neighbors of a given spectrum in embedded space increases with the size of the repository, but this task can scale smoothly to billions of vectors by employing multiple GPUs. This approach makes it possible to assign new spectra to spectrum communities nearly instantaneously upon submission to a repository, giving researchers the immediate benefit of the combined analysis efforts of the entire proteomics community.

4.4 Methods

4.4.1 Encoding mass spectra for network input

Each spectrum is encoded as a vector of 3,556 features of three types: precursor attributes, binned fragment intensities and dot-product similarities with a set of reference spectra.

Precursor mass, m/z and charge are encoded as a combined 61 features. Precursor mass and m/z are each extremely important values for which precision is critical, and so they are poorly suited for encoding as single input features for a neural network. Accordingly, we experimented with a multiple binary encodings of precursor mass and m/z, each of which gave superior performance on validation data than a real-value encoding, and settled on the encoding that gave moderately better performance than the others: a 27-bit "Gray code" binary encoding, in which successive values differ by only a single bit, preserving locality and eliminating thresholds at which many bits are flipped at once. Precursor values may span the range 400 to 6,000, so the Gray code encoding has a resolution of 0.00004. Fragment values may span the range 50.5 to 2,500, so the Gray code encoding has a resolution of

0.00002. Spectrum charge is one-hot encoded: seven features represent charge states 1–7, all of which are set to 0 except the charge corresponding to the spectrum (spectra with charge 8 or higher are encoded as charge 7).

Fragment peaks are encoded as 2,449 features. Fragment intensities are square-root transformed and then normalized by dividing by the sum of the square-root intensities. Fragments outside the range 50.5–2,500 *m/z* are discarded, and the remaining fragments are binned into 2,449 bins at 1.0005079 *m/z*, corresponding to the distance between the centers of two adjacent clusters of physically possible peptide masses,[89] with bins offset by half a mass cluster separation width so that bin boundaries fall between peaks.

Similarities of each spectrum to an invariant set of reference spectra are encoded as 500 features. Each such features is the normalized dot product between the given spectrum and one of an invariant set of 500 reference spectra chosen randomly from the training and validation datasets. For dot product calculation, spectra are binned at 0.02 *m/z*, and each spectrum's binned representation is convolved with a Gaussian with σ estimated by Param-Medic.[48]

4.4.2 Neural network structure

The embedder network (Figure 4.1b) takes each of the three types of inputs separately. The precursor features are processed through a two-layer fully-connected network with layer dimensions 32 and 5, while the binned fragment and reference spectrum similarity features are each passed through a separate single-layer convolutional neural network (CNN. Filters: 30; kernel size: 3; stride length: 1) and then through a max pooling layer. Output of the three networks is concatenated and passed to a final fully-connected layer with dimension 32. All network layers were preceded by scaled exponential linear units (SELU) activation.

To train the embedder, we construct a "Siamese network" containing two instances of the embedder with tied weights W (Figure 4.1a). Pairs of spectra s_a and s_b with peptide labels derived from database search (see below) are fed to the embedders. The Euclidean distance between the two embeddings is calculated, and the contrastive loss function is

computed as in Equation 1. Intuitively, this loss function "pulls" the network outputs on same-peptide-labeled pairs ($Y = 1$) together by penalizing large distances and "pushes" different-peptide-labeled pairs ($Y = 0$) apart by penalizing small distances. The network is trained by stochastic gradient descent with the Adam update rule[40] and a learning rate of 0.0002, implemented with the Keras framework.[7] Training and evaluation were both performed on a single core of a 3.4GHz Intel Xeon processor with a single GeForce GTX 1080 graphics processing unit.

4.4.3 Training the embedder

We assembled a repository of more than five million mass spectra from 22 publicly available data-dependent acquisition experiments comprising 800 mass spectrometry acquisitions, representing a variety of instrument types and a variety of human tissues as well as mouse, yeast and microbiome samples. Experiment information, as well as search databases and parameters used in searching each experiment, are summarized in Table 1.

Spectra were matched to peptide sequences and PSMs accepted at FDR ≤ 0.01 as follows. A database search of each run was performed using Comet[14] version 2016.01 rev. 1. Search parameters included a static modification for cysteine carbamidomethylation (57.021464) and a variable modification for methionine oxidation (15.9949), with additional modifications as appropriate for each experiment (Table 1). Samples were searched against the appropriate UniProt databases for single organisms, Human Microbiome Project stool database for gut microbiome,[30] or a site-specific sequencing-derived database for ocean microbiome.[49] We used a concatenated decoy database in which peptide sequences were reversed but C-terminal amino acids left in place. Enzyme specificity was trypsin with proline cleavage suppression, with one missed cleavage allowed. Parent ion mass tolerance and fragment mass bin size were determined separately for each sample with Param-Medic[48]; parent ion tolerance was defined around five isotopic peaks. The top five search results for each spectrum were retained for generation of training data (see below); however, for spectrum identification only the top hit was retained. False discovery rate was calculated by target-decoy competition using

Percolator,[33] and PSMs were accepted at FDR ≤ 0.01.

The identified spectra were divided into training and validation pools by experiment, with roughly 1/5 of spectra reserved for validation. For each real spectrum, we used MS2PIP[10,11] to generate two theoretical spectra: a spectrum representing the same peptide in the same charge state, with the same modifications, and a spectrum representing a randomly chosen decoy database peptides identified in the top-five database search described above.

The network was trained on positive (same-peptide) and negative (different-peptide) pairs of spectra with precursor masses within 0.2 Da of one another. During each training epoch, all the real-and-theoretical positive and negative pairs were used. To prevent the network from learning to segregate real from theoretical spectra in embedded space, we also used all 97,771 positive pairs derived from the one million real spectra and, on each training epoch, a different random set of 100,000 negative real-real spectrum pairs and a random set of 100,000 negative theoretical-theoretical spectrum pairs (see below for alternative training data structures we considered). Training consisted of 60 such epochs.

The validation dataset was constructed in a similar manner to the construction of a single epoch of the training dataset, and validation data was fixed throughout training. To assess embedder performance during training, we ordered all validation pairs by embedded distance and calculated the area under a concentrated ROC (CROC) curve,[78] a warping of a standard ROC curve designed to emphasize discrimination at high specificity, with $\alpha = 14$. After training, the network weights from the epoch with the highest AUCROC were retained.

4.5 Exploring hyperparameter space and training data structure

To arrive at the model used in GLEAMS, we explored the space of model structure and hyperparameter settings extensively but not exhaustively. To compare trained models, we assessed their performance by the area under the concentrated receiver operator characteristic (CROC) curve[78] on held-out same-label and different-label pairs of spectra, as described in Methods. Most hyperparameters were selected based on the results of training on 100,000

pairs of spectra. However, the network structure (the numbers, types and sizes of layers used for each type of input feature, and the number of fully-connected layers after concatenation) was selected from a wide variety of structures all trained on one million pairs of spectra. The model presented had the highest CROC of all models considered. Below, we describe the various models and hyperparameter settings that we explored.

We considered different types and encodings of input features. The model using all three feature types (precursor, binned fragment and reference spectrum similarity) outperformed models using one or two of those feature types. We considered encoding precursor mass and m/z as single, real-value features (with or without scaling) and with an arbitrary binary encoding lacking the locality benefits of Gray Code. We also considered binning fragment features at 0.02 Da (convolving peak intensities with a Gaussian representing estimated fragment measurement error); the resulting enormous number of features was a great impediment to training. We further considered using the hashing trick (defining a hash function to map large numbers of features to a smaller number of features, with collisions) to reduce this dimensionality to 2,000, 4,000 or 6,000 features, which improved performance but still lagged behind the 1 m/z binning. We considered using 500 and 1000 reference spectrum similarity features.

We considered many different structures for the embedder model. For each input type (precursor, fragment, reference spectrum) we considered one to three fully-connected layers of various sizes, one to three convolutional layers followed by max pooling, a recurrent neural network, a dense layer followed by convolutional layers, and long short-term memory (LSTM) layers (single-directional and bidirectional). We also considered one to four fully-connected layers after concatenation of the network outputs from the three input types. Surprisingly, deeper networks generally trained more slowly and also reached lower final AUCROCs: the only input type that benefited from more than a single layer was the precursor input type.

We considered several values for the hyperparameters associated with the convolutional neural networks (CNNs): number of filters (20, 30 or 50), kernel size (2,3,4), stride length (1,2), pooling kernel size (1,2) and pooling stride length (1,2). Of particular note, we discov-

ered that the size of the last layer on the precursor features needed to be small (we settled on five) compared to the number of filters (we settled on 30) used in the CNNs on the binned fragment and reference spectrum features.

We considered several different nonlinearities for all network layers: ELU, ReLU, SELU, PReLU and sigmoid, as well as linear activation. We considered training with a fixed learning rate, as well as with Adam, RMSprop, and Adagrad using several learning rates.

We considered batch normalization and dropout with proportion 0.0005 to 0.2, and L1 and L2 regularization. All decreased performance.

We considered several sizes for the embedded dimension: 8, 16, 24, 32, 64 and 128. Higher dimensionality gave monotonically higher CROC but slowed down operations on the embedded spectra such as k-nearest-neighbor search. The improvement from 24 to 32 was substantial, and the improvement from 32 to 64 was minimal.

We considered four approaches to training and validation data set construction before settling on the combination of observed and theoretical spectra described in Methods. First, we used only pairs of observed spectra with the same or different peptide labels, with no further restrictions. Second, we imposed a 3 Da maximum on the difference between the two precursor masses. The second approach led to higher AUCROC than the first approach, even when using a validation set without the precursor mass restriction. We suspect this improvement arose because, without the restriction, too many of the different-label pairs were "too easy," having many differences between the spectra and insufficiently representing the difficult task of discriminating pairs of spectra that share more characteristics. This observation led us to our third approach, in which we used same-label pairs of real spectra with precursor masses within 0.2 Da and different-label pairs between observed spectra and theoretical spectra generated by MS2PIP[10,11] representing decoy peptides from the top five search results from Comet search. With this approach the network learned how to separate real from theoretical spectra but did not learn as well to separate positive- and negative-label pairs of real spectra. To address this issue, we developed our fourth and final method, described in Methods, in which we added to the third approach different-labeled pairs of real

spectra and different-labeled pairs of theoretical spectra.

4.5.1 Detecting spectrum communities

To detect spectrum communities, we designed an algorithm (Algorithm 1) to greedily select communities in which a single spectrum (the "hub") is close to many neighbors (the "spokes") in the embedded space. The community selection proceeds in four steps.

First, we embedded all spectra from our repository and constructed an approximate k-nearest neighbor graph in which each node is a spectrum. This construction was carried out using the GPU-enabled Faiss library for efficient similarity search,[31] using the IVFFlat inverted file index type. FAISS produces an index containing all embedded spectra, which we queried to find the k-nearest neighbors (by Euclidean distance) of each embedded spectrum. We chose k to be as large as possible (1000) while still maintaining reasonably fast computation.

Second, we reduced this k-nearest neighbor graph by eliminating edges longer than a specified distance threshold τ (our method for determining τ is described below). In this step, 4,568,161 spectra with zero nearby neighbors were eliminated. These spectra presumably represent peptides that were observed only once, peptides that should have been merged into a community but were not due to the conservative setting of the distance threshold, and non-peptide molecular species.

Third, we selected communities in the graph in a greedy fashion. To do so, we induced an ordering on the nodes in the graph, using a two-factor sort where the primary sort key is the node degree, and the secondary sort key is the mean edge distance. We called the node with the most nearby neighbors in this list a "hub" and marked all its neighbors as members of its community. Similarly, for each subsequent node in the list, if the node and at least one of its neighbors were not already assigned to spectrum communities, then we declared the node to be the hub of a new community and its neighbors to be the other members of

the community.[1]

Finally, a fourth, post-processing step was carried out to merge nearby hubs. This step is necessary because of the limited value of k (which is much smaller than the number of nodes) and the approximate nearest-neighbor algorithm used. The greedy process left 145 hub nodes with other hub nodes within the distance threshold τ. We sorted these nodes in ascending order by degree. In this order, for each hub node h_1 and its closest neighbor h_2 of the same or higher degree, we merged h_1 into the community with hub h_2. We also added all neighbors of h_1 to the community with hub h_2 if they were within Euclidean distance τ of h_2.

We experimented with different values for the distance threshold τ until we found a value such that, among communities that contain at least one identified spectra, 99% were represented by a unique peptide sequence, treating the isobaric leucine and isoleucine residues as a single amino acid for purposes of comparison.

Alternative community detection approaches

We considered alternative approaches for detecting communities of mass spectra and found our hub-and-spoke approach to be superior. k-means clustering is a commonly used iterative method for finding clusters in multidimensional space. It relies on a single parameter k, the number of clusters. Since we didn't know the expected number of spectrum communities *a priori*, we experimented with a range of values for k using the scikit-learn and Faiss implementations of k-means clustering. Unfortunately, because k-means clustering operates on all spectra simultaneously, clustering of the entire repository proved intractable with a value of k high enough to detect spectrum communities effectively.

We also considered a method for spectrum community detection based on the k-clique-communities algorithm. In this approach, we defined a 1000-nearest-neighbors graph within

[1]This approach is similar to submodular maximization via the greedy procedure for the set cover function, with the difference that in our approach the potential hubs are not reordered after each hub-to-spoke assignment. An exploration of the effects of such small changes to the community assignment algorithm is reserved for future work.

a distance threshold chosen such that 99.95% of the pairs of identified within the threshold were same-peptide pairs. We then detected all of the k-clique-communities in this graph: connected components such that every node in the component is a member of a k-clique with $k-1$ other members of the component. We evaluated this approach for $k = 5$ and $k = 6$.

Figure 4.3b compares the hub-and-spoke method (for values of τ ranging from 0.095 to 0.12) with the k-means clustering method (for values of k ranging from 150,000 to 500,000) and the k-clique-communities method (with $k = 5$ and $k = 6$) in terms of the number of single-peptide and multi-peptide communities, showing a clear advantage for the hub-and-spoke method: the k-clique-communities method detects far fewer communities with a higher proportion of spectra in multi-peptide communities, while the proportion of multi-peptide communities remains far too high at any practical value of k. Using k-means clustering with an appropriate value of k raises insurmountable runtime issues: while the hub-and-spoke and k-clique-communities methods can be parallelized by running on subsets of spectra defined by similar precursor mass, dividing the spectra for k-means clustering requires a separate choice of k that must be optimized for each subset of spectra, which is practically intractable and prone to failure. For this reason k-means clustering does not appear to be scalable to an order of magnitude more spectra in a full-size repository (Figure 4.6 shows running times for the scikit-learn implementation of k-means, which was only slightly slower than the Faiss implementation on our data). In fact, we were effectively unable to run k-means clustering on all of our repository spectra at once, and so Figure 4.3b compares the three methods for only the 3,390,759 charge 2 spectra.

We considered evaluating other clustering approaches for assigning spectrum communities, such as spectral clustering and hierarchical clustering. However, like k-means clustering, those approaches are typically applied to problems in which a complete partitioning of the space is desired, whereas in our case we expect the majority of spectra to occupy singleton clusters.

4.5.2 Spectrum community peptide annotation

We used the communities to annotate previously unidentified spectra. First, for communities associated with a single peptide, we propagated this peptide to all unidentified spectra within the community. Second, for communities with no associated peptide, we performed a sequence of four increasingly broad searches to identify the hub spectra. If the searches were successful, then the resulting peptide was propagated from the hub to all members of those communities.

First, we performed a "tight" search of each hub spectrum against databases for all organisms that might have generated it, in order to identify spectra that had previously been searched against the wrong database. Hub spectra were separated into four groups by their fragment mass error as estimated by Param-Medic. "High-precursor-accuracy" precursors with a predicted error of 50 ppm or less were searched with precursor mass error 50 ppm, and "low-precursor-accuracy" spectra with higher predicted error were searched with precursor mass error 150 ppm. "High-fragment-accuracy" fragments with a predicted bin size of 0.02 or less were searched with fragment bin size 0.02, and "low-fragment-accuracy" fragment spectra with higher predicted bin size were searched with fragment bin size 1.0005. Each hub spectrum was searched with appropriate parameters separately against the target and decoy database appropriate for any organism associated (by experimental metadata) with any spectrum in its spectral community, as well as the Global Proteome Machine cRAP database of common contaminants. Search results were combined, and spectra were re-ranked and FDR estimated using Percolator. Search results passing 1% FDR threshold were retained.

Next, hub spectra that remained unidentified at 1% FDR were searched against all appropriate target and decoy organism databases with appropriate fragment bin widths and a wide (500 Da) precursor mass tolerance. Search results were combined as before, FDR estimated with Percolator, and search results retained at 1% FDR.

Hub spectra that remained unidentified at 1% FDR were searched with appropriate pre-

cursor mass tolerance and fragment bin size against the entire NCBI NR database (downloaded October 27, 2018) with no decoy sequences. Peptide identifications probability was estimated with PeptideProphet, and all identifications assigned probability greater than 0.95 were retained. The same spectra were also searched with the *de novo* search engine Novor v1.06.0634 with 50 ppm and 150 ppm precursor error tolerances for "high-accuracy" and "low-accuracy" precursors, and 0.02 m/z and 0.5 m/z fragment error tolerances for "high-accuracy" and "low-accuracy" fragments, respectively. For each spectrum, the longest contiguous tag with amino acid scores all greater than or equal to 36 was retained.

Experiment	**Instrument**	**Organism**	**Additional Search Parameters**
Training Datasets			
2013poulsen-PXD000307	TripleTOF	human	
2014kim-kidney[39]	Orbitrap Velos	human	
2014kim-lung[39]	Orbitrap Elite	human	
2014kim-adrenalgland[39]	Orbitrap Velos	human	
2014kim-monocytes[39]	Orbitrap Velos	human	
2014kim-rectum[39]	Orbitrap Velos	human	
2014kim-gut[39]	Orbitrap Velos	human	
2014kim-fetalovary[39]	Orbitrap Elite	human	
2014kim-fetalplacenta[39]	Orbitrap Elite	human	
2015clark-redefining[8]	LTQ Orbitrap	human	TMT 6-plex
2015tanca-impact[80]	Orbitrap Velos	human gut microbiome	
2015uszkoreit-intuitive[84]	Orbitrap Elite	mouse	
2016mann-unpublished	QExactive	human	
2016may-metapeptides[49]	QExactive	ocean microbiome	
2016saraf-dynamic[76]	LTQ-Orbitrap	human	
2016zhong-quantitative[95]	Orbitrap Velos	human	
Test Datasets			
2014kim-cd4tcell[39]	Orbitrap Elite	human	
2014kim-adultovary[39]	Orbitrap Elite	human	
2014kim-eart[39]	Orbitrap Elite	human	
2015radoshevich-isg15[72]	QExactive	human	
2016audain-in-depth[2]	LTQ Orbitrap	yeast	
2016schittmayer-cleaning[77]	Orbitrap Velos	yeast	

Table 4.1: **Experiments used in the training and validation of the embedder.**

Algorithm 1 Hub-and-spoke spectrum community detection. First, 'hub' spectra are associated with their 'spoke' neighbors (based on k-nearest neighbors search) within distance threshold τ in a greedy fashion, with the most-connected hubs chosen first. Then adjacent hub-and-spoke communities are combined if their hubs are within τ.

Input: Dictionary $\mathcal{D}_N$ mapping each spectrum to its neighbors within distance τ.

$S \leftarrow \emptyset$ ▷ S accumulates a set of assigned 'spoke' spectra.
$\mathcal{D}_{HS} \leftarrow \emptyset$ ▷ $\mathcal{D}_{HS}$ is a dictionary mapping each hub to a set of spokes.
Sort keys of $\mathcal{D}_N$ first by (1) neighbor count, and then by (2) mean neighbor distance (both descending).
for $d \in$ keys of $\mathcal{D}_N$ in order **do**
    if $d \notin S$ **then**
        $S_d \leftarrow \{n : n \in \mathcal{D}_N(d),\ n \notin S, \text{ and } n \notin \mathcal{D}_{HS}\}$
        if $S_d \neq \emptyset$ **then**
            $\mathcal{D}_{HS}(d) = S_d$
            $S \leftarrow S \cup S_d$
$H \leftarrow$ keys of $\mathcal{D}_{HS}$ ▷ H is the list of hub spectra
Sort H by number of spokes per hub (ascending).
for $h_1 \in H$ in order **do**
    $N_H \leftarrow \{n : n \in \mathcal{D}_{HS}(h_1), n \in H \text{ and } |\mathcal{D}_{HS}(n)| \geq |\mathcal{D}_{HS}(h_1)|\}$
    if $N_H \neq \emptyset$ **then**
        $h_2 = \arg\min_{x \in N_H} ||h_1 - x||$ ▷ h_2 is the closest hub neighbor of h_1 with at least as many spokes
        $\mathcal{D}_{HS}(h_2) \leftarrow \mathcal{D}_{HS}(h_2) \cup \{h_1\}$
        for $s \in \mathcal{D}_{HS}(h_1)$ **do**
            if $||h_2 - s|| < \tau$ **then**
                $\mathcal{D}_{HS}(h_2) \leftarrow \mathcal{D}_{HS}(h_2) \cup \{s\}$
        remove h_1 from $\mathcal{D}_{HS}$
return $\mathcal{D}_{HS}$

Chapter 5

CONCLUSION

The field of mass spectrometry proteomics has come a long way from the early days of painstakingly, manually interpreting individual spectra. However, although interpretation of mass spectra has been made dramatically more consistent and sped up by orders of magnitude, the general approach remains the same: compare the peaks of an observed mass spectrum directly with the peaks of theoretical spectra generated for candidate peptides.

New computational advances and growing collections of public data offer the potential for radically different ways of interpreting mass spectra. As described in Chapter 4, spectra can be embedded into a low-dimensional space in which spectra generated by the same peptide are close together. This space can then be used to find nearby neighbors of newly acquired spectra, and this association can be used to propagate identifications from one spectrum to another. The embedding approach can also be used to detect dense communities of "dark matter" – spectra representing molecular species that are observed repeatedly in diverse mass spectrometry runs but consistently fail to be identified – and focus computational resources on those spectra in order to identify many spectra at once.

Implementing approaches like this on the scale of entire proteomics repositories like PRIDE and Massive, or even networks of repositories such as ProteomeXchange, would fundamentally shift the computational proteomics landscape, moving the computational burden from individual research labs to central repositories and providing researchers instantly with the benefit of decades' worth of mass spectrometry experimentation.

Currently, each proteomics laboratory is tasked with choosing or developing a computational analysis pipeline appropriate to their data. Many laboratories that excel at experimental design and benchwork lack the computational resources to analyze their own data properly. Computationally-focused laboratories may be able to identify their spectra with higher sensitivity and proper control of false discovery rate, but a profusion of available tools and approaches creates a fragmented landscape.

The proteomics community now has the data and the methods it needs to centralize the routine identification of mass spectra. Rather than submitting experimental data and results to a public repository at the end of analysis, as a requirement for manuscript submission,

researchers could submit their raw data files immediately upon acquisition. These files could be processed by centralized resources, leveraging the hundreds of millions of mass spectra already collected by researchers all over the world to identify new spectra with minimal computational investment. These results could be made available to researchers within minutes of data submission, providing a jumpstart to the data analysis process. Laboratories not wanting to become specialists in spectrum identification could essentially outsource that part of their workflow to the community.

To reach this goal, the community must take several steps. First, an approach such as the embedding I describe must be trained on an order of magnitude more data, so that the training set encompasses the great majority of variation in mass spectra. Then, the trained model must be used to embed every mass spectrum – identified or unidentified – in an entire repository such as MassIVE or PRIDE. As we demonstrated, this dramatic scaling up will result in a far higher proportion of unidentified mass spectra embedded near identified neighbors, incurring an immediate benefit for researchers who have already submitted data: instantaneous identification of many already-submitted spectra. Finally, an application programming interface must be exposed to allow any researcher to submit their own spectra – privately, if they prefer – for immediate annotation using the resources of the community. These further steps will move mass spectrometry data analysis from a fragmented task implemented painstakingly and unevenly in individual labs to a more sensitive, consistent approach making effective use of our community's vast data resources.

BIBLIOGRAPHY

[1] Steven D Allison and Jennifer B H Martiny. Colloquium paper: resistance, resilience, and redundancy in microbial communities. *Proceedings of the National Academy of Sciences of the United States of America*, 105 Suppl(Supplement 1):11512–9, 2008.

[2] Enrique Audain, Julian Uszkoreit, Timo Sachsenberg, Julianus Pfeuffer, Xiao Liang, Henning Hermjakob, Aniel Sanchez, Martin Eisenacher, Knut Reinert, David L. Tabb, Oliver Kohlbacher, and Yasset Perez-Riverol. In-depth analysis of protein inference algorithms using multiple search engines and well-defined metrics. *Journal of Proteomics*, 150:170–182, 2016.

[3] F Azam, T Fenchel, J G Field, J C Gray, L A Meyer-Reil, and F Thingstad. The ecological role of water-column microbes in the sea. *Marine Ecology Progress Series*, 10(3):257–264, 1983.

[4] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. *ACL '01 Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 26–33, 2001.

[5] Halima Bensmail, Jennifer Golek, Michelle M. Moody, John O. Semmes, and Abdelali Haoudi. A novel approach for clustering proteomics data using Bayesian fast Fourier transform. *Bioinformatics*, 21(10):2210–2224, 2005.

[6] Brandi L. Cantarel, Alison R. Erickson, Nathan C. VerBerkmoes, Brian K. Erickson, Patricia A. Carey, Chongle Pan, Manesh Shah, Emmanuel F. Mongodin, Janet K. Jansson, Claire M. Fraser-Liggett, and Robert L. Hettich. Strategies for metagenomic-guided whole-community proteomics of complex microbial environments. *PLoS ONE*, 6(11), 2011.

[7] François Chollet et al. Keras. `https://keras.io`, 2015.

[8] David J. Clark, William E. Fondrie, Zhongping Liao, Phyllis I. Hanson, Amy Fulton, Li Mao, and Austin J. Yang. Redefining the Breast Cancer Exosome Proteome by Tandem Mass Tag Quantitative Proteomics and Multivariate Cluster Analysis. *Analytical Chemistry*, 87(20):10462–10469, 2015.

[9] Murray P Cox, Daniel A Peterson, and Patrick J Biggs. SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data. *BMC Bioinformatics*, 11(1):485, 2010.

[10] Sven Degroeve, Davy Maddelein, and Lennart Martens. MS2PIP prediction server: Compute and visualize MS2peak intensity predictions for CID and HCD fragmentation. *Nucleic Acids Research*, 43(W1):W326–W330, 2015.

[11] Sven Degroeve, Lennart Martens, and Igor Jurisica. MS2PIP: A tool for MS/MS peak intensity prediction. *Bioinformatics*, 29(24):3199–3203, 2013.

[12] A. Dutta, A. Gupta, and A. Zissermann. VGG image annotator (VIA). http://www.robots.ox.ac.uk/ vgg/software/via/, 2016.

[13] 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.

[14] 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.

[15] Jimmy K Eng, Ashley L Mccormack, and John R Yates. An Approach to Correlate Tandem Mass Spectral Data of Peptides with Amino Acid Sequences in a Protein Database. *American society for Mass Spectrometry*, 5:976–989, 1994.

[16] Brent Ewing, Ladeana Hillier, Michael C Wendl, and Phil Green. Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities. *Genome Research*, (206):175–185, 2005.

[17] Terry Farrah, Eric W. Deutsch, Michael R. Hoopmann, Janice L. Hallows, Zhi Sun, Chung Ying Huang, and Robert L. Moritz. The state of the human proteome in 2012 as viewed through PeptideAtlas. *Journal of Proteome Research*, 12(1):162–171, 2013.

[18] David Fenyö, Jan Eriksson, and Ronald Beavis. Mass Spectrometric Protein Identification Using the Global Proteome Machine. *Methods Mol Biol*, 673:1–13, 2010.

[19] Ari M Frank, Nuno Bandeira, Zhouxin Shen, Stephen Tanner, Steven P Briggs, Richard D Smith, and Pavel A Pevzner. Clustering Millions of Tandem Mass Spectra research articles. *J. Proteome Research*, pages 113–122, 2008.

[20] Anna A Georges, Heba El-Swais, Susanne E Craig, William KW Li, and David A Walsh. Metaproteomic analysis of a winter to spring succession in coastal northwest Atlantic Ocean microbial plankton. *The ISME Journal*, 8(6):1301–1313, 2014.

[21] Giulia Gonnelli, Michiel Stock, Jan Verwaeren, Davy Maddelein, Bernard De Baets, Lennart Martens, and Sven Degroeve. A decoy-free approach to the identification of peptides. *Journal of Proteome Research*, 14(4):1792–1798, 2015.

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

[23] Johannes Griss, Joseph M Foster, Henning Hermjakob, and Juan Antonio Vizcaíno. PRIDE Cluster: building the consensus of proteomics data. *Nature Methods*, 10(2):95–96, 2013.

[24] Johannes Griss, Yasset Perez-Riverol, Steve Lewis, David L Tabb, José A Dianes, Noemi Del-Toro, Marc Rurik, Mathias Walzer, Oliver Kohlbacher, Henning Hermjakob, Rui

Wang, and Juan Antonio Vizcaíno. Recognizing millions of consistently unidentified spectra across hundreds of shotgun proteomics datasets. *Nature Methods*, 13(8):651–656, 2016.

[25] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:1735–1742, 2006.

[26] Alon Halevy, Peter Norvig, and Fernando Pereira. The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

[27] Alyse K. Hawley, Heather M. Brewer, Angela D. Norbeck, Ljiljana Paša-Tolic, and Steven J. Hallam. Metaproteomics reveals differential modes of metabolic coupling among ubiquitous oxygen minimum zone microbes. *Proceedings of the National Academy of Sciences*, 111(31):11395–11400, 2014.

[28] 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.

[29] Katharina J. Hoff, Thomas Lingner, Peter Meinicke, and Maike Tech. Orphelia: Predicting genes in metagenomic sequencing reads. *Nucleic Acids Research*, 37(SUPPL. 2):101–105, 2009.

[30] Curtis Huttenhower, Dirk Gevers, Rob Knight, Sahar Abubucker, Jonathan H. Badger, Asif T. Chinwalla, Heather H. Creasy, Ashlee M. Earl, Michael G. FitzGerald, Robert S. Fulton, Michelle G. Giglio, Kymberlie Hallsworth-Pepin, Elizabeth A. Lobos, Ramana Madupu, Vincent Magrini, John C. Martin, Makedonka Mitreva, Donna M. Muzny, Erica J. Sodergren, James Versalovic, Aye M. Wollam, Kim C. Worley, Jennifer R. Wortman, Sarah K. Young, Qiandong Zeng, Kjersti M. Aagaard, Olukemi O. Abolude, Emma Allen-Vercoe, Eric J. Alm, Lucia Alvarado, Gary L. Andersen, Scott

Anderson, Elizabeth Appelbaum, Harindra M. Arachchi, Gary Armitage, Cesar A. Arze, Tulin Ayvaz, Carl C. Baker, Lisa Begg, Tsegahiwot Belachew, Veena Bhonagiri, Monika Bihan, Martin J. Blaser, Toby Bloom, Vivien Bonazzi, J. Paul Brooks, Gregory A. Buck, Christian J. Buhay, Dana A. Busam, Joseph L. Campbell, Shane R. Canon, Brandi L. Cantarel, Patrick S. G. Chain, I-Min A. Chen, Lei Chen, Shaila Chhibba, Ken Chu, Dawn M. Ciulla, Jose C. Clemente, Sandra W. Clifton, Sean Conlan, Jonathan Crabtree, Mary A. Cutting, Noam J. Davidovics, Catherine C. Davis, Todd Z. DeSantis, Carolyn Deal, Kimberley D. Delehaunty, Floyd E. Dewhirst, Elena Deych, Yan Ding, David J. Dooling, Shannon P. Dugan, Wm Michael Dunne, A. Scott Durkin, Robert C. Edgar, Rachel L. Erlich, Candace N. Farmer, Ruth M. Farrell, Karoline Faust, Michael Feldgarden, Victor M. Felix, Sheila Fisher, Anthony A. Fodor, Larry J. Forney, Leslie Foster, Valentina Di Francesco, Jonathan Friedman, Dennis C. Friedrich, Catrina C. Fronick, Lucinda L. Fulton, Hongyu Gao, Nathalia Garcia, Georgia Giannoukos, Christina Giblin, Maria Y. Giovanni, Jonathan M. Goldberg, Johannes Goll, Antonio Gonzalez, Allison Griggs, Sharvari Gujja, Susan Kinder Haake, Brian J. Haas, Holli A. Hamilton, Emily L. Harris, Theresa A. Hepburn, Brandi Herter, Diane E. Hoffmann, Michael E. Holder, Clinton Howarth, Katherine H. Huang, Susan M. Huse, Jacques Izard, Janet K. Jansson, Huaiyang Jiang, Catherine Jordan, Vandita Joshi, James A. Katancik, Wendy A. Keitel, Scott T. Kelley, Cristyn Kells, Nicholas B. King, Dan Knights, Heidi H. Kong, Omry Koren, Sergey Koren, Karthik C. Kota, Christie L. Kovar, Nikos C. Kyrpides, Patricio S. La Rosa, Sandra L. Lee, Katherine P. Lemon, Niall Lennon, Cecil M. Lewis, Lora Lewis, Ruth E. Ley, Kelvin Li, Konstantinos Liolios, Bo Liu, Yue Liu, Chien-Chi Lo, Catherine A. Lozupone, R. Dwayne Lunsford, Tessa Madden, Anup A. Mahurkar, Peter J. Mannon, Elaine R. Mardis, Victor M. Markowitz, Konstantinos Mavromatis, Jamison M. McCorrison, Daniel McDonald, Jean McEwen, Amy L. McGuire, Pamela McInnes, Teena Mehta, Kathie A. Mihindukulasuriya, Jason R. Miller, Patrick J. Minx, Irene Newsham, Chad Nusbaum, Michelle O'Laughlin, Joshua Orvis, Ioanna Pagani, Krishna Palaniappan, Shital M. Patel, Matthew Pearson,

Jane Peterson, Mircea Podar, Craig Pohl, Katherine S. Pollard, Mihai Pop, Margaret E. Priest, Lita M. Proctor, Xiang Qin, Jeroen Raes, Jacques Ravel, Jeffrey G. Reid, Mina Rho, Rosamond Rhodes, Kevin P. Riehle, Maria C. Rivera, Beltran Rodriguez-Mueller, Yu-Hui Rogers, Matthew C. Ross, Carsten Russ, Ravi K. Sanka, Pamela Sankar, J. Fah Sathirapongsasuti, Jeffery A. Schloss, Patrick D. Schloss, Thomas M. Schmidt, Matthew Scholz, Lynn Schriml, Alyxandria M. Schubert, Nicola Segata, Julia A. Segre, William D. Shannon, Richard R. Sharp, Thomas J. Sharpton, Narmada Shenoy, Nihar U. Sheth, Gina A. Simone, Indresh Singh, Christopher S. Smillie, Jack D. Sobel, Daniel D. Sommer, Paul Spicer, Granger G. Sutton, Sean M. Sykes, Diana G. Tabbaa, Mathangi Thiagarajan, Chad M. Tomlinson, Manolito Torralba, Todd J. Treangen, Rebecca M. Truty, Tatiana A. Vishnivetskaya, Jason Walker, Lu Wang, Zhengyuan Wang, Doyle V. Ward, Wesley Warren, Mark A. Watson, Christopher Wellington, Kris A. Wetterstrand, James R. White, Katarzyna Wilczek-Boney, YuanQing Wu, Kristine M. Wylie, Todd Wylie, Chandri Yandava, Liang Ye, Yuzhen Ye, Shibu Yooseph, Bonnie P. Youmans, Lan Zhang, Yanjiao Zhou, Yiming Zhu, Laurie Zoloth, Jeremy D. Zucker, Bruce W. Birren, Richard A. Gibbs, Sarah K. Highlander, Barbara A. Methé, Karen E. Nelson, Joseph F. Petrosino, George M. Weinstock, Richard K. Wilson, and Owen White. Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207–214, 2012.

[31] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *arXiv preprint*, 2017.

[32] 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.

[33] Lukas Käll, Jesse D Canterbury, Jason Weston, William Stafford Noble, and Michael J MacCoss. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nature methods*, 4(11):923–925, 2007.

[34] Katharina M. Keiblinger, Inés C. Wilhartitz, Thomas Schneider, Bernd Roschitzki, Emanuel Schmid, Leo Eberl, Kathrin Riedel, and Sophie Zechmeister-Boltenstern. Soil metaproteomics - Comparative evaluation of protein extraction protocols. *Soil Biology and Biochemistry*, 54:14–24, 2012.

[35] Andrew Keller, Alexey I. Nesvizhskii, Eugene Kolker, and Ruedi Aebersold. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Analytical Chemistry*, 74(20):5383–5392, 2002.

[36] Jeanette D. Kennelly, Felicity A. Baker, and Barbara A. Daveson. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781v3*, 2013.

[37] 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.

[38] Yong J. Kil, Christopher Becker, Wendy Sandoval, David Goldberg, and Marshall Bern. Preview: A program for surveying shotgun proteomics tandem mass spectrometry data. *Analytical Chemistry*, 83(13):5259–5267, 2011.

[39] Min-Sik Kim, Sneha M. Pinto, Derese Getnet, Raja Sekhar Nirujogi, Srikanth S. Manda, Raghothama Chaerkady, Anil K. Madugundu, Dhanashree S. Kelkar, Ruth Isserlin, Shobhit Jain, Joji K. Thomas, Babylakshmi Muthusamy, Pamela Leal-Rojas, Praveen Kumar, Nandini A. Sahasrabuddhe, Lavanya Balakrishnan, Jayshree Advani, Bijesh George, Santosh Renuse, Lakshmi Dhevi N. Selvan, Arun H. Patil, Vishalakshi Nanjappa, Aneesha Radhakrishnan, Samarjeet Prasad, Tejaswini Subbannayya, Rajesh Raju, Manish Kumar, Sreelakshmi K. Sreenivasamurthy, Arivusudar Marimuthu, Gajanan J. Sathe, Sandip Chavan, Keshava K. Datta, Yashwanth Subbannayya, Apeksha Sahu, Soujanya D. Yelamanchi, Savita Jayaram, Pavithra Rajagopalan, Jyoti Sharma, Krishna R. Murthy, Nazia Syed, Renu Goel, Aafaque A. Khan, Sartaj Ahmad, Gourav Dey, Keshav Mudgal, Aditi Chatterjee, Tai-Chung Huang, Jun Zhong, Xinyan Wu,

Patrick G. Shaw, Donald Freed, Muhammad S. Zahari, Kanchan K. Mukherjee, Subramanian Shankar, Anita Mahadevan, Henry Lam, Christopher J. Mitchell, Susarla Krishna Shankar, Parthasarathy Satishchandra, John T. Schroeder, Ravi Sirdeshmukh, Anirban Maitra, Steven D. Leach, Charles G. Drake, Marc K. Halushka, Keshava Prasad, Ralph H. Hruban, Candace L. Kerr, Gary D. Bader, Christine A. Iacobuzio-Donahue, Harsha Gowda, and Akhilesh Pandey. A draft map of the human proteome A. *Nature*, 509(7502):575–581, 2014.

[40] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, pages 1–15, 2015.

[41] David Kirchman. *Microbial ecology of the oceans.* John Wiley & Sons, 2010.

[42] Jens Roat Kultima, Shinichi Sunagawa, Junhua Li, Weineng Chen, Hua Chen, Daniel R. Mende, Manimozhiyan Arumugam, Qi Pan, Binghang Liu, Junjie Qin, Jun Wang, and Peer Bork. MOCAT: A Metagenomics Assembly and Gene Prediction Toolkit. *PLoS ONE*, 7(10):1–6, 2012.

[43] Henry Lam, Eric W. Deutsch, James S. Eddes, Jimmy K. Eng, Nichole King, Stephen E. Stein, and Ruedi Aebersold. Development and validation of a spectral library searching method for peptide identification from MS/MS. *Proteomics*, 7(5):655–667, 2007.

[44] Bin Ma. Novor: Real-Time Peptide de Novo Sequencing Software. *Journal of The American Society for Mass Spectrometry*, 26(11):1885–1894, 2015.

[45] Chunwei Ma, Yan Ren, Jiarui Yang, Zhe Ren, Huanming Yang, and Siqi Liu. Improved Peptide Retention Time Prediction in Liquid Chromatography through Deep Learning. *Analytical Chemistry*, 90:10881–10888, 2018.

[46] Lennart Martens, Henning Hermjakob, Philip Jones, Marcin Adamsk, Chris Taylor, David States, Kris Gevaert, Joël Vandekerckhove, and Rolf Apweiler. PRIDE: The proteomics identifications database. *Proteomics*, 5(13):3537–3545, 2005.

[47] Jelle Matthijnssens, Max Ciarlet, Mustafizur Rahman, Houssam Attoui, Mary K Estes, Jon R Gentsch, Miren Iturriza-gómara, Carl Kirkwood, Peter P C Mertens, Osamu Nakagomi, John T Patton, and M Franco. Elimination of Systematic Mass Measurement Errors in Liquid Chromatography-Mass Spectrometry Based Proteomics using Regression Models and a priori Partial Knowledge of the Sample Content. *Analytical Chemistry*, 153(8):1621–1629, 2009.

[48] Damon H. May, Kaipo Tamura, and William S. Noble. Param-Medic: A Tool for Improving MS/MS Database Search Yield by Optimizing Parameter Settings. *Journal of Proteome Research*, 16(4):acs.jproteome.7b00028, 2017.

[49] Damon H. May, Emma Timmins-Schiffman, Molly P. Mikan, H. Rodger Harvey, Elhanan Borenstein, Brook L. Nunn, and William Stafford Noble. An alignment-free 'metapeptide' strategy for metaproteomic characterization of microbiome samples using shotgun metagenomic sequencing. *Journal of Proteome Research*, page acs.jproteome.6b00239, 2016.

[50] Jennifer A. Mead, Luca Bianco, Vanessa Ottone, Chris Barton, Richard G. Kay, Kathryn S. Lilley, Nicholas J. Bond, and Conrad Bessant. MRMaid, the Web-based Tool for Designing Multiple Reaction Monitoring (MRM) Transitions. *Molecular & Cellular Proteomics*, 8(4):696–705, 2009.

[51] I. Melvin, J. Weston, C. Leslie, and W. S. Noble. Detecting remote evolutionary relationships among proteins by large-scale semantic embedding. *PLoS Computational Biology*, 7(1):e1001047, 2011.

[52] Bart Mesuere, Griet Debyser, Maarten Aerts, Bart Devreese, Peter Vandamme, and Peter Dawyndt. The Unipept metaproteomics analysis pipeline. *Proteomics*, 15(8):1437–1442, 2015.

[53] Bart Mesuere, Bart Devreese, Griet Debyser, Maarten Aerts, Peter Vandamme, and

Peter Dawyndt. Unipept: Tryptic peptide-based biodiversity analysis of metaproteome samples. *Journal of Proteome Research*, 11(12):5773–5780, 2012.

[54] Wang Mingxun, Jian Wang, Jeremy Carver, Benjamin S. Pullman, Seong Won Cha, and Nuno Bandeira. Assembling the Community-Scale Discoverable Human Proteome. *Cell Systems*, 7(4):412–421, 2018.

[55] Eli K. Moore, Brook L. Nunn, Jessica F. Faux, David R. Goodlett, and H. Rodger Harvey. Evaluation of electrophoretic protein extraction and database-driven protein identification from marine sediments. *Limnology and Oceanography: Methods*, 10:353–366, 2012.

[56] Eli K. Moore, Brook L. Nunn, David R. Goodlett, and H. Rodger Harvey. Identifying and tracking proteins through the marine water column: Insights into the inputs and preservation mechanisms of protein in sediments. *Geochimica et Cosmochimica Acta*, 83:324–359, 2012.

[57] Robert M Morris, Brook L Nunn, Christian Frazar, David R Goodlett, Ying S Ting, and Gabrielle Rocap. Comparative metaproteomics reveals ocean-scale shifts in microbial nutrient utilization and energy transduction. *The ISME Journal*, 4(5):673–685, 2010.

[58] Peter Mortensen, Joost W Gouw, Jesper V Olsen, Shao-en Ong, Kristoffer T G Rigbolt, Jakob Bunkenborg, Leonard J Foster, Albert J R Heck, Blagoy Blagoev, Jens S Andersen, and Matthias Mann. MSQuant , an Open Source Platform for Mass Spectrometry-Based Quantitative Proteomics research articles. *Journal of proteome research*, 9:393–403, 2010.

[59] Thilo Muth, Alexander Behne, Robert Heyer, Fabian Kohrs, Dirk Benndorf, Marcus Hoffmann, Miro Lehtevä, Udo Reichl, Lennart Martens, and Erdmann Rapp. The MetaProteomeAnalyzer: A Powerful Open-Source Software Suite for Metapro-

teomics Data Analysis and Interpretation. *Journal of Proteome Research*, page 150223140604002, 2015.

[60] Thilo Muth, Carolin A. Kolmeder, Jarkko Salojärvi, Salla Keskitalo, Markku Varjosalo, Froukje J. Verdam, Sander S. Rensen, Udo Reichl, Willem M. de Vos, Erdmann Rapp, and Lennart Martens. Navigating through metaproteomics data: A logbook of database searching. *Proteomics*, 15(20):3439–3453, 2015.

[61] David M Needham and Jed A Fuhrman. Pronounced daily succession of phytoplankton, archaea and bacteria following a spring bloom. *Nature Microbiology*, 1(4):1–7, 2016.

[62] 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.

[63] Alexey 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):2092–123, 2010.

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

[65] Hideki Noguchi, Jungho Park, and Toshihisa Takagi. MetaGene: Prokaryotic gene finding from environmental genome shotgun sequences. *Nucleic Acids Research*, 34(19):5623–5630, 2006.

[66] Hideki Noguchi, Takeaki Taniguchi, and Takehiko Itoh. MetaGeneAnnotator: Detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. *DNA Research*, 15(6):387–396, 2008.

[67] Brook L Nunn, Krystal Slattery, Karen A. Cameron, Emma Timmins-Schiffman, and Karen Junge. Proteomics of Colwellia psychrerythraea at subzero temperatures - a

life with limited movement, flexible membranes and vital DNA repair. *Environmental microbiology*, 17:2319–2335, 2014.

[68] Mari Ostendorf, Michael Collins, Shri Narayanan, W Douglas Oard, and Lucy Vanderwende. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Number April. 2009.

[69] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[70] Vladislav a Petyuk, Anoop M Mayampurath, Matthew E Monroe, Ashoka D Polpitiya, Samuel Purvine, Gordon a Anderson, David G Camp, and Richard D Smith. DtaRefinery, a software tool for elimination of systematic errors from parent ion mass measurements in tandem mass spectra data sets. *Molecular & cellular proteomics : MCP*, 9(3):486–96, 2010.

[71] Jarone Pinhassi, Farooq Azam, Johanna Hemphälä, Richard A. Long, Josefina Martinez, Ulla Li Zweifel, and Åke Hagström. Coupling between bacterioplankton species composition, population dynamics, and organic matter degradation. *Aquatic Microbial Ecology*, 17(1):13–26, 1999.

[72] Lilliana Radoshevich, Francis Impens, David Ribet, Juan J. Quereda, To Nam Tham, Marie Anne Nahori, Helene Bierne, Olivier Dussurget, Javier Pizarro-Cerda, Klaus Peter Knobeloch, and Pascale Cossart. ISG15 counteracts Listeria monocytogenes infection. *eLife*, 4(AUGUST2015):1–23, 2015.

[73] Michael S Rappé and Stephen J Giovannoni. The uncultured microbial majority. *Annual review of microbiology*, 57:369–94, 2003.

[74] Mina Rho, Haixu Tang, and Yuzhen Ye. FragGeneScan: Predicting genes in short and error-prone reads. *Nucleic Acids Research*, 38(20):1–12, 2010.

[75] Mak A. Saito, Alexander Dorsk, Anton F. Post, Matthew R. Mcilvin, Michael S. Rappé, Giacomo R. Ditullio, and Dawn M. Moran. Needles in the blue sea: Sub-species specificity in targeted protein biomarker analyses within the vast oceanic microbial metaproteome. *Proteomics*, 15(20):3521–3531, 2015.

[76] Anita Saraf, Serena Cervantes, Evelien M. Bunnik, Nadia Ponts, Mihaela E. Sardiu, Duk Won D. Chung, Jacques Prudhomme, Joseph M. Varberg, Zhihui Wen, Michael P. Washburn, Laurence Florens, and Karine G. Le Roch. Dynamic and combinatorial landscape of histone modifications during the intraerythrocytic developmental cycle of the malaria parasite. *Journal of Proteome Research*, 15(8):2787–2801, 2016.

[77] Matthias Schittmayer, Katarina Fritz, Laura Liesinger, Johannes Griss, and Ruth Birner-Gruenberger. Cleaning out the Litterbox of Proteomic Scientists Favorite Pet: Optimized Data Analysis Avoiding Trypsin Artifacts. *Journal of Proteome Research*, 15(4):1222–1229, 2016.

[78] S Joshua Swamidass, Chloé-agathe Azencott, Kenny Daily, and Pierre Baldi. A CROC stronger than ROC: measuring, visualizing and optimizing early retrieval. *Bioinformatics*, 26(10):1348–1356, 2010.

[79] David L. Tabb. The SEQUEST Family Tree. *Journal of the American Society for Mass Spectrometry*, 26(11):1814–1819, 2015.

[80] Alessandro Tanca, Antonio Palomba, Salvatore Pisanu, Maria Filippa Addis, and Sergio Uzzau. A human gut metaproteomic dataset from stool samples pretreated or not by differential centrifugation. *Data in Brief*, 4:559–562, 2015.

[81] Hanno Teeling, Bernhard M Fuchs, Dörte Becher, Christine Klockow, Antje Gardebrecht, Christin M Bennke, Mariette Kassabgy, Sixing Huang, Alexander J Mann, Jost

Waldmann, Marc Weber, Anna Klindworth, Andreas Otto, Jana Lange, Jörg Bernhardt, Christine Reinsch, Michael Hecker, Jörg Peplies, Frank D Bockelmann, Ulrich Callies, Gunnar Gerdts, Antje Wichels, Karen H Wiltshire, Frank Oliver Glöckner, Thomas Schweder, and Rudolf Amann. Substrate-Controlled Succession of Marine Bacterioplankton Populations Induced by a Phytoplankton Bloom. *Science*, 5567(May):608–611, 2012.

[82] Ngoc Hieu Tran, Zachariah Levine, Lei Xin, and Baozhen Shan. Protein identification with deep learning: from abc to xyz. *arXiv:1710.02765 [cs, q-bio]*, 2017.

[83] Ngoc Hieu Tran, Xianglilan Zhang, Lei Xin, Baozhen Shan, and Ming Li. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31):8247–8252, 2017.

[84] Julian Uszkoreit, Alexandra Maerkens, Yasset Perez-Riverol, Helmut E. Meyer, Katrin Marcus, Christian Stephan, Oliver Kohlbacher, and Martin Eisenacher. PIA: An Intuitive Protein Inference Engine with a Web-Based User Interface. *Journal of Proteome Research*, 14(7):2988–2997, 2015.

[85] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[86] Juan Antonio Vizcaino, Richard Côté, Florian Reisinger, Joseph M. Foster, Michael Mueller, Jonathan Rameseder, Henning Hermjakob, and Lennart Martens. A guide to the Proteomics Identifications Database proteomics data repository. *Proteomics*, 9(18):4276–4283, 2009.

[87] Ahmed F. Waly and Walid Y. Thabet. A Virtual Construction Environment for preconstruction planning. *Automation in Construction*, 12(2):139–154, 2003.

[88] Mingxun Wang, Jian Wang, Jeremy Carver, Benjamin S. Pullman, Seong Won Cha,

and Nuno Bandeira. Assembling the Community-Scale Discoverable Human Proteome. *Cell Systems*, 7(4):412–421.e5, 2018.

[89] Witold E Wolski, Malcolm Farrow, Anne-Katrin Emde, Hans Lehrach, Maciej Lalowski, and Knut Reinert. Analytical model of peptide mass cluster centres with applications. *Proteome science*, 4:18, 2006.

[90] Jody J Wright, Sangwon Lee, Elena Zaikova, David A Walsh, and Steven J Hallam. DNA extraction from 0.22 microM Sterivex filters and cesium chloride density gradient centrifugation. *Journal of Visualized Experiments: JoVE*, (31):3–6, 2009.

[91] Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. StarSpace: Embed All The Things! *arXiv:1709.03856*, 2017.

[92] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

[93] Weili Xiong, Paul E. Abraham, Zhou Li, Chongle Pan, and Robert L. Hettich. Microbial metaproteomics for characterizing the range of metabolic functions and activities of human gut microbiota. *Proteomics*, 15(20):3424–3438, 2015.

[94] Mitsuhiro Yoshida, Keitaro Yamamoto, and Satoru Suzuki. Metaproteomic characterization of dissolved organic matter in coastal seawater. *Journal of Oceanography*, 70(1):105–113, 2013.

[95] Lijun Zhong, Juntuo Zhou, Xi Chen, Yaxin Lou, Dan Liu, Xiajuan Zou, Bin Yang, Yuxin Yin, and Yan Pan. Quantitative proteomics study of the neuroprotective effects of B12 on hydrogen peroxide-induced apoptosis in SH-SY5Y cells. *Scientific Reports*, 6(February 2015):22635, 2016.

[96] Xie Xuan Zhou, Wen Feng Zeng, Hao Chi, Chunjie Luo, Chao Liu, Jianfeng Zhan, Si Min He, and Zhifei Zhang. PDeep: Predicting MS/MS Spectra of Peptides with Deep Learning. *Analytical Chemistry*, 89(23):12690–12697, 2017.