A learned embedding for efficient joint analysis of millions of mass spectra

Damon H. May[1], Jeffrey Bilmes[1,2], and William S. Noble[1,2]

[1]Department of Genome Sciences, University of Washington
[2]Department of Computer Science and Engineering, University of Washington

November 29, 2018

Abstract

Despite an explosion of data in public repositories, peptide mass spectra are usually analyzed by each laboratory in isolation, treating each experiment as if it has no relationship to any others. This approach fails to exploit the wealth of existing, previously analyzed mass spectrometry data. Others have jointly analyzed many mass spectra, often using clustering. However, mass spectra are not necessarily best summarized as clusters, and although new spectra can be added to existing clusters, clustering methods previously applied to mass spectra do not allow new clusters to be defined without completely re-clustering. As an alternative, we propose to train a deep neural network, called "GLEAMS," to learn an embedding of spectra into a low-dimensional space in which spectra generated by the same peptide are close to one another. We demonstrate empirically the utility of this learned embedding by propagating annotations from labeled to unlabeled spectra. We further use GLEAMS to detect groups of unidentified, proximal spectra representing the same peptide, and we show how to use these spectral communities to reveal misidentified spectra and to characterize frequently observed but consistently unidentified molecular species. We provide a software implementation of our approach, along with a tool to quickly embed additional spectra using a pre-trained model, to facilitate large-scale analyses.

KEYWORDS: mass spectrometry, machine learning, deep learning, repositories

1 Introduction

Since the publication of the SEQUEST[1] 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,[2] 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[3] and MassIVE[4] 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,[5] the Global Proteome Machine Database,[6] the NIST spectral libraries,[7] and MassIVE,[8] 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.[9;10] 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[11;12] 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,[13] predicting MS/MS spectra[14] and chromatographic retention time[15] for peptides, and protein inference.[16] 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[17] and images[18] to protein sequences and structures[19] to "all the things".[20] 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).

2 Results

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,"[21] in which two copies of an embedding network operate side by side (Figure 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.,[21] defined as

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

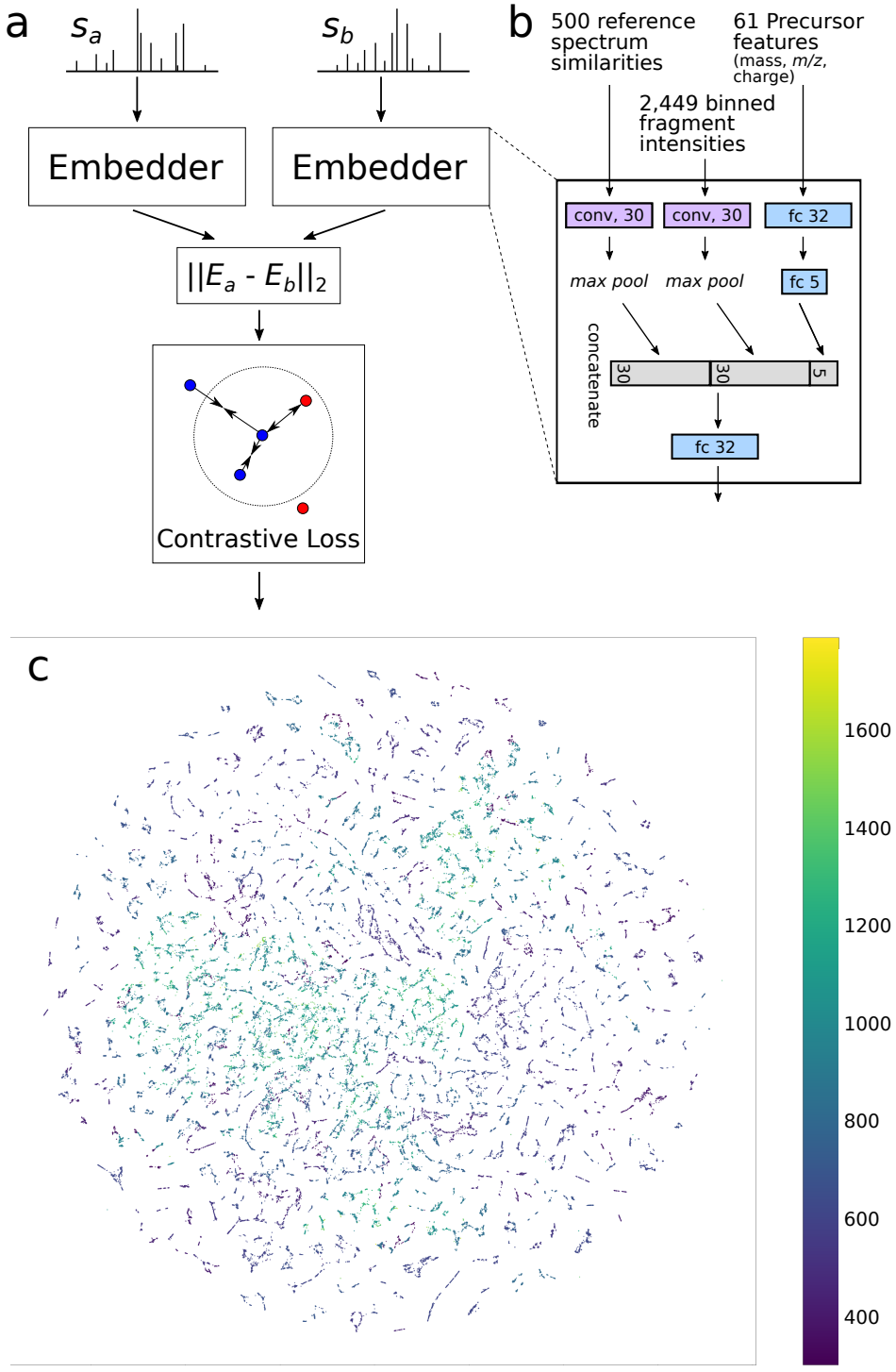

Figure 1: **Learning to embed spectra.** (A) Two spectra, s_a and s_b, are encoded and passed as input to two instances of the embedder network, with tied weights. The Euclidean distance between the two resulting embedded spectra, E_a and E_b, is passed to a contrastive loss function that penalizes far-apart same-label spectra and nearby different-label spectra, up to a margin of 1. (B) The embedder takes each of the three types of inputs separately. The precursor features are processed through a fully-connected (fc) network with two layers of sizes 32 and 5, while the binned fragment and reference spectrum similarity features are each passed through a separate convolutional (conv) neural network with 30 filters. Output of the three networks is concatenated and passed through a final fully-connected layer of size 32. (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. Each point represents a spectrum and is colored by its precursor m/z.

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.

The heart of the model itself is the embedder network (Figure 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 (Supplementary 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 (Supplementary Note 1), 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 further projection down to two dimensions. We embedded 70,000 randomly-chosen repository spectra, projected into 2D with t-SNE,[22] and colored the resulting points by precursor m/z (Figure 1C) and charge (Supplementary Figure 1A). The observed "clumpy" structure does not occur when the values for each of the 32 embedded dimensions are randomly permuted (Supplementary Figure 1B), 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 (Supplementary Figure 1C).

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 2A) 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 Supplementary 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

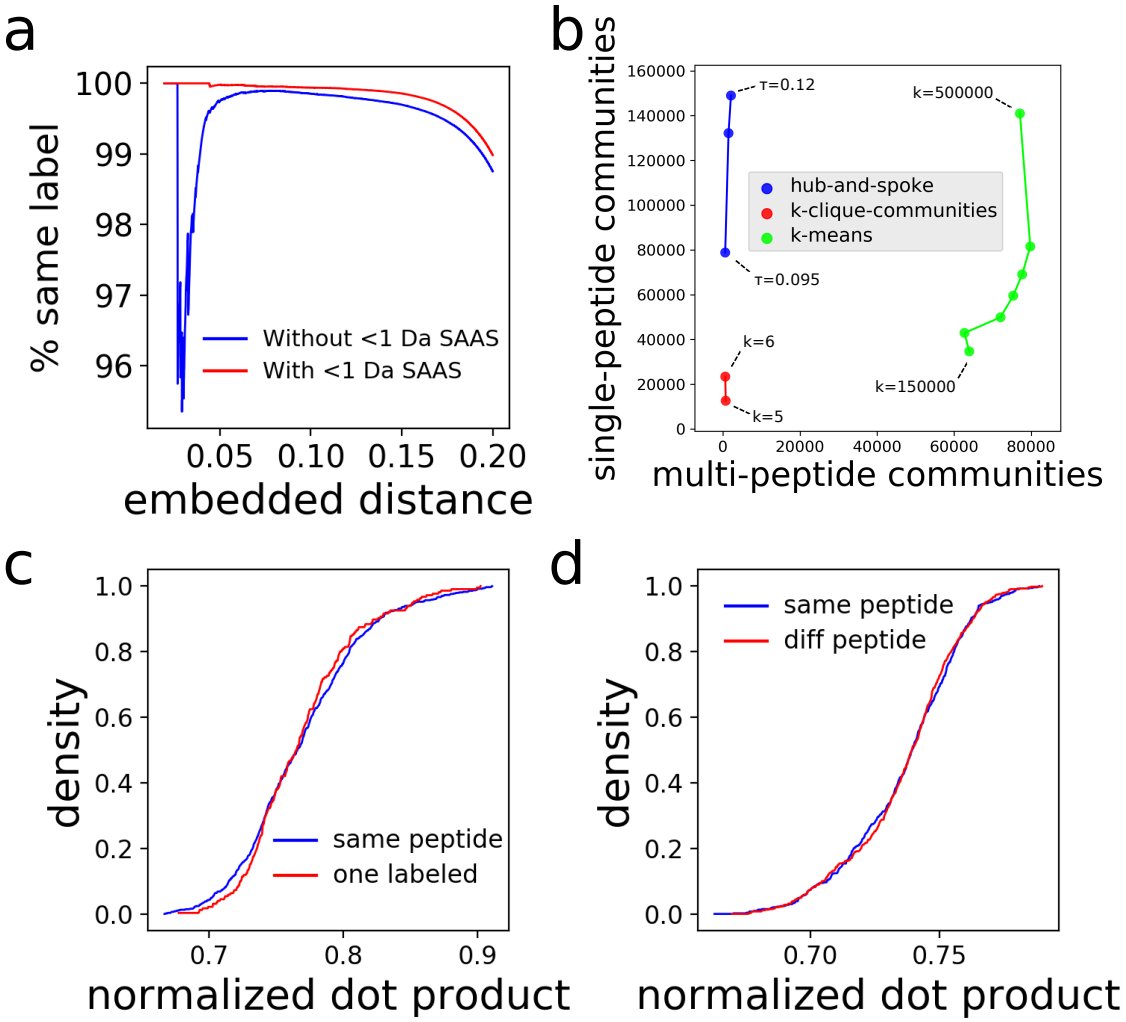

Figure 2: **Validation of the learned embedding (a)** Relationship between embedded distance and proportion of same-peptide labels. For the 1000 nearest neighbors of 200,000 randomly chosen embedded spectra, the figure plots the proportion of labeled neighbors that have the same peptide label as a function of neighbor distance threshold. 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. SAASs in nearby embedded spectra frequently indicate incorrect labels from database search. **(b)** Comparing approaches to detecting spectrum communities. 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)** Spectrum normalized dot products suggest all community spectra generated by the same peptide. 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).

based on the notion of a "k-clique community."[23] 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 (Supplementary Figure 2). 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.

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 2B, detail in Supplementary Figure 3), 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 (Supplementary Figure 4). 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 2C), 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 community 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 2D), 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.

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 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 2B). 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 Supplementary Figure 5 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.

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.[24]

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 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[25] 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.

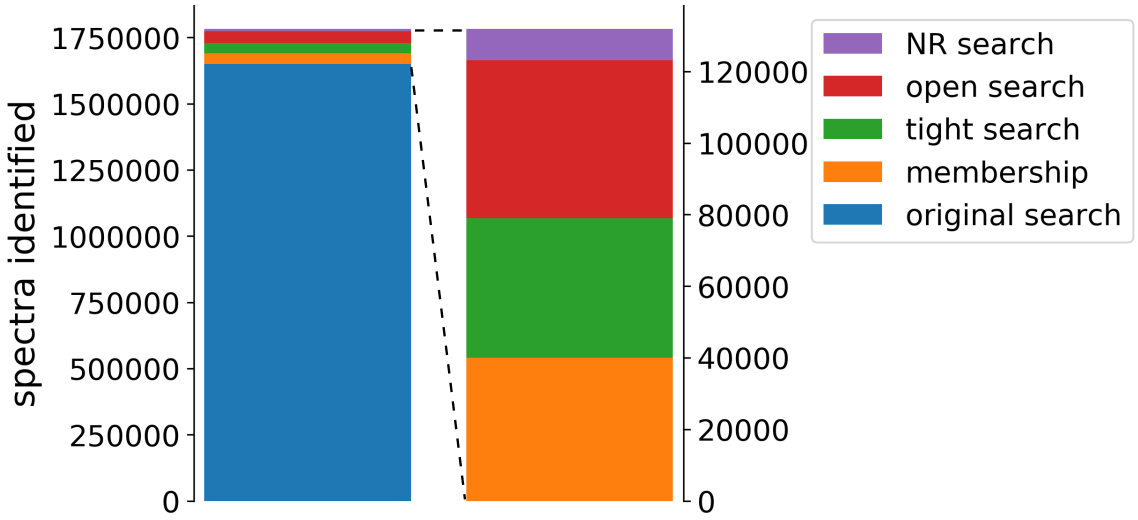

Figure 3: **Spectra newly identified using the embedding.** The original database searches of the repository spectra identified 1,650,587 spectra. The various methods of identification using the embedding identified a total of an additional 165,070 spectra, broken down by method as shown.

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 3).

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.[26;27] 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 Supplementary Figure 6A). We also searched the remaining unidentified hub spectra with the *de novo* search engine Novor[28], 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 Supplementary Figure 6B). 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.

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.[29;30] 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.

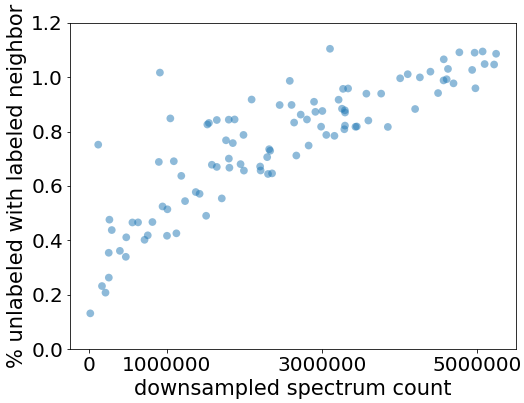

Figure 4: **A larger proportion of unidentified spectra have identified neighbors as repository size increases.** A scatter plot of the number of spectra retained (horizontal axis) versus the percentage of unidentified spectra that have an identified neighbor within τ, for 100 random downsamplings of our repository spectra.

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 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). The results show a systematic increase in this percentage, without flattening as it approaches the full size of our repository.

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,[31] including machine learning in proteomics.[32]

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.[33] 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,[34] 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 Methods

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,[35] 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.[36]

4.2 Neural network structure

The embedder network (Figure 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 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[37] and a learning rate of 0.0002, implemented with the Keras framework.[38] 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.

4.3 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 Supplementary 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[39] 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 (Supplementary Table 1). Samples were searched against the appropriate UniProt databases for single organisms, Human Microbiome Project stool database for gut microbiome,[40] or a site-specific sequencing-derived database for ocean microbiome.[41] 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[36]; 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,[42] 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[43;44] 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 Supplementary Note 1 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,[45] 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.

4.4 Detecting spectrum communities

To detect spectrum communities, we designed an algorithm (Supplementary 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,[46] 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.

4.4.1 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 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 2B 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

[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.

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 (Supplementary Figure 4 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 2B 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 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 precursor 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.

References

[1] 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. *Journal of The American Society for Mass Spectrometry*, 5:976–989, 1994.

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

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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. *J. Proteome Research*, pages 113–122, 2008.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

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

[19] 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.

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

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

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

[29] 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.

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

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

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

[39] 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.

[40] 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.

[41] 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*, 15(8):acs.jproteome.6b00239, 2016.

[42] 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.

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

[44] 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.

[45] 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.

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