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

Wout Bittremieux1, Damon H. May2, Jeffrey Bilmes3,4 and William Stafford Noble2,4

Computational methods that aim to exploit publicly available mass spectrometry repositories rely primarily on unsupervised clustering of spectra. Here we trained a deep neural network in a supervised fashion on the basis of previous assignments of peptides to spectra. The network, called ‘GLEAMS’, learns to embed spectra in a low-dimensional space in which spectra generated by the same peptide are close to one another. We applied GLEAMS for large-scale spectrum clustering, detecting groups of unidentified, proximal spectra representing the same peptide. We used these clusters to explore the dark proteome of repeatedly observed yet consistently unidentified mass spectra.

In proteomics, the dominant approach to assigning peptide sequences to tandem mass spectrometry (MS/MS) data is to treat each spectrum as an independent observation during sequence database searching. However, as public data repositories have grown to include billions of MS/MS spectra over the last decade, efforts have been undertaken to make these collections of spectra useful to researchers analyzing new datasets. For example, spectrum clustering can be used to filter for high-quality MS/MS spectra that are observed repeatedly across multiple datasets. Standard clustering is problematic, however, because it is an unsupervised approach. The input to a clustering algorithm is an unlabeled set of spectra. In practice, the labels (that is, 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 to several problems, including de novo peptide sequencing and simulating MS/MS spectra. However, to our knowledge, no one has yet applied deep neural networks to the problem of making public repository data 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. Specifically, we posit that peptide labels can be used during training of a large-scale learned model of MS/MS spectra to achieve a robust, efficient and accurate model.

Accordingly, we developed GLEAMS (GLEAMS is a Learned Embedding for Annotating Mass Spectra), which is a deep neural network that has been trained to embed MS/MS spectra into a 32-dimensional space in such a way that spectra generated by the same peptide, with the same post-translational modifications (PTMs) and charge, are close together. The learned spectrum embedding offers the advantage that new spectra can be mapped efficiently to the embedded space without requiring retraining. Our approach is fundamentally different from previous (unsupervised) spectrum clustering applications, in the sense that it uses peptide assignments generated from database search methods as labels in a supervised learning setting.

GLEAMS consists of two identical instances of an embedding neural network in a Siamese network set-up during training, the network receives pairs of spectra as input and labels indicating whether the spectra correspond to the same peptide sequence or not. Each input spectrum is encoded using three sets of features representing attributes of their precursor ion, binned fragment intensities and similarities to an invariant set of reference spectra. Each of the different feature types is processed through a separate deep neural subnetwork, after which the outputs of the three networks are concatenated and passed to a final, fully connected, layer to produce vector embeddings with dimension 32. The entire network is trained to transform the input spectra into 32-dimensional embeddings by optimizing a contrastive loss function. Intuitively, this loss function ‘pulls’ the embeddings of spectra corresponding to the same peptide together, and ‘pushes’ the embeddings of spectra corresponding to different peptides apart. The embedding network thus constitutes a function that transforms spectra into latent embeddings so that spectra corresponding to the same peptide are close to each other.

GLEAMS was trained using a set of 30 million high-quality peptide-spectrum matches (PSMs) derived from the MassIVE knowledge base (MassIVE-KB). Importantly, peptide sequence information is required only during initial supervised training of the Siamese network. Subsequent processing using an individual embedder instance is agnostic to the peptide labels and can be performed on identified and unidentified spectra in a similar fashion. After training, the embedder model was used to process 669 million spectra from 227 public human proteomics datasets included in MassIVE-KB. As an initial evaluation of the learned embeddings, these spectra were further projected down to two dimensions using uniform manifold approximation and projection (UMAP) for visual inspection. The visualizations suggest that precursor mass and precursor charge strongly influence the structure of the embedded space, and that similar spectra are indeed located close to each other. Additionally, several of the individual embedding dimensions show a correlation with the precursor mass, peptide sequence length or whether the peptides have an arginine or lysine terminus. This indicates that the GLEAMS embeddings capture latent characteristics of the spectra. Interestingly, although some of these properties were provided as input to the neural network, such as precursor mass, other properties were derived from the data without explicitly encoding them.

1Skaggs School of Pharmacy and Pharmaceutical Science, University of California San Diego, La Jolla, CA, USA. 2Department of Genome Sciences, University of Washington, Seattle, WA, USA. 3Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, USA. 4Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA. E-mail: william-noble@uw.edu
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 10 million randomly chosen embedded spectra, the relationship between neighbor distance and the proportion of labeled neighbors that have the same peptide label. The results show that neighbors at small distances overwhelmingly represent the same peptide (Fig. 1c). Furthermore, the few different peptide labels at very small distances represent almost entirely virtually indistinguishable spectra that have identical peptide labels but differ in ambiguous modification localizations. We also investigated the false negative rate, for 10 million randomly chosen embedding pairs, to understand the extent to which embeddings that correspond to the same peptide are distant in the embedded space (Extended Data Fig. 3). This analysis shows an excellent separation between same-labeled embeddings and embeddings corresponding to different peptides, with a very small false negative rate of only 1% at a distance threshold corresponding to 1% false discovery rate (FDR). Furthermore, the embeddings are robust to different types of mass spectrometry data. Phosphorylation modifications were not included in the MassIVE-KB dataset, and GLEAMS thus did not see any phosphorylated spectra during its training. Nonetheless, GLEAMS was

![Diagram](image_url)
able to embed spectra from a phosphoproteomics study with high accuracy (Extended Data Fig. 4)\textsuperscript{16}.

To further investigate the utility of the GLEAMS embedding, we performed clustering in the embedded space to find groups of similar spectra, and compared the performance with that of the spectrum clustering tools MS-Cluster\textsuperscript{13,14}, spectra-cluster\textsuperscript{4,5}, MaRaCluster\textsuperscript{15} and falcon\textsuperscript{16}. The comparison indicates that clustering in the GLEAMS embedded space is of a similar or higher quality than clusterings produced by state-of-the-art tools (Fig. 1d,e). Additionally, GLEAMS generates highly ‘complete’ clustering results (Fig. 1e). Completeness measures the extent to which multiple spectra corresponding to the same peptide are concentrated in few clusters. Compared with alternative clustering tools, GLEAMS produces larger clusters, with data drawn from more diverse studies (Extended Data Fig. 5). By minimizing the extent to which spectra generated from the same peptide are assigned to different clusters, GLEAMS achieves improved data reduction from spectrum clustering compared with alternative clustering tools. Furthermore, GLEAMS achieves excellent performance irrespective of the clustering algorithm used (Extended Data Fig. 6). This indicates that, despite their compact size, the GLEAMS embeddings are rich in information and suitable for downstream processing. We hypothesize that GLEAMS’ supervised training allows the model to focus on relevant spectrum features while ignoring confounding features—for example, peaks corresponding to a ubiquitous contaminant within a single study, boosting intrastudy spectrum similarity. This property is especially relevant when performing spectrum clustering at the repository scale, to maximally reduce the volume of heterogeneous data for efficient downstream processing.

A key outstanding question in protein mass spectrometry analysis concerns the source of spectral ‘dark matter,’ that is, spectra that are observed repeatedly across many experiments but consistently remain unidentified. Frank et al.\textsuperscript{17} have previously used MS-Cluster to identify 4 million unknown spectra included in ‘spectral archives,’ and Griss et al.\textsuperscript{5} have used spectra-cluster to obtain identifications for 9 million previously unannotated spectra in the PROteomics IDEntifications (PRIDE) repository. The original MassIVE-KB results\textsuperscript{6} include identifications for 185 million spectra out of 669 million MS/MS spectra (1% FDR), leaving a vast amount of spectral data unexplored.

To characterize the unidentified spectra, we performed GLEAMS clustering (~1% incorrectly clustered spectra) to group 511 million spectra in 60 million clusters, followed by a multistep procedure to explore the dark proteome. This procedure involved propagating peptide labels within clusters, as well as targeted open modification searching of representative spectra drawn from clusters of unidentified spectra (Methods). In total, this strategy succeeded in assigning peptides to 132 million previously unidentified PSMs, increasing the number of identified spectra by 71% (Fig. 2a). Additionally, there are 207 million clustered spectra that remained unidentified. Because these spectra are repeatedly observed and expected to be high quality, they likely correspond to true signals. Consequently, this is an important collection of spectra to investigate using newly developed computational methods to further explore the dark proteome. The open modification searching results also provided information on the presence of PTMs in the human proteome (Fig. 2b and Supplementary Table 2). Besides abundant modifications that can be introduced artificially during sample processing, such as carbamidomethylation and oxidation, biologically relevant modifications from enrichment studies, such as phosphorylation, were frequently observed. We provide all of these data as a valuable community resource to further explore the dark proteome (https://doi.org/10.25345/C52K34).

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 efficiently add an additional 71% to the identifications derived from database search. A key factor in the strong performance of GLEAMS is its unique ability to efficiently operate on hundreds of millions to billions of spectra, corresponding to the size of an entire proteomics repository (Extended Data Fig. 7). 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 characteristics that are sufficiently similar to the distribution of training spectra. This makes it possible in principle to assign new spectra to spectrum clusters nearly
instantaneously upon submission to a repository, giving researchers the immediate benefit of the combined analysis efforts of the entire proteomics community.

One caveat to the GLEAMS approach is that training the embedder relies upon the availability of peptide labels. A public repository typically contains datasets of varying quality and with varying types of analyses applied to them. Such datasets may even include invalid labels or labels not subjected to FDR control. Accordingly, we have exploited labels derived from systematic, repository-wide processing of the MassIVE database to reduce variability due to differences in analysis. This type of processing is expensive but is hidden from GLEAMS users, who will interact primarily with a pretrained embedding network.

In the future, we hypothesize that the GLEAMS embedding may have utility beyond simply transferring identifications among nearby spectra. For example, it may be that semantic relationships among spectra generated by related molecular species can be derived from the latent space. If such relationships could be mapped, then it might be possible to, for instance, predict where in the embedded space a spectrum generated by a peptide with a particular PTM would be found, based on the known location of the unmodified species. The embedding also 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. Another direction for future work is the development of statistical confidence estimation procedures suitable for this type of learned embedding. Target-decoy methods for confidence estimation are used widely but do not generalize in a straightforward fashion to a method based on propagation in the GLEAMS embedded space.

Online content
Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-022-01496-1.

Received: 29 November 2018; Accepted: 14 April 2022; Published online: 30 May 2022

References
1. Tabb, D. L. The SEQUEST family tree. J. Am. Soc. Mass Spectrom. 26, 1814–1819 (2015).
2. Pérez-Riverol, Y. et al. The PRIDE database and related tools and resources in 2019: improving support for quantification data. Nucleic Acids Res. 47, D442–D450 (2019).
3. Frank, A. M. et al. Clustering millions of tandem mass spectra. J. Proteome Res. 7, 113–122 (2008).
4. Griss, J., Foster, J. M., Hermjakob, H. & Vizcaíno, J. A. PRIDE cluster: building a consensus of proteomics data. Nat. Methods 10, 95–96 (2013).
5. Griss, J. et al. Recognizing millions of consistently unidentified spectra across hundreds of shotgun proteomics datasets. Nat. Methods 13, 651–656 (2016).
6. Wang, M. et al. Assembling the community-scale discoverable human proteome. Cell Syst. 7, 412–421.e5 (2018).
7. LeCon, V., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015).
8. Tran, N. H. et al. De novo peptide sequencing by deep learning. Proc. Natl Acad. Sci USA. 114, 8247–8252 (2017).
9. Tran, N. H. et al. Deep learning enables de novo peptide sequencing from data-independent-acquisition mass spectrometry. Nat. Methods 16, 63–66 (2018).
10. Gessulat, S. et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. Nat. Methods 16, 509–518 (2019).
11. Tiwary, S. et al. High-quality MS/MS spectrum prediction for data-dependent and data-independent acquisition data analysis. Nat. Methods 16, 519–525 (2019).
12. Hadsell, R., Chopra, S., LeCun, Y. In Proc. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (eds Fitzgibbon, A., Taylor, C. J., LeCun, Y.) 1735–1742 (IEEE, New York, 2006).
13. McInnes, L., Healy, J., Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. Preprint at ArXiv http://arxiv.org/abs/1802.03426 (2020).
14. Hijazi, M. et al. Reconstructing kinase network topologies from phosphoproteomics data reveals cancer-associated rewiring. Nat. Biotechnol. 38, 493–502 (2020).
15. The, M. & Käll, L. MaRaCluster: a fragment rarity metric for clustering fragment spectra in shotgun proteomics. J. Proteome Res. 15, 713–720 (2016).
16. Bitremieux, W., Laulens, K., Noble, W. S. & Dorrestein, P. C. Large-scale tandem mass spectrum clustering using fast nearest neighbor searching. Rapid Commun. Mass Spectrom. e9153 (2021).
17. Frank, A. M. et al. Spectral archives: extending spectral libraries to analyze both identified and unidentified spectra. Nat. Methods 8, 587–591 (2011).
18. Creasy, D. M. & Cottrell, J. S. Unimod: protein modifications for mass spectrometry. Proteomics 4, 1534–1536 (2004).

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2022
Methods

Encoding mass spectra for network input. Each spectrum is encoded as a vector of 3,010 features of three types: precursor attributes, binned fragment intensities and dot product similarities with a set of reference spectra.

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 several 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 more 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 Da to 6,000 Da, so the Gray code encoding has a resolution of $4 \times 10^{-10}$ Da. Fragment values may span the range 50.5 m/z to 2,500 m/z, so the Gray code encoding has a resolution of $2 \times 10^{-7}$ m/z. 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 m/z to 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, with bins offset by half a mass cluster separation width so that bin boundaries fall between peaks. This bin size was chosen to accommodate data acquired using various instruments and protocols, and in deference to practical constraints on the number of input features for the deep learning approach. Consequently, it is unrelated to the optimal fragment mass tolerance for database search for a given run.

Similarities of each spectrum to an invariant set of reference spectra are encoded as 500 features. Each such feature is normalized to the peak product between the given spectrum and one of an invariant set of 500 reference spectra chosen randomly from the training dataset. This can be considered as an ‘empirical kernel map’ allowing GLEAMS to represent the similarity between two spectra A and B by ‘paths’ of similarities through each one of the reference spectra R via the transitive property; that is, A is similar to B if A is similar to R and R is similar to B. In contrast to the fragment binning strategy described previously, similarities to the reference spectra are computed at native resolution. The 500 reference MS/MS spectra were selected from the training data by using submodular selection, as implemented in the apricot Python package (v.0.4.1) [Ref. 2].

The GLEAMS model was trained using the 30 million high-quality PSMs used for compilation of the MassIVE-KB spectral library. PSMs were randomly split by their MassIVE dataset identifier so that the training, validation and test sets consisted of approximately 80%, 10% and 10% of all PSMs, respectively (training set: 24,986,744 PSMs; 554,290,510 MS/MS spectra from 184 datasets; validation set: 2,762,210 PSMs; 30,380,035 MS/MS spectra from 11 datasets; test set: 2,758,019 PSMs; 84,699,214 MS/MS spectra from 24 datasets).

The Siamese neural network was trained using positive and negative spectra pairs. Positive pairs consist of two spectra with identical precursors, and negative spectra consist of two spectra that correspond to different peptides within a 10 ppm precursor mass tolerance with at most 23% overlap between their theoretical b and y fragment sets. In total, 317 million, 205 million, 43 million and 5 million positive and negative training pairs were generated for precursor charges 2 to 5, respectively; and 8.347 billion, 3.263 billion, 182 million and 5 million negative training pairs were generated for precursor charges 2 to 5, respectively.

The Siamese neural network was trained for 50 iterations using the rectified Adam optimizer14 with learning rate 0.0002. Each iteration consisted of 512 steps with batch size 256. The pair generators per precursor charge and label (positive/negative) were shuffled and rotated separately to ensure that each batch consisted of an equal number of positive and negative pairs and balanced precursor charge states. After each iteration the performance of the network was assessed using a fixed validation set consisting of up to 512,000 spectrum pairs per precursor charge.

Training and evaluation were performed on an Intel Xeon Gold 6148 processor (2.4 GHz, 40 cores) with 768 GB memory and four NVIDIA GeForce RTX 2080 Ti graphics cards.

Phosphoproteomics embedding. An independent phosphoproteomics dataset by Hijazi et al.14, generated to study kinase network topology, was used to evaluate the robustness of the GLEAMS embeddings for unseen PTMs. All raw and mzIdentML22 files were downloaded from PRIDE (project PXD015943) using pxpp34 and converted to mzML file using ThermoRawFileParser (v.1.3.4)35. As per Hijazi et al.14, the original identifications were obtained by searching with Mascot (v.2.5)36 against the SwissProt database (SwissProt_Sep2014_2015_12.fasta), with search settings of up to two tryptic missed cleavages; precursor mass tolerance 10 ppm; fragment mass tolerance 0.025 Da; cysteine carbamidomethylation as a fixed modification and N-terminal pyroglutamate formation from glutamine and deamidation of asparagine and glutamine. Mascot was configured to allow one 13C precursor mass isotope, at most one nontryptic terminus, and 10 ppm precursor mass tolerance. The search engine was also configured to use 1% false discovery rate (FDR) on the peptide level and 5% FDR on the protein level. A dynamic search space adjustment was performed during processing of the synthetic peptide spectra from the ProteomeTools project41 to account for differences in sample complexity and spectral characteristics. Next, the MassIVE-KB spectral library was generated using the top 100 PSMs for each unique precursor (that is, combination of peptide sequence and charge). The top 200 PSMs were then filtered to include only spectra with 1% FDR (of which 98.5% are phosphorylated) for 18.6 million MS/MS spectra. All spectra were embedded with the previously trained GLEAMS model, and 1.185 billion positive pairs consisting of PSMs with identical peptide sequences and file containing 185 million PSMs. Additionally, information for the 30 million filtered PSMs to create the MassIVE-KB spectral library was independently retrieved.

Neural network architecture. The embedding network (Extended Data Fig. 1) 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. The 2,449-dimension binned fragment intensities are processed through five blocks of one-dimensional convolutional layers and max pooling layers, inspired by the VGG architecture.14 The first two blocks consist of two consecutive convolutional layers, followed by a max pooling layer. The third, fourth and fifth blocks each consist of three consecutive convolutional layers, followed by a max pooling layer. The number of output filters of each of the convolutional layers is 30 for the first block, 60 for the second block, 120 for the third block and 240 for the fourth and fifth blocks. All blocks use convolutional layers with convolution window size 3 and convolution stride length 1. All max pooling layers consist of pool size 3 and stride length 2. In this fashion, the first dimension is halved after every block to ultimately convert the 2,449 × 1-dimensional input tensor to a 7 × 240-dimensional output tensor. The 500-dimensional reference spectra features are processed through a two-layer fully connected network with layer dimensions 750 and 250. The output from the network is concatenated and passed to a final, L2-regularized, fully connected layer with dimension 32.

All network layers use the scaled exponential linear units (SELU) activation function.12 The fully connected layers are initialized using LeCun normal initialization24 and the convolutional layers are initialized using the Glorot uniform initialization.24

To train the embedder, we construct a ‘Siamese network’ containing two instances of the embedder with tied weights W forming function $G_{W}(\cdot)$ (Fig. 1a). Pairs of spectra S1 and S2 are transformed to embeddings $G_{W}(S_{1})$ and $G_{W}(S_{2})$ in each instance of the Siamese network, respectively. The output of the Siamese network is the Euclidean distance between the two embeddings: $\|G_{W}(S_{1})−G_{W}(S_{2})\|$. The Siamese network is trained to optimize the following contrastive loss function:

$$L(W,Y,S_{1},S_{2}) = y \min (\|G_{W}(S_{1})−G_{W}(S_{2})\|, 1)^{2}$$

$$+ (1 − y)(\max (0, 1 − \|G_{W}(S_{1})−G_{W}(S_{2})\|))^{2}$$

where $y$ is the label associated with the pair of spectra $S_{1}$ and $S_{2}$.

Training the embedder. The GLEAMS model was trained using the 30 million high-quality PSMs used for compilation of the MassIVE-KB spectral library. PSMs were randomly split by their MassIVE dataset identifier so that the training, validation and test sets consisted of approximately 80%, 10% and 10% of all PSMs, respectively (training set: 24,986,744 PSMs; 554,290,510 MS/MS spectra from 184 datasets; validation set: 2,762,210 PSMs; 30,380,035 MS/MS spectra from 11 datasets; test set: 2,758,019 PSMs; 84,699,214 MS/MS spectra from 24 datasets).

As per Hijazi et al.14, generated to study kinase network topology, was used to evaluate the robustness of the GLEAMS embeddings for unseen PTMs. All raw and mzIdentML22 files were downloaded from PRIDE (project PXD015943) using pxp11.1.1.1 and converted to mzML files using ThermoRawFileParser (v.1.3.4)35. The search engine was configured to use 1% FDR on the peptide level and 5% FDR on the protein level. A dynamic search space adjustment was performed during processing of the synthetic peptide spectra from the ProteomeTools project41 to account for differences in sample complexity and spectral characteristics. Next, the MassIVE-KB spectral library was generated using the top 100 PSMs for each unique precursor (that is, combination of peptide sequence and charge). The top 200 PSMs were then filtered to include only spectra with 1% FDR (of which 98.5% are phosphorylated) for 18.6 million MS/MS spectra. All spectra were embedded with the previously trained GLEAMS model, and 1.185 billion positive pairs consisting of PSMs with identical peptide sequences and
Embedding clustering. Before clustering, the MS/MS spectra were converted to embeddings using the trained GLEAMS model. Next, the embeddings were split per precursor charge and partitioned into buckets based on their corresponding precursor mass so that the precursor m/z difference of consecutive embeddings in neighboring buckets exceeded the 10 ppm precursor m/z tolerance. Embeddings within each bucket were clustered separately based on their Euclidean distances. Different clustering algorithms were used, including hierarchical clustering with complete linkage, single linkage, and average linkage and DBSCAN clustering. An important advantage of these clustering algorithms is that the number of clusters is not required to be known in advance.

In some cases, jointly clustered embeddings would violate the 10 ppm precursor mass tolerance because embeddings within a cluster were connected through other embeddings with intermediate precursor mass. To avoid such false positives, the clusters were postprocessed by hierarchical clustering with complete linkage of the precursor masses of the cluster members. In this fashion, clusters were split into smaller, coherent clusters so that none of the embeddings in a single cluster had a pairwise precursor mass difference that exceeded the precursor mass tolerance.

Cluster evaluation. Five clustering algorithms—GLEAMS clustering, falcon, MaRaCluster, MS-Cluster and spectra-cluster—were run using a variety of parameter settings for each. For GLEAMS clustering, several clustering algorithms were used. For hierarchical clustering with complete linkage, Euclidean distance thresholds of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8 were used. For hierarchical clustering with complete linkage, single linkage, and average linkage and DBSCAN clustering. An important advantage of these clustering algorithms is that the number of clusters is not required to be known in advance.

In some cases, jointly clustered embeddings would violate the 10 ppm precursor mass tolerance because embeddings within a cluster were connected through other embeddings with intermediate precursor mass. To avoid such false positives, the clusters were postprocessed by hierarchical clustering with complete linkage of the precursor masses of the cluster members. In this fashion, clusters were split into smaller, coherent clusters so that none of the embeddings in a single cluster had a pairwise precursor mass difference that exceeded the precursor mass tolerance.

To assign peptide labels to previously unidentified spectra, first peptide annotations were propagated within pure clusters. For 60 million clusters that contained a mixture of unidentified spectra and PSMs with identical peptide labels, the unidentified spectra were assigned the same label, resulting in 82 million new PSMs.

Second, open modification searching was used to process the unidentified spectra. Medoid spectra were extracted from clusters consisting of only unidentified spectra by selecting the spectra with minimum embedded distances to all other cluster members. This resulted in 45 million medoid spectra representing 257 million clustered spectra. The medoid spectra were split into two groups based on cluster size—size two and size greater than two—and exported to two MGF files.

Next, the ANN-SoLo (v.0.3.3) spectral library search engine was used for open modification searching. Search settings included preprocessing the spectra by removing peaks outside the 101 m/z to 1,500 m/z range and peaks within a 1.5 ppm window around this precursor m/z, precursor mass tolerance 10 ppm for the standard searching step of ANN-SoLo’s built-in cascade search and 500 Da for the open searching step, and fragment mass tolerance 0.05 m/z. Other settings were kept at their default values. As reference spectral library, the MassIVE-KB spectral library was used. Duplicates were removed using SpectraST (v.5.0 as part of the Trans-Proteomic Pipeline version 5.1.0 (ref. 3)) by retaining only the best replicate spectrum for each individual peptide ion, and decoy spectra were added in a 1:1 ratio using the shuffle-and-reposition method. PSMs were filtered at 1% FDR by the built-in subgroup FDR procedure in ANN-SoLo.

ANN-SoLo managed to identify 5.3 million PSMs (12% of previously unidentified cluster medoid spectra). Finally, peptide labels from the ANN-SoLo PSMs were propagated to other cluster members, resulting in 44 million additional PSMs.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The data used to explore the dark proteome have been deposited to the MassIVE repository with the dataset identifier MSV000885998. It consists of MGF files containing the representative medoid spectra from GLEAMS clustering and the associated ANN-SoLo identifications in mzTab format.

All other data supporting the presented analyses have been deposited to the MassIVE repository with the dataset identifier MSV00088599. Source data are provided with this paper.

Code availability

GLEAMS was implemented in Python 3.8. Pytopes (v.4.3.2) was used to read MS/MS spectra in the mzML−, mzXML and MGF formats. spectrum util (v.0.3.4) was used for spectrum preprocessing. We performed a submodular selection using apricot (v.0.4.1)−12. The neural network code was implemented using the Tensorflow/Keras framework (v.2.2.0)−13 and fastcluster (v.1.1.24)−14 and fastcluster (v.1.1.20)−15 for hierarchical clustering. Additional scientific computing was done using NumPy (v.1.19.0)−16, SciKit-Learn (v.0.23.1)−19, Numba (v.0.50.0)−20 and Pandas (v.1.0.5)−21. Data analysis and visualization were performed using Jupyter Notebooks, matplotlib (v.3.3.0)−22, Seaborn (v.0.11.0)−23 and UMAP (v.0.6.4)−24. All code is available as open source under the permissive BSD license at https://github.com/bittremieux/GLEAMS. Code used to analyze the data and to generate the figures presented here is available on GitHub (https://github.com/bittremieux/GLEAMS_notebooks). Permanent archives of the source code and the analysis notebooks are available on Zenodo at https://doi.org/10.5281/zenodo.5794613 and https://doi.org/10.5281/zenodo.5794616, respectively−25.

References

19. Wolski, W. E. et al. Analytical model of peptide mass cluster centres with applications. Proteome Sci. 4, 18 (2006).
20. Hofmann, T., Schollkopf, B. & Smola, A. J. Kernel methods in machine learning. Ann. Stat. 36, 1171–1202 (2008).
21. Schreiber, J., Bilmes, J. & Noble, W. S. apricot: submodular selection for data reduction. Mol. Cell. Proteom. 10, R110.000133–R110.000133 (2011).
22. Kim, S. & Pezchner, P. A. MS-GF+ makes progress towards a universal database search tool for proteomics. Nat. Commun. 5, 5277 (2014).
23. Breuza, L. et al. The UniProtKB guide to the human proteome. Database 2016, ba120 (2016).
24. Zolg, D. P. et al. Building proteome tools based on a complete synthetic human proteome. Nat. Methods 14, 259–262 (2017).
25. Huettlin, E. L. et al. The BioPlex network: a systematic exploration of the human interactome. Cell 162, 425–440 (2015).
26. Griss, J. et al. The mzTab data exchange format: communicating mass-spectrometry-based proteomics and metabolomics experimental results to a wider audience. Mol. Cell. Proteom. 13, 2765–2775 (2014).
46. Bittremieux, W. spectrum_utils: a Python package for mass spectrometry data processing and visualization. *Anal. Chem.* 92, 659–661 (2020).

47. Abadi, M. et al. TensorFlow: a system for large-scale machine learning. In *Proc. 12th USENIX Conference on Operating Systems Design and Implementation* 265–283 (USENIX Association, 2016).

48. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* 17, 261–272 (2020).

49. Müllner, D. Fastcluster: fast hierarchical, agglomerative clustering routines for R and Python. *J. Stat. Softw.* 53, 1–18 (2013).

50. Harris, C. R. et al. Array programming with NumPy. *Nature* 585, 357–362 (2020).

51. Pedregosa, F. et al. Scikit-Learn: machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830 (2011).

52. Lam, S. K., Pitrou, A., Seibert, S. In *Proc. Second Workshop on the LLVM Compiler Infrastructure in HPC* (ed. Finkel, H.) 1–6 (ACP Press, 2015).

53. McKinney, W. In *Proc. 9th Python in Science Conference* (eds van der Walt, S. & Millman, J.) 51–56 (ACP Press, 2010).

54. Thomas, K., et al. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (eds Schmidt, B. & Loizides, F.) 87–90 (IOS Press, 2016).

55. Hunter, J. D. Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.* 9, 90–95 (2007).

56. Waskom, M., et al. mwaskom/seaborn: v0.11.2 (August 2021) Zenodo https://doi.org/10.5281/zenodo.592845 (2020).

57. Bittremieux, W. (2021). bittremieux/GLEAMS: v0.3 (v0.3)

58. Bittremieux, W. (2021). bittremieux/GLEAMS_notebooks: v0.3 (v0.3) Zenodo https://doi.org/10.5281/zenodo.5794616 (2021).

Acknowledgements

This work was supported by National Institutes of Health award R01 GM121818.

Author contributions

W.S.N. conceptualized the work. W.S.N. and J.B. supervised the work. W.B. and D.H.M. developed the software and carried out the analyses. W.B., D.H.M. and W.S.N. wrote the manuscript. All authors reviewed and edited the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s41592-022-01496-1.

Supplementary information

The online version contains supplementary material available at https://doi.org/10.1038/s41592-022-01496-1.

Correspondence and requests for materials should be addressed to William Stafford Noble.

Peer review information *Nature Methods* thanks Johannes Griss, Sean Hackett and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Primary Handling Editor: Allison Doerr; in collaboration with the *Nature Methods* team. Peer reviewer reports are available.

Reprints and permissions information is available at www.nature.com/reprints.
Extended Data Fig. 1 | GLEAMS embedder network. Each instance of the embedder network in the Siamese neural network separately receives each of three feature types as input. Precursor features are processed through a fully connected network with two layers of sizes 32 and 5. Binned fragment intensities are processed through five blocks of one-dimensional convolutional layers and max pooling layers. Reference spectra features are processed through a fully connected network with two layers of sizes 750 and 250. The output of the three subnetworks is concatenated and passed to a final fully connected layer of size 32.
Extended Data Fig. 2 | UMAP visualization of embeddings, colored by precursor charge. UMAP projection of 685,337 embeddings from frequently occurring peptides in 10 million randomly selected identified spectra. Note that the visualization may group peptides with similarities on some dimensions of the 32-dimensional embedding space, but which are nevertheless distinguishable based on their full embeddings.
Extended Data Fig. 3 | False negative rate between positive and negative embedding pairs. The false negative rate between positive and negative embedding pairs for 10 million randomly selected pairs from the test dataset, at distance threshold 0.5455 (gray line), corresponding to 1% false discovery rate, is 1%.
Extended Data Fig. 4 | ROC curve for GLEAMS performance on unseen phosphorylated spectra. Receiver operating characteristic (ROC) curve for GLEAMS embeddings corresponding to 7.5 million randomly selected spectrum pairs from an independent phosphoproteomics study. The ROC curve and area under the curve (AUC) show how often a same-peptide spectrum pair had a smaller distance than a different peptide spectrum pair.
Extended Data Fig. 5 | Clustering result characteristics produced by different tools. Clustering result characteristics at approximately 1% incorrectly clustered spectra over three random folds of the test dataset. (a) Complementary empirical cumulative distribution of the cluster sizes. (b) The number of datasets that spectra in the test dataset originate from per cluster (24 datasets total).
Extended Data Fig. 6 | GLEAMS performance with different clustering algorithms. Average clustering performance over three random folds of the test dataset containing 28 million MS/MS spectra each. The GLEAMS embeddings were clustered using hierarchical clustering with complete linkage, single linkage, or average linkage; or using DBSCAN. The performance of alternative spectrum clustering tools (Fig. 1d, e) is shown in gray for reference. (a) The number of clustered spectra versus the number of incorrectly clustered spectra per clustering algorithm. (b) Cluster completeness versus the number of incorrectly clustered spectra per clustering algorithm.
**Extended Data Fig. 7 |** Runtime scalability of spectrum clustering tools. Scalability of spectrum clustering tools when processing increasingly large data volumes. Three random subsets of the test dataset were combined to form input datasets consisting of 28 million, 56 million, and 84 million spectra. Evaluations of falcon and MS-Cluster on larger datasets were excluded due to excessive runtimes.
Extended Data Fig. 8 | UMAP visualization of the selected reference spectra. UMAP visualization of the selected reference spectra. The two-dimensional UMAP visualization was computed from the dot product pairwise similarity matrix between all 200,000 randomly selected spectra from the training data.
Extended Data Fig. 9 | Input features ablation test. Ablation testing during training of the GLEAMS Siamese network shows the benefit of the different input feature types. The performance is measured using the validation loss while training for 20 iterations consisting of 40,000 steps with batch size 256. The line indicates the smoothed average validation loss over five consecutive iterations, with the markers showing the individual validation losses at the end of each iteration.
Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

- The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided
- Only common tests should be described solely by name; describe more complex techniques in the Methods section.
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g. \( F \), \( t \), \( r \)) with confidence intervals, effect sizes, degrees of freedom and \( P \) value noted. Give \( P \) values as exact values whenever suitable.
- For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- Estimates of effect sizes (e.g. Cohen’s d, Pearson’s r), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection

No software was used for data collection.

Data analysis

GLEAMS was implemented in Python 3.8. Pyteomics (version 4.3.2) was used to read MS/MS spectra in the mzXML, mzXML, and MGF formats. spectrum_utils (version 0.3.4) was used for spectrum preprocessing. Submodular selection was performed using apricot (version 0.4.1). The neural network code was implemented using the Tensorflow/Keras framework (version 2.2.0), SciPy (version 1.5.0) and fastcluster (version 1.1.28) were used for hierarchical clustering. Additional scientific computing was done using NumPy (version 1.19.0), Scikit-Learn (version 0.23.1), Numba (version 0.50.1), and Pandas (version 1.0.5). Data analysis and visualization were performed using Jupyter Notebooks, matplotlib (version 3.3.0), Seaborn (version 0.11.0), and UMAP (version 0.4.6). All code is available as open source under the permissive BSD license at https://github.com/bittremieux/GLEAMS. Code used to analyze the data and to generate the figures presented here is available on GitHub (https://github.com/bittremieux/GLEAMS_notebooks). Permanent archives of the source code and the analysis notebooks are available on Zenodo at doi:10.5281/zenodo.5794613 and doi:10.5281/zenodo.5794616, respectively.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.
Data

Policy information about availability of data
All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:
- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The data used to explore the dark proteome have been deposited to the MassIVE repository with the dataset identifier MSV000088598. It consists of MGF files containing the representative medoid spectra from GLEAMS clustering and the associated ANN-SoLo identifications in mzTab format. All other data supporting the presented analyses have been deposited to the MassIVE repository with the dataset identifier MSV000088599.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

- [x] Life sciences
- [ ] Behavioural & social sciences
- [ ] Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

| Sample size | Sample size was based primarily on the availability of the Massive-KB resource, demonstrating that the method scales to very large datasets. |
|-------------|----------------------------------------------------------------------------------------------------------------------------------|
| Data exclusions | None |
| Replication | Comparison of clustering methods was replicated on three random folds of the test dataset. Method development was performed using a train-validation-test split, with performance results during training replicated on the test set for unbiased evaluation. |
| Randomization | Data was split randomly into train and test sets, as described in the manuscript. |
| Blinding | A blinded and held-out test set was used to perform model evaluation. |

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

| n/a | Involved in the study |
|-----|-----------------------|
| [x] | Antibodies |
| [x] | Eukaryotic cell lines |
| [x] | Palaeontology and archaeology |
| [x] | Animals and other organisms |
| [x] | Human research participants |
| [x] | Clinical data |
| [x] | Dual use research of concern |

Methods

| n/a | Involved in the study |
|-----|-----------------------|
| [x] | ChIP-seq |
| [x] | Flow cytometry |
| [x] | MRI-based neuroimaging |