Journal of proteome research
pubs.acs.org/jpr
Article

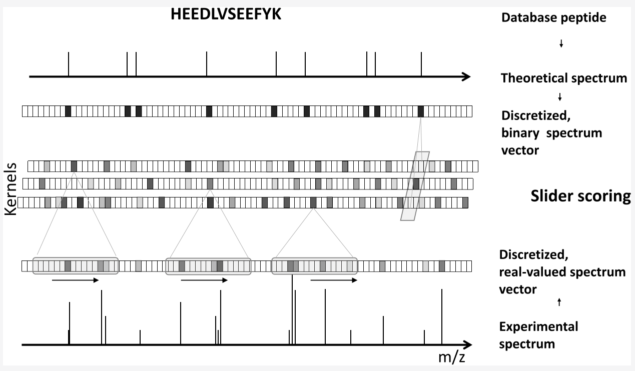
Deep Convolutional Neural Networks Help Scoring Tandem Mass Spectrometry Data in Database-Searching Approaches

Polina Kudriavtseva, Matvey Kashkinov, and Attila Kertész-Farkas*

Cite This: https://doi.org/10.1021/acs.jproteome.1c00315 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Spectrum annotation is a challenging task due to the presence of unexpected peptide fragmentation ions as well as the inaccuracy of the detectors of the spectrometers. We present a deep convolutional neural network, called Slider, which learns an optimal feature extraction in its kernels for scoring mass spectrometry (MS)/MS spectra to increase the number of spectrum annotations with high confidence. Experimental results using publicly available data sets show that Slider can annotate slightly more spectra than the state-of-the-art methods (BoltzMatch, Res-EV, Prosit), albeit 2–10 times faster. More interestingly, Slider provides only 2–4% fewer spectrum annotations with low-resolution fragmentation information than other methods with high-resolution information. This means that Slider can exploit nearly as much information from the context of low-resolution spectrum peaks as the high-resolution fragmentation information can provide for other scoring methods. Thus, Slider can be an optimal choice for practitioners using old spectrometers with low-resolution detectors.



KEYWORDS: *PSM scoring, deep learning, convolutional neural networks, fast, spectrum annotation, tandem mass spectrometry*

INTRODUCTION

Score functions in spectrum identification[1−4] are hindered by the uncertainty caused by (1) the inaccuracy of the detector in the spectrometer[5] and (2) the presence of ions from unusual peptide fragmentation.[4] As a consequence of (1), a small tolerance is introduced to handle the imprecision. In practice, this is often done by dividing the observed spectra along the m/z into small discrete bins, resulting in a vector whose components contain the sum or the maximum of the intensities of the peaks that fall in that particular bin. In modern spectrometers with detectors of high resolution, the bin width is set to 0.05 (or to 0.02), referred to as high-resolution fragmentation setting (HRFS), while for detectors of low resolution the bin width is set to 1.0005079, referred to as low-resolution fragmentation setting (LRFS). Therefore, if the heaviest observable fragment ion is 2000 m/z, then the discretization step results in a vector with 40 000 bins (also called a 40 000-dimensional vector) for HRFS with bin width of 0.05, and a vector with 1999 bins for LRFS. Score functions can take advantage of the higher degree of granularity provided by high-resolution detectors, and hence HRFS, which results in better discrimination between correct and incorrect peptide-spectrum matches (PSMs). We note that the resolution setting must be specified with respect to the spectrometer.

The score functions may confuse the secondary fragmentation ion products (SFIPs), such as the ions derived from water, carbon monoxide, or ammonia losses, with primary fragmentation ions (usually, b- or y-ions) causing additional uncertainty. To mitigate this effect, OMSSA employs an additional rule in its scoring to increase its discriminative power. It requires that at least one of the theoretical peaks must match any of the experimental peaks among the top N ($N = 3$ by default) most intensive peaks. MS-GF+[6] considers neutral losses, hydrogen transfers, and z-ions in its scoring functions. Andromeda's score function[7] defined as

$$A(s, h) = \max_{q,\text{loss}}\left\{-10 \log \sum_{j=k}^{n}\left[\binom{n}{j}\left(\frac{q}{100}\right)^{j}\left(1-\frac{q}{100}\right)^{n-j}\right]\right\} \tag{1}$$

for a discretized experimental spectrum s and a theoretical spectrum h, where n denotes the total number of theoretical peaks, k denotes the number of matching peaks, and q indicates the number of the most intense peaks in every 100 Dalton. Finally, the loss is a boolean option whether or not to consider SFIPs during such as water, ammonia, or modification-specific losses. Therefore, scoring involves an

Received: April 17, 2021

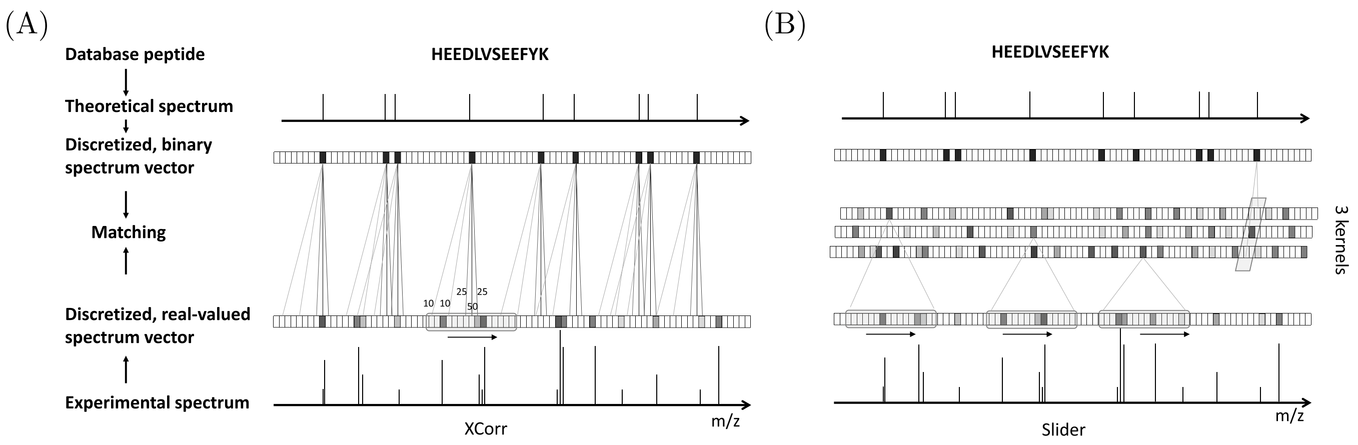


Figure 1. Graphical models of XCorr and Slider score functions. (A) XCorr weights matching ions by 50, flanking peaks by 25, and losses by 10; the weight values were specified manually. (B) Slider is a deep convolutional neural network in which the kernels act as weighted sliding windows and the weights are optimized from the MS/MS data in an end-to-end learning fashion.

optimization to find the best number q of the highest intensity peaks that are taken into account per 100 m/z intervals with or without considering SFIPs to get the highest score. The XCorr function of SEQUEST[5,8−11] additionally incorporates signals from the flanking bins of the discretized spectrum vector,[12] SFIPs, and highly charged theoretical fragmentation ion masses depending on the charge state of the precursor ion. The XCorr is formalized as

$$\mathrm{XCorr}(s, h) = E(s, h) - Z(s, h) \tag{2}$$

where E puts a weight of 50 on the matching primary fragmentation ions, a weight of 25 on the matching flanking peaks, and a weight of 10 on the matching peaks of SFIPs, and $Z(s, h)$ is a correction factor defined as $Z(s, h) = \frac{1}{151}\sum_{\tau=-75}^{+75} E(s, h[\tau])$, where, in $h[\tau]$, all vector elements of h are shifted by τ steps.[5,13] The consideration of losses can essentially be regarded as a weighted sliding window technique represented with a vector. For LRFS, the vector may consist of 61 bins, in which there is a weight of 50 in the center of the weight vector (at the 31st bin), weights of 25 besides the center for flanking bins (the 30th and 32nd bins), and weights of 10 for the SFIPs that are 17, 18, and 28 steps toward the lower end (the 14th, 13th, and 3rd bins) from the center, while all other elements in the weight vector are zeros. Such sliding windows can augment a peak by the weighted sum of the intensities of nearby peaks. Figure 1A illustrates the matching of an experimental and a theoretical spectrum by E using such a sliding window technique. Several new score functions and database searching tools have also been introduced, including Mascot,[14] HyperScore of X!Tandem,[15] Morpheus,[16] and MS Amanda;[17] however, these traditional methods are based on manually constructed score functions and have resulted in only minor improvements as compared with SEQUEST.[18] Recent studies have focused on (1) score calibration methods to provide a well-defined accurate semantics so that the spectrum annotations can be compared with each other[6,13,19−21] or (2) postprocessing methods, such as Percolator,[22] Prosit,[23] PeptideProphet,[24] ProteoTorchDNN,[25] and DeepRescore[26] to re-rank database search results to improve the number of spectrum annotations at various false discovery rates (FDRs).

Machine-learning-based scoring methods, i.e., those that operate directly on the peaks of the experimental and theoretical spectra, have emerged recently and they have resulted in an improved spectrum annotation. For instance, DRIP[27] and DeepNovo,[28] model fragmentation processes in their internal states in a temporal, sequential manner, meaning that they presume that the fragmentation ions are generated one after another in time. We also recently introduced the BoltzMatch method,[29] which works by modeling the joint probability of observing an experimental and a theoretical spectrum via a fully connected, stochastic neural network. We showed that it can enhance peaks depending on their semantic context or even reconstruct peaks of unobserved but expected fragmentation ions during its internal scoring mechanism. Although BoltzMatch annotates 30−50% more spectra than XCorr, the training of BoltzMatch is cumbersome because the BoltzMatch model is very large, containing 4 million parameters for LRFS and 1.6 billion parameters for HRFS, so training becomes time-consuming and the model becomes too prone to overfitting. For HRFS, the BoltzMatch model does not fit in the memory of common GPUs that are commercially available nowadays so the training must be done with standard CPUs.

In this article, we present a deep convolutional neural network architecture (ConvNet)[30] for scoring peptide-spectrum matches (PSMs). The main components of ConvNets are kernels that are essentially sliding windows with trainable parameters; therefore, we refer to the ConvNet architecture as Slider. The slider can be thought of as a generalized version of the XCorr scoring method with more than one sliding window in which the parameters are learned from the observed spectrum data. A graphical illustration of Slider is shown in Figure 1B.

■ SLIDER

Spectrum Preprocessing

Raw spectra were preprocessed in the same way as in the SEQUEST program but without the application of the cross-correlation penalty. Specifically, spectra were discretized along the m/z-axis with a bin width of 1.0005079 for LRFS and with a bin width of 0.05 for HRFS. The discretization step was followed by the standard normalization procedure from SEQUEST;[5] i.e., (a) peaks around the precursor ion in a window of 1.5 Da were removed, (b) peak intensities were replaced by their square root, and (c) spectra were divided into 10 equal-length regions on the mass axis, and intensities in each segment were normalized separately. Note that the intensities were scaled to a [0, 1] range in each segment. The cross-correlation penalty was not applied to the spectra. Finally, the spectra are discretized with the appropriate bin width according to LRFS or HRFS. The heaviest observable fragment ion is limited to 2000 m/z, and all peaks above are discarded. Therefore, the discretization results in a spectrum vector with 1999 bins for LRFS and 40 000 bins for HRFS.

Slider Architecture

The Slider method is based on a multilayer convolutional neural network with the following architecture.

Input Layer. The input layer takes an observed, discretized spectrum s. In terms of deep learning terminology, the input is a tensor with a shape of $[1, D]$, where D corresponds to the dimensions (bins) with $D = 1999$ for LRFS and $D = 40\,000$ for HRFS.

Batch-Normalization (BN) Layer. The input layer is followed by a BN layer, which essentially performs a data centralization and normalization step with learnable scaling parameters.[31] It is formalized as $\widehat{s_d} = \gamma_d \frac{s_d - m_d}{\sqrt{\sigma_d + \epsilon}} + b_d$, where s denotes the discretized input spectrum vector, m and σ denote the mean and the standard deviation of the data in the current minibatch, $d \in [1, ..., D]$ runs over the spectrum vector bins (i.e., the dimensions), and ϵ is a small constant that it is added for numerical stability. Finally, γ_d and b_d are learnable parameters. BN layers were introduced to mitigate the effects of the internal covariate shifts.

Convolutional Layer. The subsequent layer is a convolutional layer consisting of 20 kernels. Each kernel has a window size corresponding to ± 10.0 Da, which is a shape of $[1, 21]$ for LRFS and a shape of $[1, 401]$ for HRFS. A kernel slides over the normalized spectrum vector $\hat{s}$ and performs a weighted sum of the components in the window. The weight in the center corresponds to the current bin (i.e., peak), while the weights at $\pm i$ steps from the center correspond to the bins (i.e., peaks) $\pm i$ steps away from the center. Thus, a kernel augments a peak in the center with the weighted sum of the intensities of nearby peaks. Kernels can be regarded as different filters extracting relevant information around the small neighborhood of each peak. This layer results in a tensor with a shape of $[20, D]$, also known as a feature map.

BN Layer. The following layer performs another batch normalization. The output of this layer is a tensor with a shape of $[20, D]$.

Sigmoid Activation Layer. Each value from the BN layer is squashed into the range $(0, 1)$ using the sigmoid activation function defined as $g(x) = (1 + \exp(-x))^{-1}$. The output of this layer is a tensor with a shape of $[20, D]$

Convolutional Layer. The activations from the previous layer are aggregated with a convolutional layer consisting of only one kernel with a size of $[20, 1]$. It can be thought of as a weighted sum of the outputs from the previous kernels. This results in a tensor with a shape of $[1, D]$.

BN Layer. The next layer performs a batch normalization, resulting in a tensor with a shape of $[1, D]$.

Sigmoid Activation Layer (Output Layer). Finally, the sigmoid activation function is applied to squash the results into a range of $(0, 1)$. The shape of the output is $[1, D]$, which is the same as the shape of the input spectrum vector. The output can be regarded as a preprocessed or transformed spectrum vector s_o of the input spectrum vector s.

Slider is illustrated in Figure 1B, and the exact model indicating the trainable parameters at different layers is shown in Figure 2.

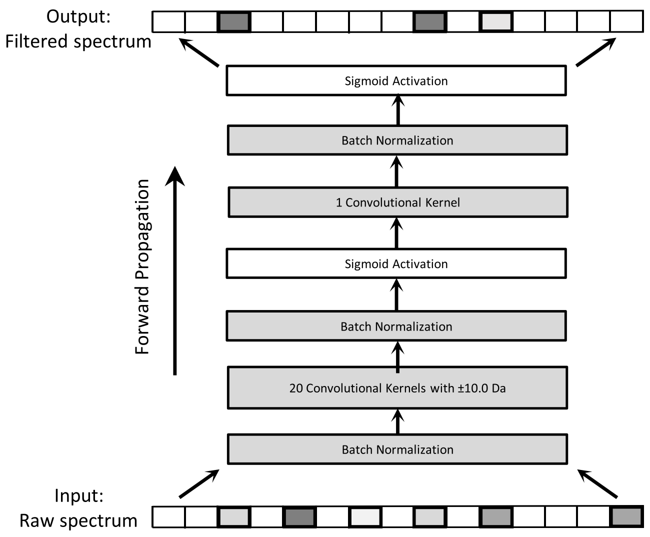


Figure 2. Architecture of Slider. Boxes with a gray background contain trainable parameters.

Training of Slider

The training of Slider is carried out in a conventional way. The training data were constructed on the fly by carrying out a standard database search and keeping the PSMs with q-values less than 0.005. The input data to Slider are the experimental spectra, while the expected output data are the corresponding theoretical spectra. Therefore, Slider is forced to transform the experimental into their theoretical spectra. The cost function to be minimized with standard backpropagation during the training was the cross-entropy between the transformed and the theoretical spectra. The optimizer was a stochastic gradient descent method using Nesterov momentum with a parameter of 0.9 and a learning rate of 0.001. The batch size was 128. We ran the training for 100 epochs but convergence was usually reached after 30−40 epochs. The training was carried out with Python using the Pytorch toolbox and it was run on a GTX Titan X GPU. The source code is freely available in the GitHub link www.github.com/kfattila/Slider, and our training of all data sets can be reproduced by executing the run_all.sh bash script.

The training of Slider is stable and it can be trained with relatively small datasets because of the following three reasons. First, the kernels of Sliders are narrow, meaning they cover only the range of ± 10.0 Da in the experimental spectra and we suspect that the patterns to be learned are simple. Second, the total number of trainable parameters of Slider are 465 for LRFS and 4105 for HRFS, respectively, which can be considered relatively small compared to deep learning architectures (having millions of parameters) employed in computer vision. Third, as Slider is a fully ConvNet, it shares weights at each spectrum bin position. This means that Slider can be considered as a fully connected neural network, which aims to predict the presence of a theoretical fragmentation ion at any position from a relatively small range of the experimental spectrum. That is, Slider uses the same parameters to predict an ion at the position, say, 467 from a range [456, 477], and at a position, say, 1723 from a range of

Table 1. Summary of Mass Spectrometry Data Sets

name	instrument	#spectra	tolerance (ppm)[a]	#proteins[b]	#peptides[c]	ACP[d]	MVM[e]	modifications[f]
Chopin	Orbitrap	12 892	20	193 634	4 952 900	479.9	0	none
HeLa	Orbitrap	10 865	20	193 634	4 952 900	496.3	0	none
HumVar	LTQ Orbitrap	15 057	50	91 464	3 420 673	1353.7	2	O[V], TMT6-plex[V]
iPRG	MALDI 5600	14 141	10	42 450	4 283 235	185.5	1	O[V]
Malaria	LTQ Orbitrap	12 594	50	11 737	2 091 849	324.5	1	O[V], TMT6-plex[S]

[a]Precursor ion tolerance. No isotope error was allowed. [b]The number of the target protein sequences in the fasta file. [c]The number of the modified target and decoy peptides obtained. The peptides were in silico generated with the lys-C for Malaria, Trypsin/P for the HeLa, and trypsin digestion rule for the rest of the data sets. Two missed cleavages were allowed for the HeLa data set and one missed cleavage was allowed for other data sets. The minimum length of the peptide was 7 amino acids and the maximum was 50. [d]Average number of candidate peptides per spectrum-charge combination (ACP). [e]Maximal variable modifications per peptide sequence. [f]Variable (V) and static (S) modifications, TMT-labeling (229.162932 Da) on lysine (K) and on N-terminal (Nt) modifications, and oxidation (O) of methionine (+15.9949 Da). Static carbamidomethylation modification of cysteine (+57.02) was used for all data sets.

[1712, 1733] for LRFS. This further implies that if there are 2000 bins in an experimental spectrum then it would provide 2000 training instances for LRFS. Thus, if the training dataset consists of 10 000 MS/MS experimental spectra, then there are 2 million training instances per 465 trainable parameters for LRFS and there are 40 million training instances per 4105 parameters for HRFS.

Albeit the training of Slider is already fast, it requires only a few minutes (it will be discussed in the Results section) and the training could be supported by transfer learning[32] as well. A single Slider model could be pretrained with different datasets obtained with different instruments in the development phase. Slider possibly would require more parameters to store information with respect to various types of instruments. Finally, the Slider model could be fine-tuned to the actual experiment with few epochs, which could be done quickly. However, we suspect that most wet laboratories have only one (or few) mass spectrometer so, in practice, Slider needs to be trained just only once at the beginning to adjust its parameters to a given instrument. We mention that the transfer learning technique cannot be applied to transition from different spectrum discretization parameters (bin width) because different bin widths would produce incompatible ConvNets (i.e., kernels with different window lengths) to cover the ±10.0 Da range.

Evaluation of Slider in Spectrum Identification

The scoring of an observed spectrum s and a theoretical spectrum h with Slider is carried out by (1) propagating s through Slider, obtaining a preprocessed spectrum vector s_o, and (2) carrying out a standard dot product between s_o and h. We performed the Slider scoring with the CRUX toolkit in the following way. Once Slider was trained, the experimental spectra were propagated forward through the Slider architecture, and the output spectra were exported in. MS2 file format. The spectra were then searched with the Tide-search program from the CRUX toolkit, but without performing any spectrum preprocessing steps—skip-preprocessing = T and without using neutral-loss peaks—use-neutral-loss-peaks = F and flanking peaks—use-flanking-peaks = F. We note that each value in the preprocessed spectrum vector s_o is in the range (0, 1), but it is never exactly 0.0. Therefore, Slider introduces small noise-like grass peaks into each bin but the effect of these grass peaks on scoring can be removed with score calibration methods, such as the exact p-value (XPV) method,[33] Tailor,[13] and the normalization factor of XCorr.[5] In our experiments, the PSM scores obtained with Slider were calibrated with the XPV method[33] for LRFS and with the heuristic Tailor method[13] for HRFS unless stated otherwise. We note that the XPV method is not compatible with HRFS.[34]

DATA SETS

Data Sets

In our experiments, we used five, publicly available MS2 data sets from previous publications, which provided us a total of 65 549 spectra. Here, we give only a brief summary of the main features of the data, the detailed information about data, sample preparation, and their availability can be found in the Supporting Information Notes S1. Table 1 presents the most important parameters used in database searching for each data set. The **Chopin** data set[35] contained 12 892 spectra with high-resolution MS1 and MS2 information, which was acquired with an Orbitrap instrument. The **HeLa** data set[36] contained 10 865 spectra with high-resolution MS1 and MS2 information from the proteome of 12 250 protein-coding human genes. The **HumVar** (Human Variation) was derived from lymphoblastoid cell lines from 95 HapMap individuals, including 53 Caucasians, 33 Yorubans, 9 eastern Asians, and one Japanese.[37] The complete data set contains high-resolution MS1 and MS2 information. In our study, we used only the Linfeng_012511_HapMap39_3.mzML file, which contained 15 057 experimental spectra. The **iPRG** was designed for a competition to detect modified peptides in a complex mixture by the Proteome Informatics Research Group (iPRG) at the Association of Biomolecular Resource Facilities (ABRF).[38] The data set contains 14 141 spectra of high-resolution MS1 and MS2 information generated by MALDI 5600 spectrometer from Yeast proteins. The **Malaria** data set is derived from a recent study of the erythrocytic cycle of the malaria parasite *Plasmodium falciparum* and obtained from ref37. In this study, we used only the v07548_UofF_malaria_TMT_10.ms2 file, which contains 12 594 spectra with high-resolution MS1 and MS2 information obtained with an Orbitrap spectrometer and the data was acquired from ref39.

We mention that all of these data sets contain high-resolution MS1 and MS2 information but all of these sets were searched with low- and high-resolution fragmentation settings, respectively.

METHODS

Database Search Engines

We used the following programs: **BoltzMatch**[29] with CRUX[39] and **Tide** from CRUX[39] with XCorr score function with (a) exact p-value (XPV) calibration for LRFS[33] and (b) Tailor

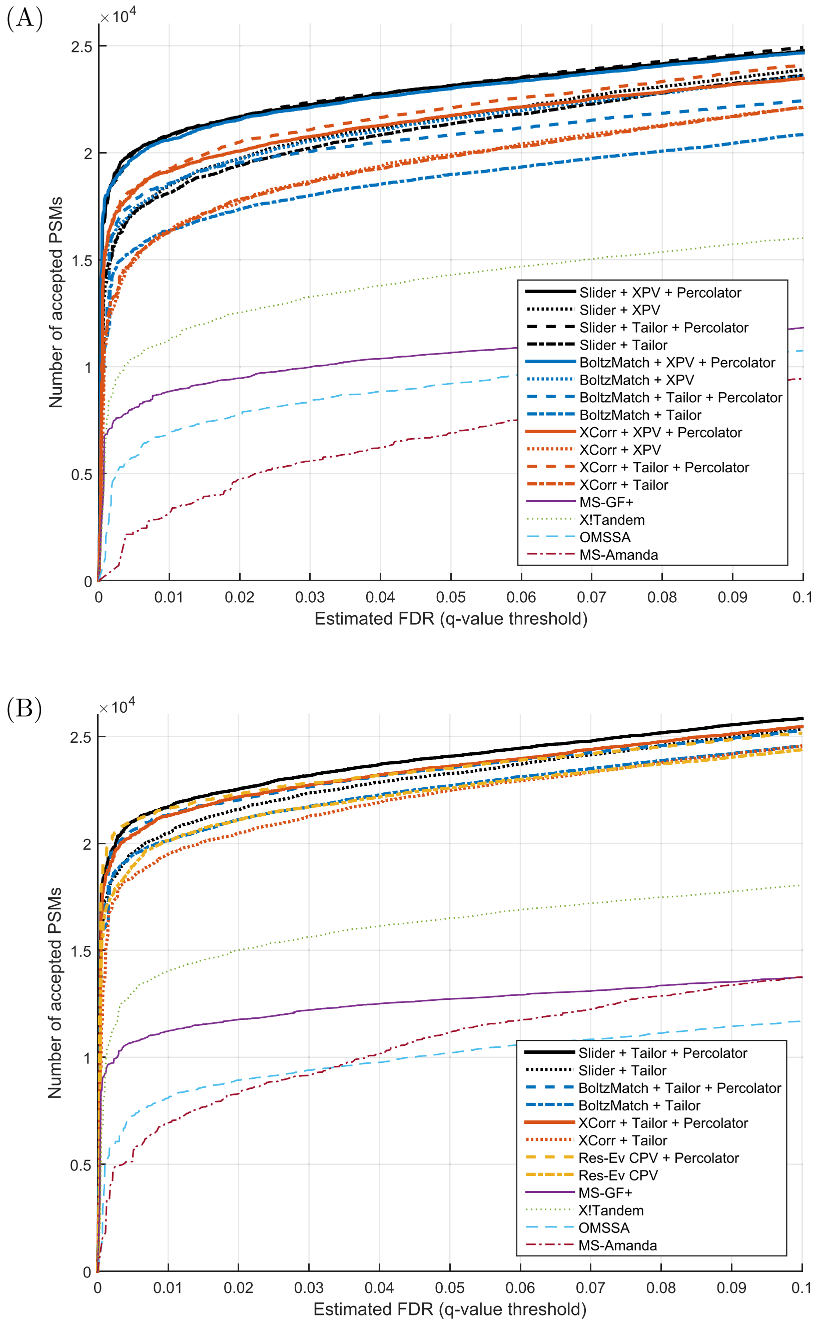


Figure 3. Cumulative spectrum annotation results with various search engines and with data sets of low-resolution (panel A) and high-resolution (panel B) fragmentation information.

calibration for HRFS;[13] **Res-Ev** with combined exact p-values (CPV) calibration,[34] **MS-GF+**,[6] **X!Tandem**,[15] **OMSSA**,[40] **MS Amanda**,[17] **Percolator**,[22] **Andromeda**[7] from MaxQuant, and **Prosit**.[23] For more details and for the parameterization of these programs, see Supporting Information Note S2. The scripts we used in our experiments to run these programs can be found in the GitHub repository of the Slider project. The fragmentation ion tolerance parameter, often called bin width, was set to 1.0005079 for LRFS and 0.05 for HRFS, respectively, for all methods and for all data sets. We note that most database-search-engines recommend a bin width of 0.02 for HRFS but with the consideration of the flanking bins in scoring. We observed that in our experiments that binning with 0.05 but neglecting flanking bins in scoring provides roughly the same results, albeit with less complication.

False Discovery Rate Calculation

False discovery rate (FDR) was estimated using a concatenated target-decoy search.[41] For every target protein sequence from the FASTA file, a decoy protein sequence was generated via a protein-reverse approach, its header was appended with the label "DECOY", and the decoy protein was concatenated to the protein FASTA file. We emphasize that the database search programs were neither allowed to generate decoy peptides nor to carry out FDR estimation. The FDR estimation was carried out by us using the following formula: $\widetilde{\mathrm{FDR}}(t) = \frac{\#\{\mathrm{Decoy} > t\} + 1}{\#\{\mathrm{Target} > t\}}$ for threshold t, where #{Target >

Table 2. Run Time in Seconds

name	Slider training[a,b]	Slider Tailor[c,d]	Slider XPV[c,d,e]	BoltzMatch training[a,f]	BoltzMatch Tailor[c,d]	BoltzMatch XPV[c,d]	XCorr Tailor[d]	XCorr XPV[d]	Res-Ev CPV[d,g]
Low-Resolution MS/MS Settings									
Chopin	35	124	1500	315	67	288	103	1560	na
HeLa	40	105	893	456	66	345	108	973	na
HumVar	34	264	6190	803	69	721	308	7080	na
iPRG	33	80	970	126	100	214	43	1180	na
Malaria	32	65	3500	79	28	780	8	629	na
total	174	638	13 053	1779	381	2348	570	11 422	na
High-Resolution MS/MS Settings									
Chopin	404	948	na	2361	290	na	200	na	4850
HeLa	185	801	na	1534	241	na	180	na	5600
HumVar	992	1190	na	2946	340	na	392	na	47 300
iPRG	281	979	na	1384	214	na	98	na	2910
Malaria	797	871	na	854	345	na	12	na	10 400
total	2659	4131	na	9077	1430	na	882	na	71 060

[a]Training time only. [b]By using GPU. [c]Crux's execution time with the transformed spectrum data sets obtained with Slider or BoltzMatch. [d]By using 1 core CPU. [e]XPV is not compatible with HRFS. [f]BoltzMatch was trained with GPU and HRFS, and with CPU and HRFS because the BoltzMatch model did not fit the memory of the GPU in the case of HRFS. [g]Res-EV is designed for HRFS only.

$t\}$ (#{Decoy > t}) denotes the number of target (decoy) peptides identified with a PSM score larger than t, as suggested by refs42, 43. The q-values[44] for each PSM were reported based on the FDR defined above, and the number of accepted PSMs were reported as a function of the q-values.

■ RESULTS AND DISCUSSION

Spectrum Annotation

First, we trained Slider on our benchmark data set with LRFS and we benchmarked it against other search engines described above. A number of annotated spectra are shown as a function of the estimated FDR with black lines for Slider, blue lines for BoltzMatch, and orange lines for XCorr with XPV (short-dotted lines) and Tailor (dash-dotted lines) calibration methods, resp., in Figure 3A. The results suggest that Slider is on par with BoltzMatch but outperforms XCorr by roughly 10% in the number of annotations. Furthermore, Slider annotates around 205, 161, 264, and 593% more spectra than MS-GF+ (purple), X!Tandem (green), OMSSA (light blue), and MS-Amanda (red), respectively, at any FDR. When the PSMs are re-ranked with the Percolator postprocessing method, Slider outperforms BoltzMatch by a small margin and XCorr by roughly 10% at any FDR level. Similar trends were observed with the application of Tailor score calibration, except that the BoltzMatch annotates around 20% fewer spectra than Slider.

When using HRFS, Slider, and BoltzMatch with the Tailor calibration method are on par, and Slider slightly outperforms the Res-EV with the combined exact p-values (CPV) method by 2.3% and XCorr by around 5%. Furthermore, Slider outperforms MS-GF+, X!Tandem, OMSSA, and MS-Amanda by a large margin, namely, by 182, 145, 253, and 295%, respectively. When search results are re-ranked with Percolator, then Slider slightly outperforms Res-EV, BoltzMatch, and XCorr. These results are shown in Figure 3B. The search results with all of the methods on different data sets with LRFS and HRFS, respectively, can be found in separate plots of Supporting Information Figure S1.

The trends under strict FDR control slightly change. When using LRFS and Percolator, the winner at 0.1% FDR is BoltzMatch, which annotates 3.5% more PSMs than Slider, while Slider annotates 11.3% more PSMs than XCorr. When using HRFS, the winner at 0.1% FDR is Res-EV, which annotates 2.4% more PSMs than Slider, but Slider annotates 4.0% more PSMs than XCorr and 10.1% more PSMs than BoltzMatch; however, all four methods seem to be on par at an FDR of less than 0.05%. These results are shown on a log scale in Supporting Information Figures S2 and S3. A number of spectra annotated at FDRs of 0.1, 1.0, and 5.0% with and without Percolator are shown in Supporting Information Tables S1 and S2, and their relative performance is shown in Supporting Information Table S3.

Our conclusions about these results are that Slider has a slightly less discriminative power than BoltzMatch under strict FDR control (around 0.1%) with low- or high-resolution fragmentation settings. This, however, can be attributed to the fact that BoltzMatch contains 17 thousand times more trainable parameters than Slider (BoltzMatch: 4 million; Slider: 465) for LRFS, and 379 thousand times more (BoltzMatch: 16 Billion; Slider: 4105) for HRFS. The Res-EV method also outperforms Slider by 2.4% PSMs at around 0.1% FDR. However, this can be attributed to the fact that Res-EV is used with combined exact p-value methods (CPV) while Slider is executed with the heuristic Tailor method so we cannot claim that the Res-EV scoring method has better discriminative power than the Slider method. Nevertheless, Slider slightly outperforms the BoltzMatch and Res-EV methods at FDRs higher than 0.5%.

Slider, similarly to other deep learning models, is deemed as a black-box model and the back-propagation during training rarely provides a human-interpretable weight explanation. Therefore, the weight investigation is complicated but we speculate that the convolutional kernels learn isotopic patterns since they weigh adjacent elements in a relatively narrow, ± 10.0 Da sliding window. Supporting Information Figure S4 shows the effects of the XCorr-penalty and the Slider using an experimental spectrum from the HumVar datasets for comparison purposes. It can be noticed that Slider diminishes the peaks (in intensity) standing alone, such as the peaks around 1400−1700 m/z marked with green, while Slider boosts peaks (in intensity), which are matched to primary fragmentation ions and are surrounded by other peaks in the near vicinity, such as peaks around 200−500 m/z in the example shown.

Timing

Perhaps the biggest advantage of Slider over BoltzMatch and Res-EV is its execution time. Slider is faster to train than BoltzMatch because it has fewer trainable parameters. Training takes 10 times longer for BoltzMatch than for Slider with LRFS, and 3.4 times longer with HRFS. Interestingly, training and evaluation of Slider take 10 times less time than running Res-EV with HRFS. The run times can be found in Table 2.

Comparison between Low- and High-Resolution Fragmentation Settings

Perhaps the most interesting observation is that Slider with LRFS can provide almost as many spectrum annotations as Slider or other methods with HRFS. Investigating the results in Supporting Information Table S4 and Figure S5 shows that Slider with LRFS provides only 4.1, 3.8, 2.5, and 2.3% fewer PSMs at 1% FDR, but is 8.3, 87.5, 12.9, and 1.08 times faster than Slider, Res-EV, BoltzMatch, and XCorr with HRFS, respectively. This allows us to conclude that explicitly incorporating information from nearby peaks into scoring methods for data with low-resolution MS2 information may provide nearly as much advantage as the higher granularity of MS2 information implicitly provides standard scoring methods. Therefore, Slider can provide nearly as many spectrum annotations for data obtained using old spectrometers with low-resolution detectors as other methods can provide for data obtained with modern instruments with high-resolution detectors.

Bias Test

Although machine-learning-based methods can acquire a preference for target peptides that could result in better performance,[45] we argue on two fronts, which is not the case for Slider.

On the one hand, the kernels of Slider can be regarded as sliding windows, and they cannot assign weights to peaks at specific m/z locations in the spectra to acquire preference toward target over decoy peptides. However, the kernels could assign weights to peak pairs that are at a certain distance away. For instance, if the amino acid arginine (R) would be more frequent among the target than the decoy peptides, then a kernel could assign higher weights to peak pairs that are at the mass of arginine away from each other. This is not the case for Slider because the kernel windows (±10.0 Da) are too small to consider peak pairs that are separated by a mass (m/z) as heavy as the mass of the amino acid.

For practical purposes, we took all the 15 057 top-scoring PSMs from the HumVar data set, which were obtained with Slider, and selected the 5000 worst-scoring PSMs (i.e., the bottom one-third from the ranked PSM list) for further ROC analysis. We note that the tail of a ranked PSM list should contain only incorrect spectrum annotations, which are equally likely to be matched with either target or decoy peptides in the case of an unbiased scoring method. Consequently, the distribution of the target PSM scores (i.e., PSMs in which the spectra are matched with target peptides) and the distribution of the decoy PSM scores should be indistinguishable, which can be tested with a ROC analysis. The area under the ROC curve (AUC) obtained with our data and shown in Supporting Information Figure S6 is 0.51051, which has an associated p-value of 0.1979 obtained with a two-sided Mann−Whitney U test. This shows that the PSM score distribution of target peptides is stochastically not greater than the one of decoy peptides at a significance level $\alpha = 0.1$. We note that the 2000 worst-scoring PSMs of the ranked PSM list result in an AUC of 0.4977 with an associated p-value of 0.8589.

Comparison with Prosit

Recently, several deep learning methods have been proposed, such as pDeep,[46] Prosit,[23] DeepMass,[47] and ProteoTorchDNN,[25] to improve the number of annotations. Prosit extracts nearly 60 features for every spectrum annotation obtained with the Andromeda/MaxQuant system,[7] including the application of the intensities of the theoretical spectra predicted by deep LSTM networks. Prosit, using the features extracted, then employs Percolator to re-score the top-scoring PSMs. At the time of writing, Prosit does not handle search results reporting modifications other than oxidation. Thus, we evaluated the performance of Slider, BoltzMatch, and Prosit using only the HeLa dataset because Malaria and HumVar datasets included TMT6-plex labeling. Slider without Percolator annotates 23.2% more PSMs than Prosit with Percolator (1757 vs 1426) and 7.7% more PSMs than BoltzMatch without Percolator (1757 vs 1631) at 0.1% FDR. Slider with Percolator yields 24.2% more PSMs than Prosit (1772 vs 1426) but 6.1% fewer PSMs than BoltzMatch with Percolator (1757 vs 1888) at 0.1% FDR. The trends change at 1% FDR: Prosit annotates 8.0 and 5.1% PSMs more than Slider and BoltzMatch, respectively, with Percolator. The results are shown in Figure 4. However, we emphasize that

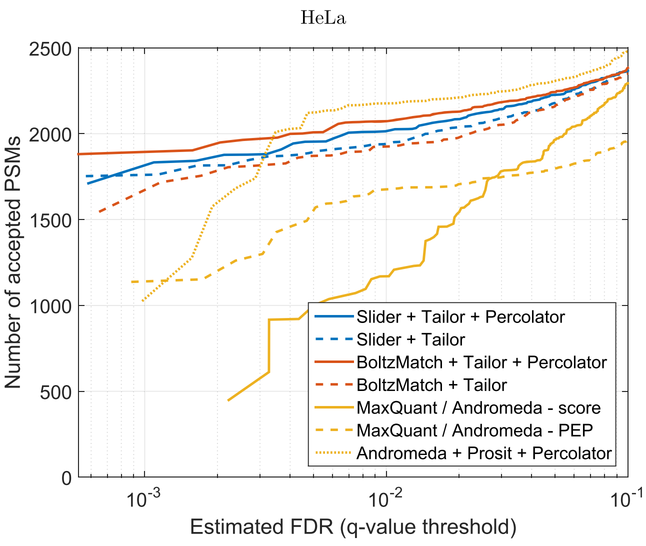


Figure 4. Search results for the HeLa data set obtained with Slider, BoltzMatch, Percolator, and Prosit. Slider outperforms Prosit at 0.1% FDR level with and without using Percolator but Prosit outperforms Slider at the 1% FDR level. Note that spectra annotated with either only CRUX or only Prosit were removed from the annotations to ensure a fair comparison of their scoring capability.

Slider is not a competitor of Prosit, ProteoTorchDNN, etc.: the top scoring PSMs found by Slider could be re-scored with Prosit or ProteoTorchDNN as well. Unfortunately, at the time of writing, Prosit does not support search results from search engines other than MaxQuant; thus, we could not evaluate the two methods in combination.

Training Stability

Deep learning methods are prone to overfitting and sensitive to hyperparameter selection. Here, we show that these are not the cases for Slider. The Supporting Information Figure S7 shows the learning curves during the training of the Slider on the HumVar dataset with standard configuration (20 convolutional kernels and ±10 Da kernel width). The data was randomly split into training and test sets in a 90−10% ratio. The results show that Slider achieves nearly the same error on the training data set as on the test data set after few, 4−6 epochs with both LRFS and HRFS.

We have performed a grid search over the convolutional kernel number and its width to find the best architecture for Slider using the HumVar data set. The kernel number parameters were set to 5, 10, 20, 30, 50; the kernel window width parameters were set to 5, 10, 20, 30, 50 Da. The results can be found in the Supporting Information Excel file. We concluded that Slider is robust to architecture selection. In the case of LRFS, the number of accepted spectrum annotations at 1% FDR varies within around STD = 1% (50 spectrum annotations). In the case of HRFS, we found Slider stable for all kernel numbers we considered; however, Slider failed to generalize for window lengths larger than ±20 Da, possibly due to a large number of trainable parameters. We found out that Slider provides good results with 20 kernels with a width of ±10 Da for LRFS and HRFS; however, changing these parameters would also result in a good performance.

CONCLUSIONS

Slider is a deep, convolutional neural network for PSM scoring to improve the number of spectrum annotations in database-searching-based systems. Slider learns an optimal feature extraction for the spectrum data without human intervention to achieve the best performance in spectrum annotation. Additionally, Slider does not require manual instrument-specific or experiment protocol-based parametrization nor does it need manual weight calibration for the matching peaks (unlike XCorr). Slider is stable and fast to train and it slightly outperforms the current state-of-the-art methods; it is 5−10 times faster with either low- or high-resolution fragmentation settings. More interestingly, Slider annotates only around 2−4% fewer spectra with low-resolution fragmentation information than with high-resolution fragmentation information, albeit around 10 times faster. This allows us to conclude that Slider can compensate for the advantage of high-resolution information in scoring by exploiting information from nearby peaks with low-resolution information. Therefore, Slider can provide nearly as many spectrum annotations using mass spectrometers with low-resolution detectors as modern instruments having high-resolution detectors.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jproteome.1c00315.

Detailed description of the experimental data used (Note S1); detailed description of the methods used (Note S2); additional tables reporting the exact numbers of spectrum annotations at 0.1, 1, and 5% FDR for the datasets and methods used in the experiments (Note S3); number of spectrum annotations obtained without Percolator at 0.1, 1, and 5% FDR (Table S1); number of spectrum annotations obtained with Percolator at 0.1, 1, and 5% FDR (Table S2); comparison of the number of annotations obtained with different methods at 0.1, 1, and 5% FDR (Table S3); relative change in the number of annotations using high- and low-resolution fragmentation settings (Table S4); additional figures showing the number of spectrum annotations at various FDRs (Note S4); spectrum annotation results at various FDRs for separate data sets (Figure S1); spectrum annotation results on log scale obtained with low-resolution fragmentation settings (Figure S2); spectrum annotation results on log scale obtained with high-resolution fragmentation settings (Figure S3); illustration of the effects of Slider on an experimental spectrum (Figure S4); spectrum annotation results obtained with low- and high- resolution fragmentation settings (Figure S5); ROC analysis for bias test (Figure S6); learning curves observed during training Slider (Figure S7) (PDF)

Search results obtained with Slider during the grid search of the kernel number and the kernel width parameters (Data S1) (XLSX)

AUTHOR INFORMATION

Corresponding Author

Attila Kertész-Farkas − *Laboratory on AI for Computational Biology, Faculty of Computer Science, HSE University, Moscow 109028, Russian Federation*; orcid.org/0000-0001-8110-7253; Phone: +7 (499) 152-07-41; Email: akerteszfarkas@hse.ru

Authors

Polina Kudriavtseva − *Laboratory on AI for Computational Biology, Faculty of Computer Science, HSE University, Moscow 109028, Russian Federation*

Matvey Kashkinov − *Faculty of Computer Science, HSE University, Moscow 109028, Russian Federation*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jproteome.1c00315

Author Contributions

M.K. developed the early version of Slider with a shallow architecture and trained it in an unsupervised manner. P.K. developed the deep Slider and trained it in a supervised fashion. A.K.-F. conceived the idea, designed the experiments, and wrote the manuscript.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of NVIDIA Corporation for the donation of the GTX Titan X GPU used for training in this research.

REFERENCES

(1) Aebersold, R.; Mann, M. Mass spectrometry-based proteomics. *Nature* **2003**, *422*, 198−207.

(2) Nesvizhskii, A. I.; Aebersold, R. Analysis, statistical validation and dissemination of large-scale proteomics datasets generated by tandem MS. *Drug Discovery Today* **2004**, *9*, 173−181.

(3) Kertész-Farkas, A.; Reiz, B.; P Myers, M.; Pongor, S. Database searching in mass spectrometry based proteomics. *Curr. Bioinf.* **2012**, *7*, 221−230.

(4) Noble, W. S.; MacCoss, M. J. Computational and statistical analysis of protein mass spectrometry data. *PLoS Comput. Biol.* **2012**, *8*, No. e1002296.

(5) Eng, J. K.; McCormack, A. L.; Yates, J. R. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.* **1994**, *5*, 976−989.

(6) Kim, S.; Pevzner, P. A. MS-GF+ makes progress towards a universal database search tool for proteomics. *Nat. Commun.* **2014**, *5*, No. 5277.

(7) Cox, J.; Neuhauser, N.; Michalski, A.; Scheltema, R. A.; Olsen, J. V.; Mann, M. Andromeda: a peptide search engine integrated into the MaxQuant environment. *J. Proteome Res.* **2011**, *10*, 1794−1805.

(8) Yates, J. R.; Eng, J. K.; McCormack, A. L.; Schieltz, D. Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. *Anal. Chem.* **1995**, *67*, 1426−1436.

(9) Park, C. Y.; Klammer, A. A.; Kall, L.; MacCoss, M. J.; Noble, W. S. Rapid and accurate peptide identification from tandem mass spectra. *J. Proteome Res.* **2008**, *7*, 3022−3027.

(10) Eng, J. K.; Fischer, B.; Grossmann, J.; MacCoss, M. J. A fast SEQUEST cross correlation algorithm. *J. Proteome Res.* **2008**, *7*, 4598−4602.

(11) Diament, B. J.; Noble, W. S. Faster SEQUEST searching for peptide identification from tandem mass spectra. *J. Proteome Res.* **2011**, *10*, 3871−3879.

(12) Eng, J. K.; Hoopmann, M. R.; Jahan, T. A.; Egertson, J. D.; Noble, W. S.; MacCoss, M. J. A deeper look into Comet-implementation and features. *J. Am. Soc. Mass Spectrom.* **2015**, *26*, 1865−1874.

(13) Sulimov, P.; Kertész-Farkas, A. Tailor: Universal, Rapid, Non-Parametric Score Calibration Method for Database Search-Based Peptide Identification in Shotgun Proteomics. *J. Proteome Res.* **2019**, *19*, 1481−1490.

(14) Perkins, D. N.; Pappin, D. J.; Creasy, D. M.; Cottrell, J. S. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* **1999**, *20*, 3551−3567.

(15) Fenyö, D.; Beavis, R. C. A method for assessing the statistical significance of mass spectrometry-based protein identifications using general scoring schemes. *Anal. Chem.* **2003**, *75*, 768−774.

(16) Wenger, C. D.; Coon, J. J. A proteomics search algorithm specifically designed for high-resolution tandem mass spectra. *J. Proteome Res.* **2013**, *12*, 1377−1386.

(17) Dorfer, V.; Pichler, P.; Stranzl, T.; Stadlmann, J.; Taus, T.; Winkler, S.; Mechtler, K. MS Amanda, a universal identification algorithm optimized for high accuracy tandem mass spectra. *J. Proteome Res.* **2014**, *13*, 3679−3684.

(18) Kim, S.; Gupta, N.; Pevzner, P. A. Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. *J. Proteome Res.* **2008**, *7*, 3354−3363.

(19) Keich, U.; Noble, W. S. On the importance of well-calibrated scores for identifying shotgun proteomics spectra. *J. Proteome Res.* **2015**, *14*, 1147−1160.

(20) Kertesz-Farkas, A.; Keich, U.; Noble, W. S. Tandem mass spectrum identification via cascaded search. *J. Proteome Res.* **2015**, *14*, 3027−3038.

(21) Keich, U.; Kertesz-Farkas, A.; Noble, W. S. Improved false discovery rate estimation procedure for shotgun proteomics. *J. Proteome Res.* **2015**, *14*, 3148−3161.

(22) Käll, L.; Canterbury, J. D.; Weston, J.; Noble, W. S.; MacCoss, M. J. Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat. Methods* **2007**, *4*, 923−925.

(23) Gessulat, S.; Schmidt, T.; Zolg, D. P.; Samaras, P.; Schnatbaum, K.; Zerweck, J.; Knaute, T.; Rechenberger, J.; Delanghe, B.; Huhmer, A.; et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nat. Methods* **2019**, *16*, 509−518.

(24) Keller, A.; Nesvizhskii, A. I.; Kolker, E.; Aebersold, R. Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal. Chem.* **2002**, *74*, 5383−5392.

(25) Halloran, J. T.; Urban, G.; Rocke, D. M.; Baldi, P. F. Deep Semi-Supervised Learning Improves Universal Peptide Identification of Shotgun Proteomics Data. *bioRxiv* **2020**, No. 380881.

(26) Li, K.; Jain, A.; Malovannaya, A.; Wen, B.; Zhang, B. DeepRescore: leveraging deep learning to improve peptide identification in immunopeptidomics. *Proteomics* **2020**, *20*, No. 1900334.

(27) Halloran, J. T.; Bilmes, J. A.; Noble, W. S. In *Learning Peptide-Spectrum Alignment Models for Tandem Mass Spectrometry*, Conference on Uncertainty in Artificial Intelligence; NIH Public Access, 2014; p 320.

(28) Tran, N. H.; Zhang, X.; Xin, L.; Shan, B.; Li, M. De novo peptide sequencing by deep learning. *Proc. Natl. Acad. Sci. U.S.A.* **2017**, *114*, 8247−8252.

(29) Sulimov, P.; Voronkova, A.; Kertész-Farkas, A. Annotation of tandem mass spectrometry data using stochastic neural networks in shotgun proteomics. *Bioinformatics* **2020**, *36*, 3781−3787.

(30) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436−444.

(31) Ioffe, S.; Szegedy, C. In *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, International Conference on Machine Learning; PMLR, 2015; 448−456.

(32) Bozinovski, S.; Fulgosi, A. In *The Influence of Pattern Similarity and Transfer Learning Upon Training of a Base Perceptron b2*, Proceedings of Symposium Informatica, 1976; pp 3−121.

(33) Howbert, J. J.; Noble, W. S. Computing exact p-values for a cross-correlation shotgun proteomics score function. *Mol. Cell. Proteomics* **2014**, *13*, 2467−2479.

(34) Lin, A.; Howbert, J. J.; Noble, W. S. Combining High-Resolution and Exact Calibration To Boost Statistical Power: A Well-Calibrated Score Function for High-Resolution MS2 Data. *J. Proteome Res.* **2018**, *17*, 3644−3656.

(35) Davis, S.; Charles, P. D.; He, L.; Mowlds, P.; Kessler, B. M.; Fischer, R. Expanding proteome coverage with CHarge Ordered Parallel Ion aNalysis (CHOPIN) combined with broad specificity proteolysis. *J. Proteome Res.* **2017**, *16*, 1288−1299.

(36) Bekker-Jensen, D. B.; Kelstrup, C. D.; Batth, T. S.; Larsen, S. C.; Haldrup, C.; Bramsen, J. B.; Sørensen, K. D.; Høyer, S.; Ørntoft, T. F.; Andersen, C. L.; et al. An optimized shotgun strategy for the rapid generation of comprehensive human proteomes. *Cell Syst.* **2017**, *4*, 587−599.

(37) Pease, B. N.; Huttlin, E. L.; Jedrychowski, M. P.; Talevich, E.; Harmon, J.; Dillman, T.; Kannan, N.; Doerig, C.; Chakrabarti, R.; Gygi, S. P.; Chakrabarti, D. Global analysis of protein expression and phosphorylation of three stages of *Plasmodium falciparum* intra-erythrocytic development. *J. Proteome Res.* **2013**, *12*, 4028−4045.

(38) Chalkley, R. J.; Bandeira, N.; Chambers, M. C.; Clauser, K. R.; Cottrell, J. S.; Deutsch, E. W.; Kapp, E. A.; Lam, H. H.; McDonald, W. H.; Neubert, T. A.; et al. Proteome informatics research group (iPRG) _2012: a study on detecting modified peptides in a complex mixture. *Mol. Cell. Proteomics* **2014**, *13*, 360−371.

(39) McIlwain, S.; Tamura, K.; Kertesz-Farkas, A.; Grant, C. E.; Diament, B.; Frewen, B.; Howbert, J. J.; Hoopmann, M. R.; Käll, L.; Eng, J. K.; et al. Crux: rapid open source protein tandem mass spectrometry analysis. *J. Proteome Res.* **2014**, *13*, 4488−4491.

(40) Geer, L. Y.; Markey, S. P.; Kowalak, J. A.; Wagner, L.; Xu, M.; Maynard, D. M.; Yang, X.; Shi, W.; Bryant, S. H. Open mass spectrometry search algorithm. *J. Proteome Res.* **2004**, *3*, 958−964.

(41) Elias, J. E.; Gygi, S. P. Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods* **2007**, *4*, 207−214.

(42) Levitsky, L. I.; Ivanov, M. V.; Lobas, A. A.; Gorshkov, M. V. Unbiased false discovery rate estimation for shotgun proteomics based on the target-decoy approach. *J. Proteome Res.* **2017**, *16*, 393−397.

(43) He, K.; Fu, Y.; Zeng, W.-F.; Luo, L.; Chi, H.; Liu, C.; Qing, L.-Y.; Sun, R.-X.; He, S.-M. A Theoretical Foundation of the Target-Decoy Search Strategy for False Discovery Rate Control in

Proteomics. 2015, arXiv:physics/1501.00537. arXiv.org e-Print archive. http://arxiv.org/abs/1501.00537.

(44) Storey, J. D.; Tibshirani, R. Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. U.S.A.* **2003**, *100*, 9440−9445.

(45) Danilova, Y.; Voronkova, A.; Sulimov, P.; Kertesz-Farkas, A. Bias in false discovery rate estimation in mass-spectrometry-based peptide identification. *J. Proteome Res.* **2019**, *18*, 2354−2358.

(46) Zhou, X.-X.; Zeng, W.-F.; Chi, H.; Luo, C.; Liu, C.; Zhan, J.; He, S.-M.; Zhang, Z. pDeep: predicting MS/MS spectra of peptides with deep learning. *Anal. Chem.* **2017**, *89*, 12690−12697.

(47) Tiwary, S.; Levy, R.; Gutenbrunner, P.; Soto, F. S.; Palaniappan, K. K.; Deming, L.; Berndl, M.; Brant, A.; Cimermancic, P.; Cox, J. High-quality MS/MS spectrum prediction for data-dependent and data-independent acquisition data analysis. *Nat. Methods* **2019**, *16*, 519−525.