Deep Semi-Supervised Learning Improves Universal Peptide Identification of Shotgun Proteomics Data

John T. Halloran*[1], Gregor Urban[2,3], David Rocke[1], and Pierre Baldi[2,3]

[1]Department of Public Health Sciences, University of California, Davis, Davis, CA, USA
[2]Department of Computer Science, University of California, Irvine, Irvine, CA, USA
[3]Institute for Genomics and Bioinformatics, University of California, Irvine, Irvine, CA, USA

Abstract

In proteomic analysis pipelines, machine learning post-processors play a critical role in improving the accuracy of shotgun proteomics analysis. Most often performed in a semi-supervised manner, such post-processors accept the peptide-spectrum matches (PSMs) and corresponding feature vectors resulting from a database search, train a machine learning classifier, and recalibrate PSM scores based on the resulting trained parameters, often leading to significantly more identified peptides across q-value thresholds. However, current state-of-the-art post-processors rely on shallow machine learning methods, such as SVMs, gradient boosted decision trees, and linear discriminant analysis. In contrast, the powerful learning capabilities of deep models have displayed superior performance to shallow models in an ever-growing number of other fields. In this work, we show that deep neural networks (DNNs) significantly improve the recalibration of shotgun proteomics data compared to the most accurate and widely used post-processors, such as Percolator and PeptideProphet. Furthermore, we show that DNNs are able to adaptively analyze complex datasets and features for more accurate universal post-processing, leading to both improved Prosit analysis and markedly better recalibration of recently developed p-value scoring functions.

Keywords: deep learning, tandem mass spectrometry, proteomics, deep neural networks, machine learning

1 Introduction

The field of proteomics has undergone explosive growth in the past two decades, largely fueled by technological breakthroughs in mass spectrometry. Most often accomplished through liquid-chromotography tandem mass spectrometry (LC-MS/MS) followed by a peptide-database search, proteomics experiments have concurrently seen rapid increases in the size and complexity of generated datasets, resulting in the proteome-scale analysis of whole biological systems[1,27,46]. In practice, however, the peptide-spectrum matches (PSMs) resulting from a database search are often uncalibrated, diminishing overall identification accuracy. To overcome this, machine learning post-processors are widely used to *recalibrate* PSMs, greatly improving the yield of identifications in proteomic analysis pipelines.

*email jthalloran@ucdavis.edu

The most accurate of these post-processors are semi-supervised[6,23], using decoy PSMs as negative training cases and examples gathered from target PSMs as positive training cases. For instance, one of the most popular post-processors, PeptideProphet,[6,25] uses linear discriminant analysis (LDA) for fixed feature sets and pre-computed weights to recalibrate input PSMs. Another extremely popular post-processor, Percolator,[23] adaptively learns feature weights by iteratively training a support vector machine (SVM) and recalibrating input PSMs using the final learned SVM parameters. Subsequent works have used shallow neural networks,[42] Naive Bayes Classifiers,[38] and gradient boosted decision trees.[22] However, these approaches limit analysis to specific search engines and sets of MS/MS features, similar to PeptideProphet. In contrast, *universal* post-processing–where arbitrary sets of MS/MS features may be analyzed accurately–has been made possible through Percolator's adaptive algorithm.

By analyzing arbitrary feature sets and adaptively learning all parameters, universal post-processing has enabled two important developments for MS/MS analysis. The first is the recent development of machine learning methods which extract large numbers of sophisticated features from MS/MS datasets[11,15,16,44] and rely on universal post-processing to use these complex feature sets for better PSM recalibration. The most notable of these feature extraction methods is Prosit,[11] which uses deep neural networks to extract 60 informative features (per PSM) for Andromeda[7] searches and feeds these features into Percolator to improve search results. The second is the quick and easy use of PSM recalibration by newly developed search algorithms. For instance, while Percolator was initially adopted by many established database-search scoring algorithms after its introduction (e.g., Mascot,[5] XCorr,[10,23,34] and X!Tandem[47]), Percolator analysis has since been rapidly adopted by more recent search algorithms near their initial development (e.g., MS-GF+,[12] XCorr p-values,[19] DRIP,[14] and combined res-ev p-values[31]).

However, while Percolator has demonstrated impressive performance for general feature sets–especially compared to other popular post-processors[45]–the use of shallow machine learning models (such as SVMs and gradient boosted decision trees) potentially leaves identifiable peptides on the table. In particular, Deep Learning[3], has lead to many recent groundbreaking advances in other fields, such as computer vision,[29,33] speech recognition,[17,18] genomics,[2,48] particle physics,[4] climate analysis,[43] and medical diagnosis.[32,35,41]

We show that deep neural networks (DNNs) improve MS/MS universal post-processing accuracy across a large number of diverse datasets, identifying more PSMs than Percolator for both Prosit analysis and the post-processing of a recently developed scoring algorithm designed for new MS/MS machines and datasets.[31] Most notably, DNNs offer the highest performance gains for the most sophisticated feature sets, demonstrating the intrinsic feature learning capabilities of deep models are more effective at exploiting rich PSM information compared to shallow models. Furthermore, compared to Percolator and the non-adaptive post-processors Scavager,[22] Q-ranker,[42] and PeptideProphet,[25] DNNs largely improve recalibrating PSM scores collected from the widely-supported Comet search algorithm,[10] only identifying fewer PSMs (at less stringent q-values, $q \geq 0.05$) than one other method on a single evaluated dataset out of twelve. The deep semi-supervised learning algorithm presented herein is available in the new universal post-processing package, *ProteoTorch*.

2 Methods

2.1 Deep learning model and semi-supervised training process

ProteoTorch uses an iterative, semi-supervised training procedure to recalibrate input targets and decoy PSMs (illustrated in Figure 1). By construction, all decoy PSMs are incorrect identifications, and are thus assigned negative labels, whereas positive training examples are estimated as

the set of target PSMs with scores achieving a stringent, user-specified q-value. These scores are then re-evaluated in each training iteration and the positive label assignments are updated, using the predictions of a classifier that is trained to distinguish between positively and negatively labeled PSMs. This overall process repeats either for a user-specified number of iterations or until convergence. Furthermore, to prevent overfitting and to improve generalizability, three-fold cross-validation (CV) is carried out in the overall procedure, where the dataset is partitioned into three separate test and train splits (as described in[13]). Thus, PSMs from a training set are always disjoint from the corresponding test set (i.e., the set of PSMs to be re-scored). When the classifier is a linear SVM,[24] the described training scheme is equivalent to Percolator.[23]

The DNN classifier used in ProteoTorch is a Multilayer Perceptron, which is the most suitable deep learning architecture for this problem as the input PSM features have no regular (or temporal) structure that could be exploited by convolutional or recurrent neural networks. To avoid confusion in the DNN training details that follow, we distinguish the use of the term *iteration* to refer to a single step in the overall semi-supervised training procedure, and the use of the term *epoch* to refer to a single pass through the data when training the DNN–thus, each iteration consists of several epochs of training within a CV fold, and in turn, each epoch consists of many gradient update steps of small batches of training data.

By default, the DNN consists of three hidden layers with 200 ReLU (rectified linear unit) neurons and is trained for 50 epochs using the Adam[28] optimization algorithm. Within an iteration, DNN training starts with a learning rate of 0.001, which is periodically reduced (ten times) by a factor of up to 50 over the course of five training epochs and reset again. At the low-point of each of these ten reduction cycles, a snapshot model[20] is taken. The resulting ten snapshot models are combined into one final ensemble model, while taking into account the validation accuracy of the individual models, and this ensemble model serves as the trained classifier for the iteration. This snapshot-based process has the distinct advantage of producing a reasonably diverse ensemble classifier without requiring additional training time. The loss function used during training is a modified cross entropy loss, which increases the loss incurred by false positive predictions by a factor of four (leading to higher penalties for incorrectly predicted training decoys). Additionally, label smoothing is used, which effectively shifts the target probability for each class slightly away from 100%. These loss function modifications were implemented to account for the asymmetry in the reliability of the labels in PSM datasets.

Prior to the first iteration, a large ensemble of snapshot models (30 by default) is used to estimate initial PSM scores, which decreases the number of iterations necessary for overall convergence. Furthermore, to regularize training with coarse initial scores, the dropout rate in the first iteration is set to 0.5. As the estimated scores improve in subsequent iterations, the benefit of dropout lowers significantly and, thus, the dropout rate is set to zero for all iterations beyond the first.

All discussed DNN hyperparameters are the default values used in Section 3.

2.2 Datasets and Processing

Database search results were collected using Crux[34] and Comet[10] for four different high-resolution MS1/MS2 datasets: a SARS-COV-2 dataset collected from COVID-19 patients,[21] downloaded from the PRoteomics IDEntifications (PRIDE) database (project PXD018682); *Saccharomyces cerevisiae* (yeast)[39] and *Plasmodium falciparum* datasets,[40] both downloaded from MassIVE MSV000084000; and a draft of the human proteome,[27] downloaded from PRIDE project PXD000561. COVID-19, yeast, and Plasmodium datasets were searched with Crux using the mzML spectra files supplied in the respective data repositories, and converted to ms2 format using msconvert[26] for Comet searches. RAW human files were converted to ms2 format using msconvert with peak-picking and

deisotoping filters. Yeast proteins were downloaded from Uniprot on April 26, 2020, resulting in 6,729 sequences, and plasmodium falciparum proteins were accessed the same day from PlasmoDB, resulting in 14,722 sequences. For the COVID-19 dataset, both human and SARS-COV-2 proteins were searched by combining the UniProt Human reference proteome and SARS2 organism proteins (https://covid-19.uniprot.org), both accessed April 30, 2020 and resulting in 20,363 sequences. The database used to search the human dataset was the same Uniprot human reference proteome used for the COVID-19 data.

All searches were fully tryptic, allowed two missed cleavages, specified a fixed modification of carbamidomethylation cysteine, and concatenated target-decoy results. The COVID-19 and human datasets were searched using a precursor tolerance of 10 ppm and fragment mass tolerance of 0.05 Da, while the yeast and Plasmodium datasets were searched using a precursor tolerance of 50 ppm and fragment mass tolerance of 0.02 Da. For searches of the COVID-19 dataset, methionine oxidation and aspartic acid deamidation were specified as variable modifications. For searches of the yeast and Plasmodium datasets, a variable peptide N-terminal pyro-glu modification was specified and, for Comet searches, a variable protein N-terminal acetylation was further specified (this variable modification was not yet supported in the utilized version of Crux). For the large-scale human dataset (consisting of 426 files and 24,931,642 total spectra), a variable methionine oxidation modification and a variable N-terminal glutamine cyclization were specified, as well as a variable protein N-terminal acetylation for Comet searches.

Crux tide-search was run with `--exact-p-value true --score-function both`, thus ranking PSMs by `NegLog10CombinePValue` (i.e., the calibrated combination of XCorr[19] and res-ev[31] p-values) during database search. Comet's `num_output_lines` parameter was set to five (matching Crux tide-search's default). Both Comet and Crux searches were set to output pepXML and PIN (Percolator INput) files. All Crux and Comet settings not discussed were left to their default values. For each dataset, resulting pepXML files were combined using InteractParser from the Trans-Proteomic Pipeline[8] (resulting PIN files were similarly concatenated using Linux commandline utilities). Per dataset, the combined pepXML files were subsequently converted to tsv format (for Q-ranker processing) using Crux psm-convert.

Prosit PIN files were downloaded from PRIDE project PXD010871. As described in,[11] these four PIN files are the result of Prosit post-processing Andromeda searches over the same human gut MS data[37] with four increasingly complex protein databases: (1) Human (20,260 proteins), (2) Human+Bacteria (276,628 proteins), (3) All organisms (469,313 proteins), and (4) IGC+All, which included the human gut microbial integrated gene catalog[30] (IGC, 10,330,558 proteins).

The described benchmark files are summarized in Table 1. All database search and post-processing results are available at http://jthalloran.ucdavis.edu/proteoTorchData.html.

2.3 Software details

ProteoTorch is implemented in Python, uses PyTorch[36] as its deep learning backend, and is available for download at https://github.com/proteoTorch/proteoTorch. Crux version 3.2-2372dae was used for combined res-ev p-value searches using tide-search, Q-ranker post-processing using q-ranker, and conversion of pepXML files to tsv format using psm-convert. Additionally, a patch was necessary to prevent Q-ranker from crashing while writing recalibrated scores, available at https://github.com/johnhalloran321/crux-toolkit. Comet version 2019.01 rev. 5 was used for all Comet searches. Percolator post-processing was performed using version 3.05. PeptideProphet post-processing was run using the Trans-Proteomic Pipeline[8] version v5.2.0 Flammagenitus. Scavager post-processing was performed using version 0.2.1a3.

Unless specified otherwise, all post-processing parameters were left to their defaults. Percolator

and ProteoTorch were run using target-decoy competition[9] with flags `-Y` and `--tdc true`, respectively. Additionally, Percolator was run with flags `-m $targetfile -M $decoyfile -U`, thus reporting both target and decoy results and skipping taking the maximum-scoring PSM per peptide. PeptideProphet was run in the semi-supervised mode and reporting of all recalibrated target/decoy PSM scores, i.e., with flags `ZERO DECOY=DECOY_ DECOYPROBS`. To ensure methods utilized the same set of input features for as fair of a comparison of recalibration performance as possible, retention time prediction was not used in post-processors supporting this option (Percolator and Scavager).

2.4 Hardware details

All described database searches and post-processing results were run on the same machine with an Intel Xeon E5-2620, 64GB RAM, and an NVIDIA Tesla K40 GPU.

3 Results

Post-processing performance for all datasets in Table 1 are plotted in Figure 2, and the number of significant PSMs identified at q-value threshold 1% for each method per dataset is listed in Table 2. Prosit and Crux `NegLog10CombinePValue` (i.e., the calibrated combination of res-ev and XCorr p-values) datasets were analyzed using universal post-processors ProteoTorch and Percolator. Comet searches were supported, and recalibrated, using all discussed post-processors (i.e., ProteoTorch, Percolator, PeptideProphet, Q-ranker, and Scavager). For Comet and Crux datasets, search function accuracy is plotted for Comet XCorr and Crux `NegLog10CombinePValue`, respectively. For all evaluated methods, q-values were calculated using target-decoy competition.[9]

At a q-value threshold of 1%, deep semi-supervised learning identifies more significant PSMs than all considered methods. Across all other evaluated q-values, deep semi-supervised learning achieves more significant identifications than Percolator for Prosit analysis, recalibration of `NegLog10CombinePValue` search results, and recalibration of Comet search results. For the Prosit dataset generated using the largest protein database (IGC+All, produced searching 10,330,558 protein sequences), the use of DNNs in ProteoTorch increases the number of significant PSMs over Percolator by 5.8% at a q-value of 1%. For the complex search function `NegLog10CombinePValue`–developed to accurately compute p-values for high-resolution MS/MS data–DNNs increase the number of significant identifications by 8.2% on average when using a strict q-value threshold of 1%, whereas Percolator results in an average increase of only 4.1%. For the non-adaptive methods, which support post-processing Comet search results, DNNs outperform all methods on all datasets except for one case (Scavager on the Comet Yeast dataset) for q-value thresholds greater than 5%.

4 Discussion

This was the first study to explore the impact of recent deep learning advances for semi-supervised post-processing of shotgun proteomics results. Across all datasets, the use of DNNs allows ProteoTorch to outperform the other universal post-processor, Percolator. This increase in performance is markedly pronounced for complex datasets; for two of the Crux datasets generated using the already calibrated (and highly accurate) `NegLog10CombinePValue` search function, the shallow networks used by Percolator have difficulty improving discrimination between calibrated target and decoy PSM scores (even failing to improve performance for the Crux Human dataset). In contrast, DNNs are capable of using these calibrated features to better distinguish target and decoy PSMs, resulting in double the recalibration improvement compared to using SVMs in Percolator at a

strict q-value threshold of 1%. Furthermore, for the complex Prosit dataset, which is generated by searching a massive protein database of over ten-million sequences (i.e., Prosit IGC+All), deep models are able to better differentiate targets from decoys, identifying over five-percent more PSMs than Percolator at a q-value threshold of 1%. For Comet searches, post-processing performance using DNNs largely outperforms all other methods; over all twelve benchmark datasets, DNNs in ProteoTorch only identify fewer PSMs on one dataset than one other method for higher considered q-values.

For the non-adaptive method PeptideProphet, it is also worth noting the effect of using precomputed LDA weights across different datasets. While PeptideProphet improves XCorr accuracy across all q-values for the datasets Comet Plasmodium, Comet Yeast, and Comet Human, no post-processed matches are deemed significant at stringent (near zero) q-values for Comet COVID-19, which leads to worse performance than simple XCorr (i.e. without post-processing) for this low q-value range. This demonstrates that the use of precomputed weights prevents effective generalization across the different datasets that were evaluated, and even degrades Comet search accuracy over one of the datasets at very strict q-values.

5 Conclusions

In this work, the impact of using deep learning models to further improve the recalibration of MS/MS database-search results was explored and compared to the shallow models, which are used in the most accurate and widely-used MS/MS post-processors. It was shown that deep semi-supervised learning is adaptable to diverse MS/MS data and feature sets, thus enabling effective universal post-processing and outperforming the state-of-the-art universal post-processor Percolator across a variety of downstream tasks: for Prosit analysis, recalibration analysis of a high-resolution p-value search function, and for recalibration of Comet search results. Furthermore, DNNs significantly improve the recalibration of Comet search results compared to the other evaluated, non-adaptive, post-processors PeptideProphet, Q-ranker, and Scavager. Thus, it was demonstrated that deep learning may be used to significantly improve the universal post-processing of shotgun proteomics data beyond the capabilities of existing approaches that rely on shallow machine learning models.

Acknowledgments and Funding

The work of JH and DR is in part supported by the National Center for Advancing Translational Sciences (NCATS), National Institutes of Health, through grant UL1 TR001860 and a GPU donation from the NVIDIA Corporation. The work of GU and PB is in part supported by grants NSF NRT 1633631 and NIH GM123558 to PB.

Conflicts of Interest

The authors declare no conflicts of interest. The sponsors had no role in the design, execution, interpretation, or writing of the study.

References

[1] R. Aebersold and M. Mann. Mass-spectrometric exploration of proteome structure and function. *Nature*, 537(7620):347–355, 2016.

[2] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

[3] P. Baldi. *Deep Learning in Science: Theory, Algorithms, and Applications*. Cambridge University Press, Cambridge, UK, 2021. In press.

[4] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):1–9, 2014.

[5] M. Brosch, L. Yu, T. Hubbard, and J. Choudhary. Accurate and sensitive peptide identification with Mascot Percolator. *Journal of Proteome Research*, 8(6):3176–3181, 2009.

[6] H. Choi and A. I. Nesvizhskii. Semisupervised model-based validation of peptide identifications in mass spectrometry-based proteomics. *Journal of Proteome Research*, 7(1):254–265, 2008.

[7] Jurgen Cox, Nadin Neuhauser, Annette Michalski, Richard A Scheltema, Jesper V Olsen, and Matthias Mann. Andromeda: a peptide search engine integrated into the maxquant environment. *Journal of proteome research*, 10(4):1794–1805, 2011.

[8] Eric W Deutsch, Luis Mendoza, David Shteynberg, Terry Farrah, Henry Lam, Natalie Tasman, Zhi Sun, Erik Nilsson, Brian Pratt, Bryan Prazen, Jimmy K Eng, Daniel B Martin, Alexey I Nesvizhskii, and Ruedi Aebersold. A guided tour of the trans-proteomic pipeline. *Proteomics*, 10(6):1150–1159, 2010.

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

[10] J. K. Eng, M. R. Hoopmann, T. A. Jahan, J. D. Egertson, W. S. Noble, and M. J. MacCoss. A deeper look into Comet — implementation and features. *Journal of the American Society for Mass Spectrometry*, 2015. In press.

[11] S. Gessulat, T. Schmidt, D. P. Zolg, P. Samaras, K. Schnatbaum, J. Zerweck, T. Knaute, J. Rechenberger, B. Delanghed, A. Huhmer, U. Reimer, H. Ehrlich, S. Aiche, B. Kuster, and M. Wilhelm. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature methods*, 16(6):509, 2019.

[12] V. Granholm, S. Kim, J. C. F. Navarro, E. Sjölund, , R. D. Smith, and L. Käll. Fast and accurate database searches with MS-GF+Percolator. *Journal of Proteome Research*, 13:890–897, 2014.

[13] Viktor Granholm, William Stafford Noble, and Lukas Käll. A cross-validation scheme for machine learning algorithms in shotgun proteomics. *BMC bioinformatics*, 13(S16):S3, 2012.

[14] John T Halloran, Jeff A Bilmes, and William S Noble. Dynamic bayesian network for accurate detection of peptides from tandem mass spectra. *Journal of proteome research*, 15(8):2749–2759, 2016.

[15] John T Halloran and David M Rocke. Gradients of generative models for improved discriminative analysis of tandem mass spectra. In *Advances in Neural Information Processing Systems*, pages 5728–5737, 2017.

[16] John T Halloran and David M Rocke. Learning concave conditional likelihood models for improved analysis of tandem mass spectra. In *Advances in Neural Information Processing Systems*, pages 5420–5430, 2018.

[17] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[18] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[19] J. J. Howbert and W. S. Noble. Computing exact p-values for a cross-correlation shotgun proteomics score function. *Molecular and Cellular Proteomics*, 13(9):2467–2479, 2014.

[20] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017.

[21] Christian Ihling, Dirk Tanzler, Sven Hagemann, Astrid Kehlen, Stefan Huttelmaier, Christian Arlt, and Andrea Sinz. Mass spectrometric identification of sars-cov-2 proteins from gargle solution samples of covid-19 patients. *Journal of proteome research*, 2020.

[22] Mark V Ivanov, Lev I Levitsky, Julia A Bubis, and Mikhail V Gorshkov. Scavager: a versatile postsearch validation algorithm for shotgun proteomics based on gradient boosting. *Proteomics*, 19(3):1800280, 2019.

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

[24] S. S. Keerthi and D. DeCoste. A modified finite newton method for fast solution of large scale linear svms. *Journal of Machine Learning Research*, 6(Mar):341–361, 2005.

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

[26] Darren Kessner, Matt Chambers, Robert Burke, David Agus, and Parag Mallick. Proteowizard: open source software for rapid proteomics tools development. *Bioinformatics*, 24(21):2534–2536, 2008.

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

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[30] Junhua Li, Huijue Jia, Xianghang Cai, Huanzi Zhong, Qiang Feng, Shinichi Sunagawa, Manimozhiyan Arumugam, Jens Roat Kultima, Edi Prifti, Trine Nielsen, et al. An integrated catalog of reference genes in the human gut microbiome. *Nature biotechnology*, 32(8):834–841, 2014.

[31] A. Lin, J. Howbert, and W. Noble. Combining high-resolution and exact calibration to boost statistical power: A well-calibrated score function for high-resolution ms2 data. *Journal of proteome research*, 17(11):3644–3656, 2018.

[32] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

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

[35] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6(1):1–10, 2016.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[37] Julia Rechenberger, Patroklos Samaras, Anna Jarzab, Juergen Behr, Martin Frejno, Ana Djukovic, Jaime Sanz, Eva M González-Barberá, Miguel Salavert, Jose Luis López-Hontangas, et al. Challenges in clinical metaproteomics highlighted by the analysis of acute leukemia patients with gut colonization by multidrug-resistant enterobacteriaceae. *Proteomes*, 7(1):2, 2019.

[38] BC Searle, CJ Emmons, and B Head. Improving peptide and protein identification rates using a novel semi-supervised approach in scaffold (abstract (3141)). In *ASMS 61st Conference on Mass Spectrometry*, pages 9–13, 2013.

[39] Brian C Searle, Lindsay K Pino, Jarrett D Egertson, Ying S Ting, Robert T Lawrence, Brendan X MacLean, Judit Villen, and Michael J MacCoss. Chromatogram libraries improve peptide detection and quantification by data independent acquisition mass spectrometry. *Nature communications*, 9(1):1–12, 2018.

[40] Brian C Searle, Kristian E Swearingen, Christopher A Barnes, Tobias Schmidt, Siegfried Gessulat, Bernhard Küster, and Mathias Wilhelm. Generating high quality libraries for dia ms with empirically corrected peptide predictions. *Nature communications*, 11(1):1–10, 2020.

[41] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[42] Marina Spivak, Jason Weston, Léon Bottou, Lukas Käll, and William Stafford Noble. Improvements to the percolator algorithm for peptide identification from shotgun proteomics data sets. *Journal of proteome research*, 8(7):3737–3745, 2009.

[43] Karen Stengel, Andrew Glaws, Dylan Hettinger, and Ryan N King. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117(29):16805–16815, 2020.

[44] Pavel Sulimov, Anastasia Voronkova, and Attila Kertész-Farkas. Annotation of tandem mass spectrometry data using stochastic neural networks in shotgun proteomics. *Bioinformatics*, 2020.

[45] C. Tu, Q. Sheng, J. Li, D. Ma, X. Shen, W. Wang, Y. Shyr, Z. Yi, and J. Qu. Optimization of search engines and postprocessing approaches to maximize peptide and protein identification for high-resolution mass data. *Journal of Proteome Research*, 14(11):4662–4673, 2015.

[46] Mathias Wilhelm, Judith Schlegl, Hannes Hahne, Amin Moghaddas Gholami, Marcus Lieberenz, Mikhail M Savitski, Emanuel Ziegler, Lars Butzmann, Siegfried Gessulat, Harald Marx, et al. Mass-spectrometry-based draft of the human proteome. *Nature*, 509(7502):582–587, 2014.

[47] Mingguo Xu, Zhendong Li, and Liang Li. Combining percolator with x! tandem for accurate and sensitive peptide identification. *Journal of proteome research*, 12(6):3026–3033, 2013.

[48] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

Tables

Data set	Species	Search engine	# of PSMs
Prosit Human (human gut[37])	Human	Andromeda	4,127,437
Prosit Human+Bacteria (human gut[37])	Human and Bacteria	Andromeda	4,264,003
Prosit All (human gut[37])	All organisms	Andromeda	4,325,290
Prosit IGC+All (human gut[37])	IGC and all organisms	Andromeda	4,643,570
COVID-19[21]	Human and SARS-COV-2	Crux tide-search	1,336,153
COVID-19[21]	Human and SARS-COV-2	Comet	1,363,163
Plasmodium[40]	*P. falciparum*	Crux tide-search	1,336,386
Plasmodium[40]	*P. falciparum*	Comet	1,622,376
Yeast[39]	*S. cerevisiae*	Crux tide-search	1,543,282
Yeast[39]	*S. cerevisiae*	Comet	1,579,758
Human[27]	Human	Crux tide-search	19,961,754
Human[27]	Human	Comet	20,321,131

Table 1: Benchmark datasets

Data set	ProteoTorch DNN	Percolator	Q-ranker	PeptideProphet	Scavager	`NegLog10CombinePValue`	XCorr
Prosit Human	**16,755**	16,427	–	–	–	–	–
Prosit Human+Bact.	**33,719**	33,021	–	–	–	–	–
Prosit All	**35,443**	34,618	–	–	–	–	–
Prosit IGC+All	**106,059**	100,240	–	–	–	–	–
Crux COVID-19	**36,275**	35,070	–	–	–	32,344	–
Crux Plasm.	**104,988**	98,507	–	–	–	91,915	–
Crux Yeast	**64,740**	63,573	–	–	–	62,907	–
Crux Human	**1,568,346**	1,512,650	–	–	–	1,517,760	–
Comet COVID-19	**32,169**	31,100	29,366	22,973	28,180	–	15,657
Comet Plasm.	**131,678**	129,509	125,324	120,179	115,006	–	100,326
Comet Yeast	**117,370**	115,522	112,459	106,217	111,319	–	88,379
Comet Human	**1,684,087**	1,652,032	1,538,570	1,529,994	1,465,707	–	1,302,484

Table 2: Number of significant PSMs identified at a q-value threshold of 1% for the datasets listed in Table 1. For all methods, q-values were measured using target-decoy competition. Database-search functions `NegLog10CombinePValue` and XCorr are listed for Crux and Comet datasets, respectively. For post-processors Q-ranker, PeptideProphet, and Scavager, dashes denote the data's feature set is not supported. The largest number of significant PSMs per dataset is highlighted in bold.

Figures

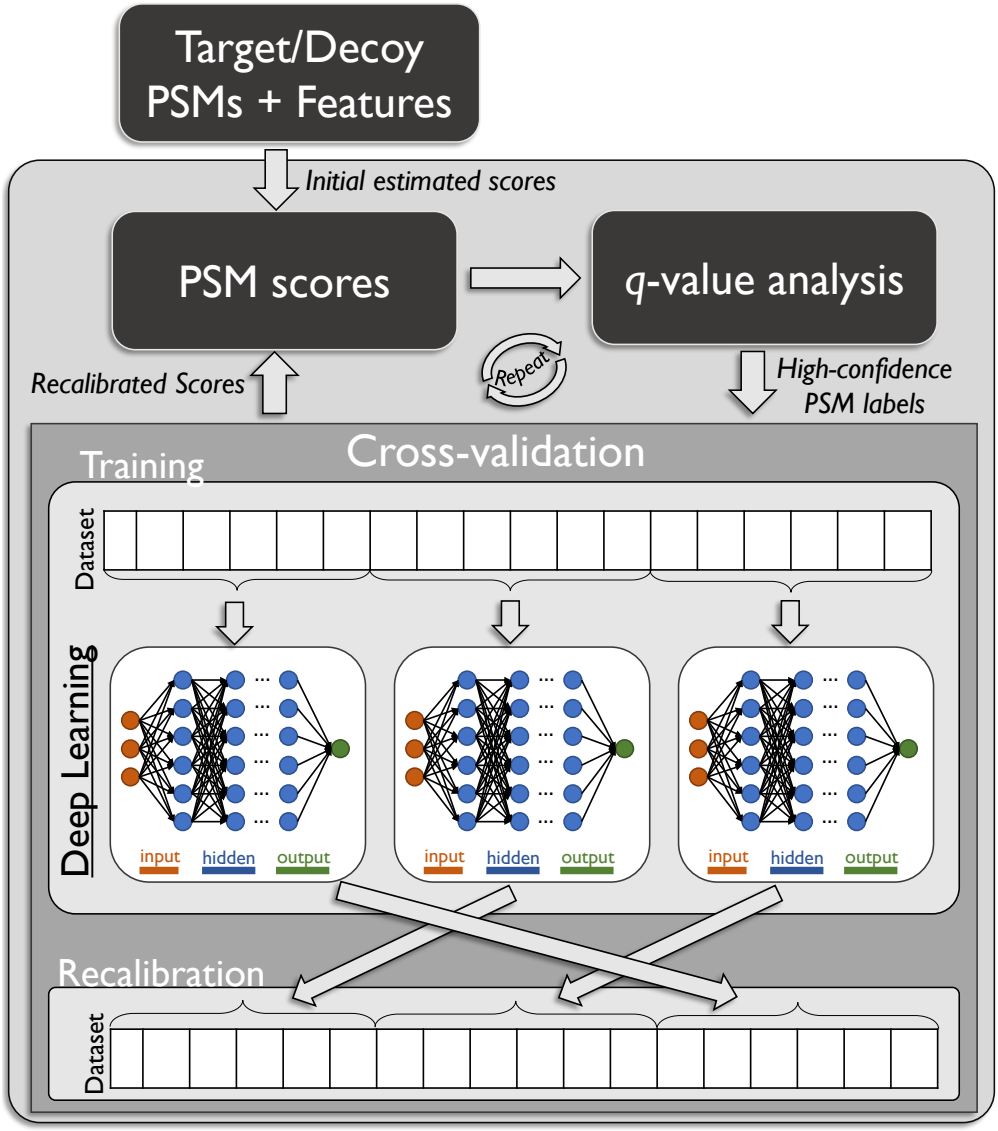
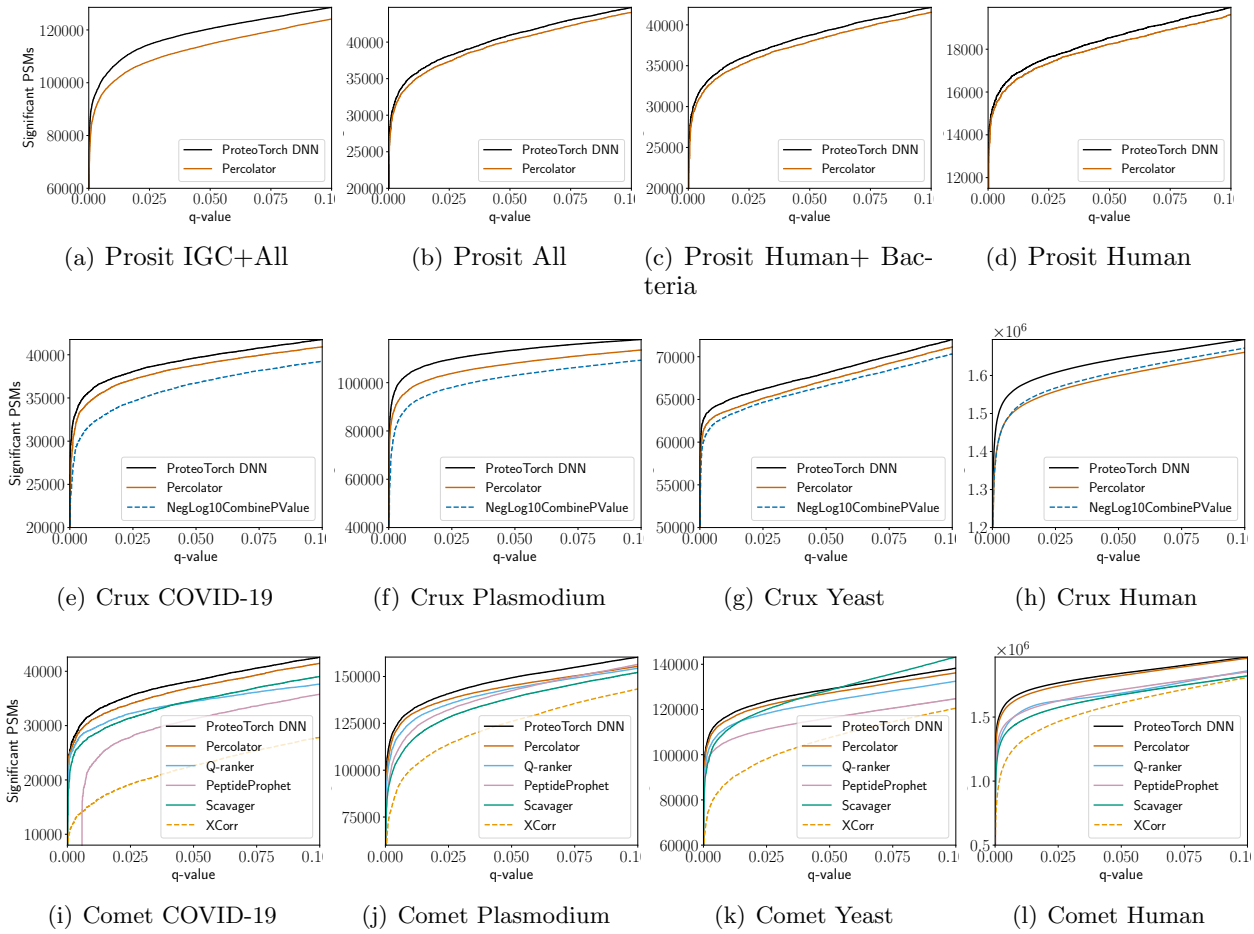


Figure 1: Semi-supervised deep learning algorithm. Target and decoy PSMs, with corresponding feature vectors, are input and used to estimate initial PSM scores. These scores are used to calculate q-values and determine highly-confident (given a q-value threshold) target PSMs for training. The dataset is split into separate cross-validation (CV) bins, and the PSMs in each bin (i.e., decoys and high-confidence targets) are used to train a deep neural network classifier. Each trained classifier is then used to rescore PSMs in a separate CV bin, and this process repeats either til convergence or a user-specified number of iterations has passed.



(a) Prosit IGC+All (b) Prosit All (c) Prosit Human+ Bacteria (d) Prosit Human

(e) Crux COVID-19 (f) Crux Plasmodium (g) Crux Yeast (h) Crux Human

(i) Comet COVID-19 (j) Comet Plasmodium (k) Comet Yeast (l) Comet Human

Figure 2: Recalibration performance plots for the Prosit, Crux tide-search, and Comet benchmark datasets listed in Table 1. The x-axis corresponds to q-values and the y-axis displays the number of PSMs deemed significant at each q-value. Target-decoy competition[9] was used to compute q-values for all methods. Prosit and Crux `NegLog10CombinePValue` post-processing are currently only supported by universal post-processors ProteoTorch and Percolator. Underlying search function performance for Comet's XCorr and Crux's `NegLog10CombinePValue` are additionally plotted using dashes.