pDeep3: Towards More Accurate Spectrum Prediction with Fast Few-Shot Learning

Ching Tarn[†,‡,¶] and Wen-Feng Zeng[*,†,‡,¶]

†*Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, China.*

‡*University of Chinese Academy of Sciences, Beijing, China.*

¶*These authors contributed equally to this work.*

E-mail: jalew188@gmail.com

Abstract

Spectrum prediction using deep learning has attracted a lot of attention in recent years. Although existing deep learning methods have dramatically increased the prediction accuracy, there is still considerable space for improvement, which is presently limited by the difference of fragmentation types or instrument settings. In this work, we use the few-shot learning method to fit the data online to make up for the shortcoming. The method is evaluated using ten datasets, where the instruments includes Velos, QE, Lumos, and Sciex, with collision energies being differently set. Experimental results show that few-shot learning can achieve higher prediction accuracy with almost negligible computing resources. For example, on the dataset from a untrained instrument Sciex-6600, within about 10 seconds, the prediction accuracy is increased from 69.7% to 86.4%; on the CID (collision-induced dissociation) dataset, the prediction accuracy of the model trained by HCD (higher energy collision dissociation) spectra is increased from 48.0% to 83.9%. It is also shown that, the method is not critical to data qual-

ity and is sufficiently efficient to fill the accuracy gap. The source code of pDeep3 is available at `http://pfind.ict.ac.cn/software/pdeep3`.

Machine learning algorithms currently play important roles in mass spectrometry-based proteomics.[1–4] In recent years, deep learning-based methods have attracted more and more attention, for both data-dependent acquisition (DDA) and data-independent acquisition (DIA) data analysis.[5–16]

For DDA data analysis, the recently published pDeep,[5] Prosit,[6] and DeepMass:Prism[7] showed that integrating spectrum prediction into DDA search engines would dramatically increase the identification rates. Our recent work, pValid,[17] further demonstrated that the spectra predicted by pDeep could be applied to filter out unreliable peptide spectrum matches (PSMs), resulting in more accurate peptide identifications. We also used the predicted spectra to re-rank the *de novo* sequenced peptides to achieve higher recall and precision.[18]

For DIA data analysis, Prosit and DeepMass:Prism showed that the performance of predicted spectral libraries was close to that of experimental ones in spectral library search. DeepDIA built predicted spectral libraries of peptides from nearly the full human proteome for DIA analysis and detected more proteins than those using the pre-built DDA spectral libraries.[11] Searle *et al.* also used a full proteome library predicted by Prosit and corrected it using empirical spectra.[19] A recent work combined the experimental spectral library with the predicted library for proteins of interest to deepen the coverage of targeted protein families.[20]

It turns out that more accurate predictions would result in better identification performance for both DDA and DIA analysis.[6,7] However, in practical uses, not all MS/MS predictions are sufficiently accurate.[9,11,21] Prosit tried to correct the predicted spectra by enumerating all predicting collision energies; DeepDIA aimed at training instrument-specific models for more accurate predictions. Unfortunately, either enumerating all energies or training models for specific instruments is time-consuming, and hence, accelerating devices such as GPUs are indispensable. Moreover, our previous work, pDeep2, showed that transfer learning is an effective way to improve the prediction of modified peptides from a pre-trained

unmodified model requiring only tens of thousands of spectra with modifications for training. Due to the computational costs, the training step could be done within minutes using a GPU, but it will take tens of minutes or even hours when only CPUs are available.

Pre-trained offline models can hardly achieve the same accuracy as online-trained models since the latter can automatically learn the characteristics of different MS fragmentation and instrument settings. However, online learning from scratch requires many computational resources to process spectra in real-time and needs a lot of data from the specific instrument. Inspired by the few-shot learning technique,[22] we use a few-shot fine-tuning method to train models for specific instruments based on a pre-trained offline model using only hundreds of training spectra, achieving higher accuracy. As the name "few-shot" implies, few-shot learning refers to the practice of learning a model with a small amount of training data and has been widely used in computer vision and other machine learning tasks. There are two advantages of few-shot learning (also called fine-tuning in this work): 1) fine-tuning a low accuracy pre-trained model would increase the accuracy of predictions; 2) fine-tuning could be done within only about 10 seconds on CPU, making training lab-specific or even sample-specific models applicable on a laptop or a personal computer. The source code of pDeep3 is available at `http://pfind.ict.ac.cn/software/pdeep3`.

Materials and Methods

Data preparation

Diverse datasets are collected from the test datasets of our previous work,[9] which are all publicly available.[23–27] For each dataset, we only consider spectra from a subset of RAW files, which are sufficient for fine-tuning and testing. Dataset information is listed in Table 1. Sciex-6600 TripleTOF-based CID (Collision-Induced Dissociation) data and LIT (Linear Ion Trap)-based CID data are also considered. For LIT-based CID data in ProteomeTools datasets, MaxQuant search results at 0.1% FDR are used, and b/y ions are extracted within

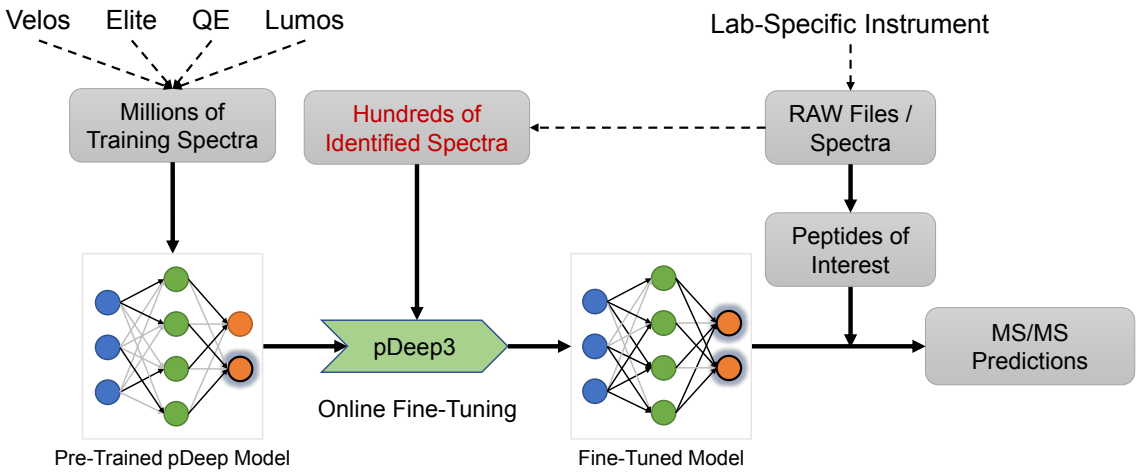

Figure 1: Workflow of the fine-tuning procedure of pDeep3.

±0.5 Da matching mass tolerance.

Table 1: Dataset information

Dataset	Description	Abbr.	Instrument	NCE	#RAWs	#PSMs
Kuster-Velos	Human, Milk	KV	Velos	40	5	35740
Olsen-293-QE	HEK293T, Trypsin	O2	QE	28	5	71788
Olsen-HeLa-QE	HeLa, Trypsin	OH	QE	28	5	106006
Olsen-GC-QE	HeLa, GluC	OG	QE	28	5	74826
Olsen-CT-QE	HeLa, Chymotrypsin	OC	QE	28	5	60191
Olsen-LC-QE	HeLa, LysC	OL	QE	28	5	98985
PT-Lumos	ProteomeTools	PL	Lumos	30	5	41772
Pandey-Velos	Human, Fetal-Gut	PV	Velos	35	5	46114
LFQBench-HE	HeLa and E.coli	-	Sciex-6600	-	2	28333
PT-CID	ProteomeTools	-	Lumos	35	2	12345

Spectrum similarity measurement

We use Pearson correlation coefficient (PCC) to measure the similarity between the predicted and experimental spectra. For a given spectrum and its identified peptide, the intensities of all $\mathrm{b_i}$ and $\mathrm{y_i}$ ions with 1+ and 2+ charge states are extracted from the spectrum, where $\mathrm{i} = 1, 2, ..., l-1$, and l is the peptide length. This $4 \times (l-1)$ intensity matrix is then reshaped

into a 1-dim vector x. The predicted $4 \times (l-1)$ b/y intensity matrix is reshaped into a 1-dim vector y, then PCC can be calculated as

$$\mathrm{PCC}(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}.$$

We used the median PCC, $\mathrm{PCC}_{>0.75}$ and $\mathrm{PCC}_{>0.90}$ to evaluate the performance of different methods, where $\mathrm{PCC}_{>t}$ is defined as the proportion of PCCs greater than t in the corresponding dataset.

The few-shot learning procedure

The few-shot learning procedure of pDeep3 is illustrated in Figure 1. The pre-trained model was trained using millions of spectra from different instruments and collision energies, as described in our previous work.[9] For a specific dataset, the network parameters could be then fine-tuned with only a few hundred spectra to achieve better predictions. The main hyper-parameters of fine-tuning are: epoch number = 2, learning rate = 0.001, mini-batch size = 1024. Since the peptides with different lengths are assigned into different mini-batch, the actual size of each mini-batch depends on peptides' length distribution.

The quality of the pre-trained model is the key to fast fine-tuning. It ensures that the network parameters are already close to the optimal ones for a new dataset, and hence we can tune the model to fit the new spectra with only a few back-propagation updates. As reported in our previous work,[9] the pre-trained pDeep model was trained and tested on Velos, Elite, QE, and Lumos datasets. It achieved about 0.98 median PCCs in most test datasets, indicating that the pre-trained model is suitable for fine-tuning. *

*The pre-trained model includes improvments and bug fixes of the previous work[9] and is re-trained.

Grid search of parameters for unseen instruments

In the current version of pDeep3, the instrument type is encoded as a 5-dimensional (5-dim) one-hot vector, corresponding to QE (including QE, QE+, QE-HF, and QE-HFX), Velos, Elite, Fusion, and Lumos, respectively. When the instrument type is not in the above instrument list, pDeep3 could use the grid search strategy to find a suitable instrument and NCE parameter combination for a given training dataset of this unseen instrument. For the grid search, pDeep3 enumerates all combinations of the above five instrument types and the NCE values ranging from 10 to 45, and then determines the instrument and NCE by the best $\text{PCC}_{>0.90}$. Fine-tuning is further applied based on the determined instrument and NCE to obtain a better model.

Results and Discussions

Hyper-parameter selection

In the fine-tuning step, the number of epochs e and the size of the training set n are two key hyper-parameters for the trade-off of the training cost and prediction accuracy. Here we use three sets of experiments to select these hyper-parameters. It is worth mentioning that, for each dataset, we only partially select peptide-spectrum matches (PSMs) from a RAW file when fine-tuning, and test the fine-tuned model using all PSMs from another RAW file.

To select an appropriate e in the fine-tuning step, we try e in $\{0, 1, \ldots, 8\}$, and set n as 100. The test results are shown in Figure 2a. It turns out that, as e increases, the performance on most of the datasets improves as well, and the test $\text{PCC}_{>0.75}$ and $\text{PCC}_{>0.90}$ become stable after about two epochs. In particular, as the pre-trained model has been trained by other peptides of ProteomeTools,[9] the test results on PL are already close to the best accuracy, and hence there is no further improvement in the PL dataset as e increases (Figure 2a). However, the performance is not decreasing, indicating that fine-tuning may

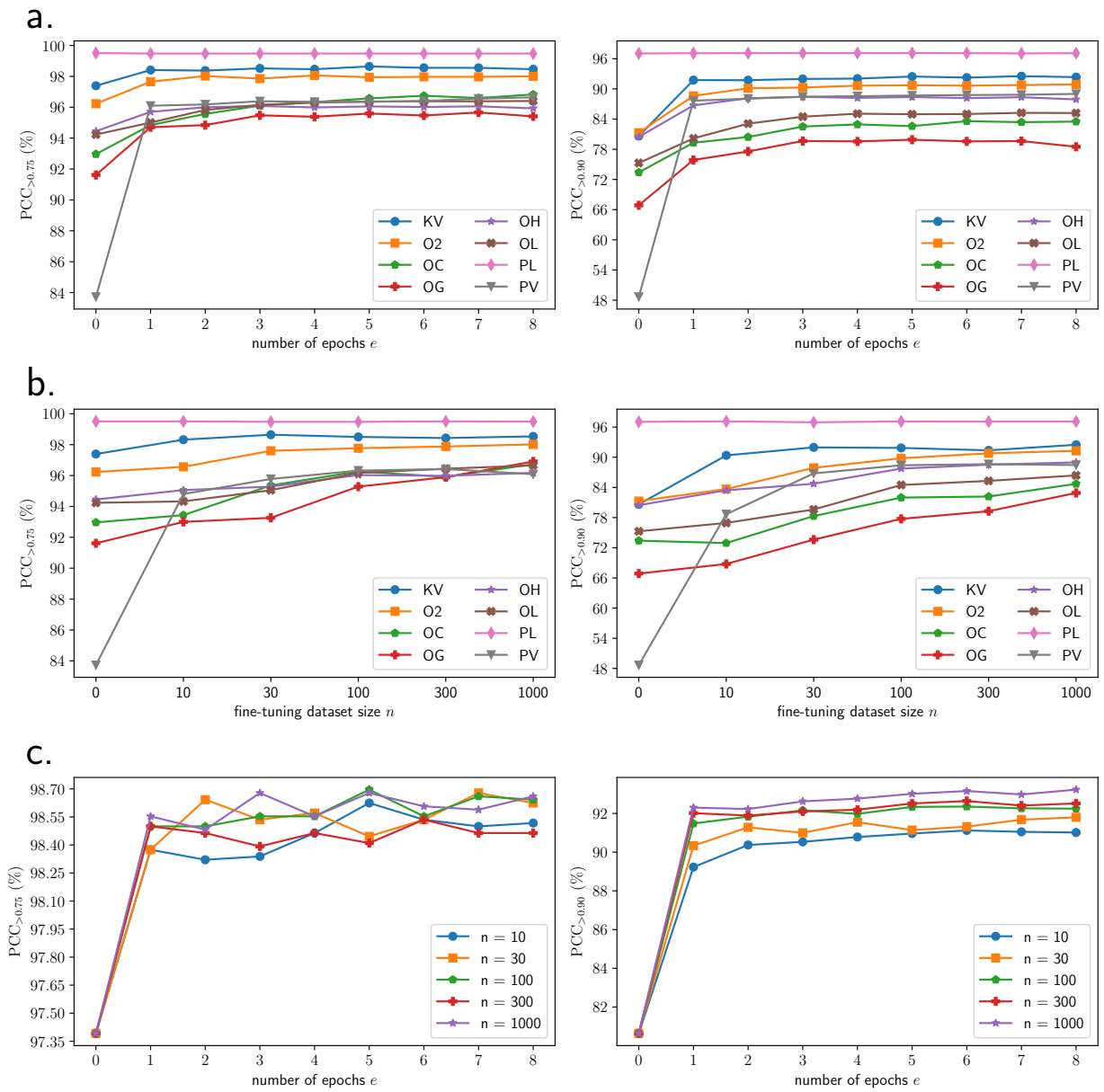

Figure 2: Tests of different hyper-parameters, training epochs e and training set size n, on different datasets. (a) Tests of e with $n = 100$. (b) Tests of n with $e = 2$. (c) Tests of different combinations of e and n on the KV dataset.

not suffer from the potential risk of over-fitting.

We also try n in $\{0, 10, 30, 100, 300, 1000\}$, and set e as 2 for fine-tuning. The results are demonstrated in Figure 2b. The performance steadily gets better as n increases. And for PL, the results are not much changed as well. The improvements on most datasets gradually slow down as n increases after $n > 100$. Specifically, the improvements on KV have achieved the optimal when only about 10 or 30 samples are used; for Olsen's datasets (O2, OC, OG, OH, OL), the increases are roughly linearly related to $\log n$.

We further try different combinations of e and n for $e \in \{0, 1, \ldots, 8\}$ and $n \in \{10, 30, 100, 300, 1000\}$. The test results on the KV dataset are illustrated in Figure 2c. As shown in Figure 2c, the effects of e and n are quite independent. That means even if n is different, the method achieves a large increase after about two epochs, and the improvement slows down thereafter. With different numbers of epochs e, the larger the size of the training set, the better the results could be, and the growth rate gradually slows down as n increases. Results on other datasets can be found in Data S1.

As the time required for fine-tuning is relative to the number of epochs e and the size of the training set n, to ensure that fine-tuning can be done using CPU within the acceptable time, it is necessary to limit the scale without significantly sacrificing the performance. The test results in Figure 2 demonstrate that the fine-tuned model can reach good performance in both fine-tuning speed and predicting accuracy using 100 training samples and two epochs.

In these tests, the fine-tuning can be finished within only about 10 seconds using an Intel CPU (Xeon CPU E5-2680 v4 @ 2.40GHz, without GPU devices), showing that fine-tuning is always recommended before predicting for any datasets.

Intra-RAW and inter-RAW tests

For a dataset, we question whether we need to fine-tune different models for different RAW files of the dataset (run-specific fine-tuning), or we only need to fine-tune a model for all RAW files (sample-/instrument-specific fine-tuning). To answer it, we apply intra-RAW and

inter-RAW test experiments.

For each dataset, we split PSMs in each RAW file into a training set (100 PSMs) and a testing set (the rest of the RAW). The pre-trained model is fine-tuned by 5 training sets, respectively ($e = 2$, $n = 100$), resulting in 5 fine-tuned models. Each model is then tested separately on all the five testing sets, and we get a 5×5 $\text{PCC}_{>x}$ matrix $\mathbf{S}$. Each set of results of a model corresponds to a row of the matrix $\mathbf{S}$, where $\mathbf{S}_{ij}$ is the result of fine-tuning with RAW file i and testing with RAW file j. Test results on the five RAW files of KV and O2 datasets are listed in Table 2. We use the standard deviation (σ) to measure the similarity between the results. On the same testing RAW file, fine-tuning by the training set of any RAW file achieves the similar $\text{PCC}_{>0.75}$ ($\sigma = 0.56\%$ for KV and $\sigma = 0.17\%$ for O2) and $\text{PCC}_{>0.90}$ ($\sigma = 1.51\%$ for KV and $\sigma = 0.38\%$ for O2), indicating that the model can be fine-tuned only once on a RAW file and applies to others in the same dataset if the instrument settings are not changed. Results of other datasets can be found at Data S2.

Table 2: Intra-RAW and inter-RAW tests. r_i means the model is trained on the training set from RAW i, and s_j means it is tested on the testing set from RAW file j.

(a) KV

	$\text{PCC}_{>0.75}$ (%, $\sigma = 0.56\%$)					$\text{PCC}_{>0.90}$ (%, $\sigma = 1.51\%$)				
	s_1	s_2	s_3	s_4	s_5	s_1	s_2	s_3	s_4	s_5
r_1	98.65	98.34	98.66	97.40	99.21	93.21	91.84	94.26	92.69	95.84
r_2	98.76	98.69	98.75	97.74	99.27	93.10	91.65	93.32	92.46	96.10
r_3	98.65	98.42	98.66	97.57	99.23	93.25	91.74	93.75	92.57	95.98
r_4	98.78	98.53	98.73	97.62	99.21	94.25	92.17	93.89	93.06	96.22
r_5	98.57	98.48	98.66	97.50	99.21	92.88	91.11	93.05	92.19	95.86

(b) O2

	$\text{PCC}_{>0.75}$ (%, $\sigma = 0.17\%$)					$\text{PCC}_{>0.90}$ (%, $\sigma = 0.38\%$)				
	s_1	s_2	s_3	s_4	s_5	s_1	s_2	s_3	s_4	s_5
r_1	97.64	97.89	97.54	97.92	97.89	89.70	90.12	89.40	89.34	89.24
r_2	97.74	98.00	97.61	97.96	98.01	90.04	90.44	89.52	89.73	89.91
r_3	97.74	97.98	97.70	98.01	97.99	89.92	90.14	89.97	89.57	89.78
r_4	97.50	97.85	97.55	97.90	97.89	90.00	90.43	89.69	90.27	90.13
r_5	97.62	97.88	97.56	97.95	97.91	89.27	89.50	89.07	89.40	89.66

Tests of training set selection bias

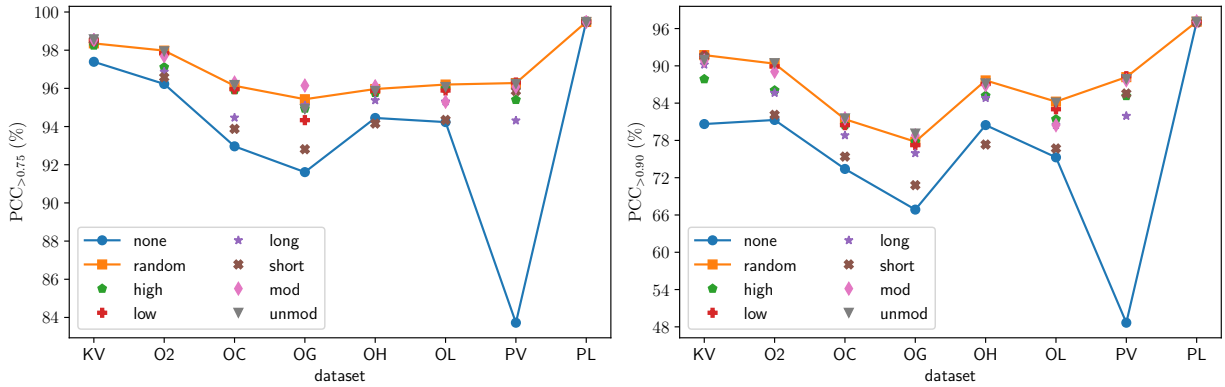

Figure 3: Test of different training set selection methods for fine-tuning.

One concern of the fine-tuning step is that the 100 randomly selected PSMs hardly cover all peptide characteristics (different lengths, different charge states, etc.). Once some properties of peptides are not selected for fine-tuning, the fine-tuned model may prefer the selected properties, resulting in poor predictions on peptides with non-selected properties. We call it "training set selection bias" of the fine-tuning model. To test whether the bias exists, we evaluate six training set selection methods, as listed in follows:

1. *low*. The training set only contains peptides with precursor charge ≤ 3;
2. *high*. The training set only contains peptides with precursor charge > 3;
3. *short*. The training set only contains peptides with length ≤ 12;
4. *long*. The training set only contains peptides with length > 12;
5. *mod*. The training set only contains peptides with Oxidation[M];
6. *unmod*. The training set only contains peptides without any variable modification.

We also consider "random selection" (denoted as *random*) and "no fine-tuning" (denoted as *none*) for comparisons. All the models are fine-tuned by selected training sets ($e = 2$,

$n = 100$) and tested by another RAW file. The results is illustrated in Figure 3, and show that, in most cases, a model with fine-tuning is always better than that without fine-tuning. Random selection usually achieves the best performance.

Grid search and fine-tuning on Sciex-6600 and LIT-based CID datasets

Although the TripleTOF-based CID of Sciex-6600 is also the beam-type CID which is the same as Orbitrap-based HCD, there are still many differences between Sciex-6600 and Orbitrap-based instruments, including the collision energy settings, the mass analyzer, ion detector, signal processing, etc. As the pDeep model is only trained on HCD data, we use grid search to find the best instrument and NCE for the Sciex-6600 dataset before fine-tuning (Elite@NCE=31, as shown in Figure 4a). Even after the best instrument and NCE are determined, fine-tuning is still very useful to obtain a better model. As shown in Figure 4c, $PCC_{>0.90}$ increases from 69.7% to 86.4% after fine-tuning with $n = 100$ and $e = 8$.

ProteomeTools also provides a lot of high-quality LIT-based CID data. Unlike beam-type CID including HCD, LIT-based CID cleaves peptides and detects fragment ions in the linear ion trap, it has some characteristics: (1) LIT-based fragmentation is a resonance-type fragmentation, under which most ions are fragmented only once (i.e. single cleavage); (2) LIT has a "1/3 rule", meaning that low mass (m/z lower than 1/3 of the precursor m/z) can be hardly detected in the LIT; (3) LIT is a low-resolution mass analyzer. These three characteristics make CID spectra quite different from HCD spectra. Therefore, it is not surprising that the pre-trained model with RAW-file-recorded NCE=35 cannot give a good prediction for CID spectra (Figure 4a). With grid search, prediction by the pre-trained model with the best instrument and NCE values (Lumos@NCE=19) gets much better $PCC_{>0.90}$ than that with Lumos@NCE=35. After fine-tuning by using 1000 CID spectra and 16 epochs with Lumos@NCE=19, the $PCC_{>0.90}$ achieves 83.9% (Figure 4d).

Figure 5 clearly illustrates how much the prediction is improved for peptide "GFHPDPEALK (3+)". Predicted by the pre-trained model which is only trained by HCD data,

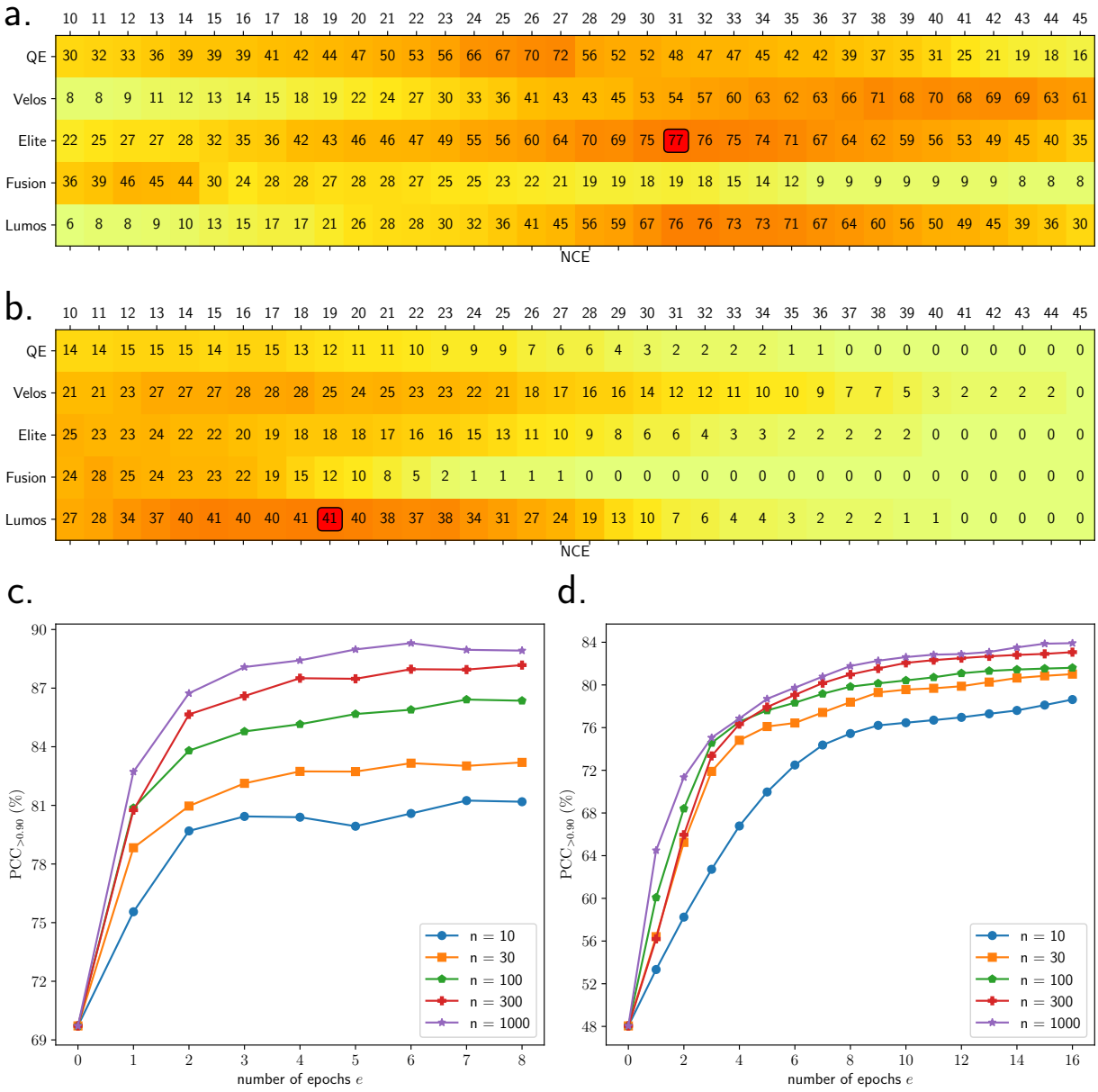

Figure 4: Grid search and few-shot learning for Sciex-6600 instrument and LIT-based CID spectra. (a) $PCC_{>0.90}$ of each setting of grid search for best instrument and NCE for Sciex-6600 data. The best setting is Elite@NCE=31. The grid search takes 59.4 seconds. (b) $PCC_{>0.90}$ of each setting of grid search for best instrument and NCE for PT-CID data. The best setting is Lumos@NCE=19. The grid search takes 114.2 seconds. (c) Fine-tuning for Sciex-6600 data using the best setting. (d) Fine-tuning for CID data using the best setting.

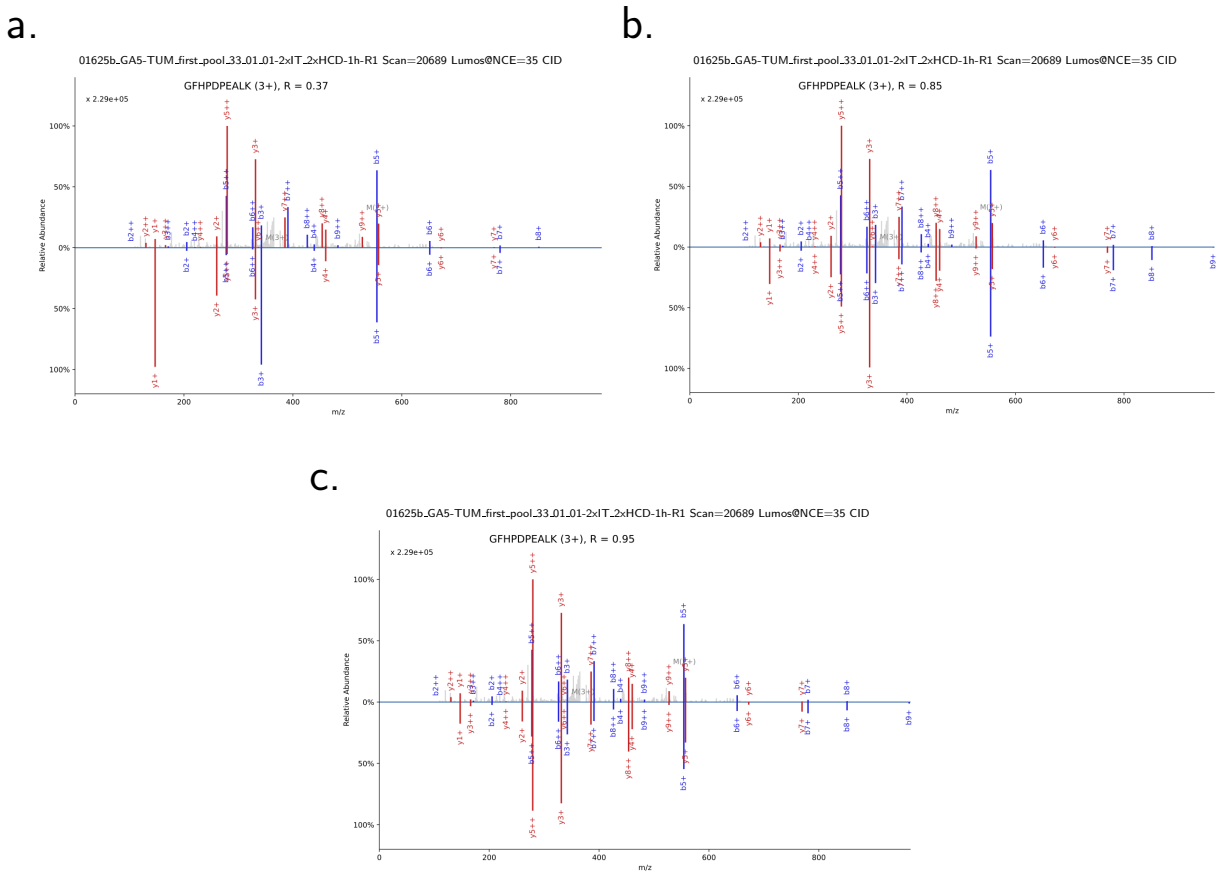

Figure 5: Examples of the prediction for the CID spectrum of peptide "GFHPDPEALK (3+)". "R" refers the PCC. (a) Prediction with pre-trained model, Lumos@NCE=35. (b) Prediction with pre-trained model, Lumos@NCE=19. (c) Prediction with fine-tuned model, Lumos@NCE=19. The model is fine-tuned with 1000 PSMs and 16 epochs. The fine-tuning costs less than 15 seconds.

the low-mass fragments get quite high intensities, but they could hardly get high intensities due to the "1/3 rule" in LIT-based CID (Figure 5a). After fine-tuning, the model learns the characteristics of LIT-based CID, and hence gives a better prediction (Figure 5c). Fine-tuning on LIT-based CID data with 1000 PSMs for 16 epochs costs less than 15 seconds.

Therefore, few-shot learning coupled with grid search shows a great potential to train better models for unseen instrument types.

Conclusions and Discussions

In this work, we show that a pre-trained pDeep model can be significantly improved using the few-shot learning technique. Few-shot learning aims to fine-tune the pDeep model with a small training set and a small number of training epochs and can increase the predicting accuracy within only seconds.

These years, Bruker's timsTOF has attracted a lot of attention in proteomics.[28] Although pDeep3 could not currently access the timsTOF data, we believe that fine-tuning could also be used for timsTOF data as it shows promising results for Sciex-6600 and LIT-based CID data.

Predicting the intensities of peptides with common or low-abundance PTMs has been discussed in pDeep2. It is also very promising that pDeep3 would be able to give a better prediction for peptides with PTMs. This would be very useful when users focus on PTM analysis.

In this work, all fine-tuning steps are done by CPU, hence we believe few-shot learning or transfer learning could be very useful to train a better model for users who do not have too many computational resources.

Acknowledgement

The authors would like to thank Dr. Si-Min He for his invaluable supervision, our other col-

leagues of pFind Team for their helpful discussions and suggestions, and Dr. Ruixiang Sun from National Institute of Biological Sciences for his advice about HCD and CID fragmentation techniques. This work was supported by the National Key Research and Development Program of China (No. 2016YFA0501301).

Supporting Information Available

- **Note A**: Settings of the pre-trained model.
- **Note B**: Fine-tuning on HLA data.
- **Note C**: Building predicted spectral libraries for DIA analysis.
- **Data S1**: Test results of different combinations of e and n on all test datasets.
- **Data S2**: Intra-RAW and inter-RAW test results on all test datasets.

References

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

(2) Sun, S.; Yang, F.; Yang, Q.; Zhang, H.; Wang, Y.; Bu, D.; Ma, B. MS-Simulator: predicting y-ion intensities for peptides with two charges based on the intensity ratio of neighboring ions. *Journal of Proteome Research* **2012**, *11*, 4509–4516.

(3) Xu, L. L.; Young, A.; Zhou, A.; Röst, H. L. Machine Learning in Mass Spectrometric Analysis of DIA Data. *PROTEOMICS* **2020**, *20*, 1900352.

(4) Fei, Z. Improving Tandem Mass Spectra Analysis with Hierarchical Learning. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,

IJCAI-20. 2020; pp 4345–4351, Special track on AI for CompSust and Human well-being.

(5) 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. *Analytical Chemistry* **2017**, *89*, 12690–12697.

(6) 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. *Nature Methods* **2019**, *16*, 509–518.

(7) 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. *Nature Methods* **2019**, *16*, 519–525.

(8) Zhou, X.-X.; Ren, R.; Gao, W.-L.; Huang, Y.-Y.; Zeng, W.-F.; Kong, D.-F.; Hao, T.-S.; Zhang, Z.-F.; Zhan, J.-F. Trends on Methods for Prediction of Tandem Mass Spectra of Peptides. *Progress in Biochemistry and Biophysics* **2019**, *46*, 169–180.

(9) Zeng, W.-F.; Zhou, X.-X.; Zhou, W.-J.; Chi, H.; Zhan, J.; He, S.-M. MS/MS Spectrum Prediction for Modified Peptides Using pDeep2 Trained by Transfer Learning. *Analytical Chemistry* **2019**, *91*, 9724–9731.

(10) Guan, S.; Moran, M. F.; Ma, B. Prediction of LC-MS/MS Properties of Peptides from Sequence by Deep Learning. *Molecular and Cellular Proteomics* **2019**, *18*, 2099–2107.

(11) Yang, Y.; Liu, X.; Shen, C.; Lin, Y.; Yang, P.; Qiao, L. In silico spectral libraries by deep learning facilitate data-independent acquisition proteomics. *Nature Communications* **2020**, *11*, 146.

(12) Liu, K.; Li, S.; Wang, L.; Ye, Y.; Tang, H. Full-Spectrum Prediction of Peptides Tandem Mass Spectra using Deep Neural Network. *Analytical Chemistry* **2020**, *92*, 4275–4283, PMID: 32053352.

(13) Tran, N. H.; Zhang, X.; Xin, L.; Shan, B.; Li, M. De novo peptide sequencing by deep learning. *Proceedings of the National Academic of Sciences* **2017**, *114*, 8247–8252.

(14) Lin, Y.-M.; Chen, C.-T.; Chang, J.-M. MS2CNN: predicting MS/MS spectrum based on protein sequence using deep convolutional neural networks. *BMC Genomics* **2019**, *20*, 1–10.

(15) Wen, B.; Zeng, W.-F.; Liao, Y.; Shi, Z.; Savage, S. R.; Jiang, W.; Zhang, B. Deep Learning in Proteomics. *Proteomics* **2020**, *20*, 1900335.

(16) Müller, J. B.; Geyer, P. E.; Colaço, A. R.; Treit, P. V.; Strauss, M. T.; Oroshi, M.; Doll, S.; Winter, S. V.; Bader, J. M.; Köhler, N. et al. The proteome landscape of the kingdoms of life. *Nature* **2020**, *582*, 592–596.

(17) Zhou, W.-J.; Yang, H.; Zeng, W.-F.; Zhang, K.; Chi, H.; He, S.-M. pValid: Validation Beyond the Target-Decoy Approach for Peptide Identification in Shotgun Proteomics. *Journal of Proteome Research* **2019**, *18*, 2747–2758.

(18) Yang, H.; Chi, H.; Zeng, W.-F.; Zhou, W.-J.; He, S.-M. pNovo 3: precise de novo peptide sequencing using a learning-to-rank framework. *ISMB/Bioinformatics* **2019**, *35*, i183–i190.

(19) Searle, B. C.; Swearingen, K. E.; Barnes, C. A.; Schmidt, T.; Gessulat, S.; Kuster, B.; Wilhelm, M. Generating high quality libraries for DIA MS with empirically corrected peptide predictions. *Nature Communications* **2019**, *11*, 1548.

(20) Lou, R.; Tang, P.; Ding, K.; Li, S.; Tian, C.; Li, Y.; Zhao, S.; Zhang, Y.; Shui, W.

Hybrid Spectral Library Combining DIA-MS Data and a Targeted Virtual Library Substantially Deepens the Proteome Coverage. *iScience* **2020**, *23*, 100903.

(21) Xu, R.; Sheng, J.; Bai, M.; Shu, K.; Zhu, Y.; Chang, C. A Comprehensive Evaluation of MS/MS Spectrum Prediction Tools for Shotgun Proteomics. *Proteomics* **2020**, *20*, 1900345.

(22) Wang, Y.; Yao, Q.; Kwok, J. T.; Ni, L. M. Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Comput. Surv.* **2020**, *53*.

(23) Wilhelm, M.; Schlegl, J.; Hahne, H.; Gholami, A. M.; Lieberenz, M.; Savitski, M. M.; Ziegler, E.; Butzmann, L.; Gessulat, S.; Marx, H. et al. Mass-spectrometry-based draft of the human proteome. *Nature* **2014**, *509*, 582–587.

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

(25) 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 systems* **2017**, *4*, 587–599.e4.

(26) Zolg, D. P.; Wilhelm, M.; Schnatbaum, K.; Zerweck, J.; Knaute, T.; Delanghe, B.; Bailey, D. J.; Gessulat, S.; Ehrlich, H.-C.; Weininger, M. et al. Building ProteomeTools based on a complete synthetic human proteome. *Nature Methods* **2017**, *14*, 259–262.

(27) Navarro, P.; Kuharev, J.; Gillet, L. C.; Bernhardt, O. M.; MacLean, B.; Röst, H. L.; Tate, S. A.; Tsou, C.-C.; Reiter, L.; Distler, U. et al. A multicenter study benchmarks software tools for label-free proteome quantification. *Nature Biotechnology* **2016**, *34*, 1130–1136.

(28) Meier, F.; Brunner, A.-D.; Frank, M.; Ha, A.; Bludau, I.; Voytik, E.; Kaspar-Schoenefeld, S.; Lubeck, M.; Raether, O.; Bache, N. et al. diaPASEF: parallel accumulation–serial fragmentation combined with data-independent acquisition. *Nature Methods* **2020**, *17*, 1229–1236.