DeepPhospho: A Transformer-based Deep Network for Peptide Tandem Mass Spectra Prediction

Weizhen Liu
School of Information Science and Technology
ShanghaiTech University
liuwzh@shanghaitech.edu.cn

Abstract

In mass-spectrometry-based proteomics, the identification and quantification of peptides and proteins heavily rely on sequence database searching or spectral library matching. The size and quality of sequence database is crucial for the searching. There are researches that propose we could build a virtual sequence database composed of the peptide sequence with its retention time and spectra by computational method. Nowadays, the deep learning model has show its power in computer vision and natural language processing, however, in proteomics, the lack of accurate predictive models for fragment ion intensities and retention time impairs the realization of the full potential of these proposals. Here, we propose our new method based on LSTM + Transformer model and focus on the phosphorylation peptide data, especially. Using our model, we could build a larger and more accurate peptide library for protein identification in mass-spectrometry-based proteomics.

1. Introduction

A fundamental problem in proteome analysis is to seek a predictive relationship between a peptide and its measured chemico-physical properties, such as the chromatographic retention time (RT) and fragment ion intensity in LC-MS/MS (ref). In particular, bottom-up proteomic approaches can greatly benefit from high-quality mass spectra prediction from amino acid sequences (ref). However, such a prediction task is particularly challenging due to the complex quantum effect within peptides and noisy measurement process.

Recently, thanks to the availability of large-scale peptide databases, data-driven strategies have been adopted to tackle this problem [**?**]. Notably, deep learning based approaches have shown promising performances in the task of retention time and/or mass spectrum estimation [**?**, **?**, **?**, **?**, **?**]. The existing deep learning methods typically first embed amino acids into a vector representation, and then feed them into a multilayer neural network (CNN or LSTM) which extracts a global representation of the input peptide and predicts its retention time/ion intensity. Specifically, some of early works adopt convolutional neural networks (CNN) or its variants, which have limited capacity in modeling sequences of varying lengths and are largely restricted to predicting a single property of peptides (e.g., DeepRT [**?**]). Recent efforts attempt to utilize recurrent neural networks, such as LSTMs [**?**], to capture the structural dependency in the peptide sequences [**?**]. Nevertheless, they only achieves limited success due to their restrictive structural assumption on the design of deep networks, including the linear embedding of amino acids and recurrent network topology.

In this work, we present a novel deep learning framework, termed DeepPhospho, which tackles the challenge of peptide representation learning for RT and ion intensity prediction. To this end, we develop a hybrid deep network design, capable of better capturing the global structure of the peptide. Specifically, we first employ a bi-LSTM network to compute the embedding of amino acids. This produces a context-aware local representations as each amino acid is enriched by the features of other amino acids in the same peptide. Given the new embedding, we then introduce a flexible Transformer-based network that uses self-attention to model long-range dependency in the peptide sequences. The Transformer module enables us to directly attend to multiple sites of the peptide, such as the pairs of b and y ion, without needing the recurrent computation. Finally, the network outputs a new representation for the input peptide, which is then fed into a linear regressor to generate predictions for RT or ion intensities.

We demonstrate the efficacy of the DeepPhospho on multiple challenging benchmarks of RT/ion-intensity prediction and provide detailed ablative study on the model design. Moreover, we have built a ready-to-use web server based on our model for the scientific community [1], and will

[1] https://xxx.shanghaitech.edu.cn/xxx

also release our code ().

2. Related Work

2.1. Retention Time Prediction

For retention time prediction, efforts to date to predict peptide RTs are mainly based on retention coefficients (Rc) of amino acids, while SSRCalc[**?**] is the most popular Rc-based predictor. Rc is a parameter to appraise the contribution of an individual amino acid to peptide RT, and the sum of all the Rcs of amino acids in a peptide could serve for RT estimation. Additional factors such as peptide length, charge, and helicity are also considered during peptide RT prediction. Several predictors based on Rc and other measurable factors have been proposed and reported in some studies. For example, Elude[**?**, **?**] and GPTime[**?**] developed from support vector machine (SVM) and Gaussian process regression employed Rcs learned from data sets and can also provide RT prediction for post-translationally modied (PTM) peptides. All these tools produced RT with R^2 values of less than 0.965 on various data sets. On the other hand, it is well recognized that we are still lacking of enough knowledge to fully understand the physicochemical properties of peptides and the complex interactions between peptides and stationary phase, which leads to the less-than optimum prediction of peptide RT. In terms of the algorithm, the traditional model shows its limitation in tracing the many subtle factors that affect the peptide behaviors on LC. Hence, in the eld of peptide RT prediction, there is still large room for improvement.

Deep learning, an advanced machine learning method, has shown extraordinary capability to learn complex relationships from large-scale data. There have been several tools that successfully utilized deep learning in RT prediction, such as DeepRT[**?**], and Prosit[**?**].

DeepRT use the capsule network (CapsNet) [**?**] model. DeepRT could foresaw the RTs for the peptides at even different modication status included as oxidation of methionine, phosphorylation of serine, threonine, tyrosine and at varied LC conditions included RPLC, SCX, HILIC. Prosit utilizes the LTSM model and like the DeepRT, it could predict the retention time given the peptide sequence. However, the DeepRT did not explain very well for the choice of CapsNet, and it did not fully consider the sequence's characteristics. Prosit use the LSTM model to capture the this pattern. But, LSTM model has been beaten by the transformer model in multiple tasks[**?**]. Herein, we select the LSTM+transformer architecture.

2.2. Ion Intensity Prediction

Investigation of the peptide fragmentation is valuable both in theory and in practice. There are some researchers focusing on the prediction of theoretical MS/MS spectra of peptides, including kinetic model-based methods and machine learning based methods. MassAnalyzer[**?**, **?**] and MS-Simulator[**?**, **?**] are two major kinetic model-based tools designed based on the mobile proton hypothesis with some basic assumptions, and the key parameters of the models are tuned to t the data by statistics. The disadvantage of the kinetic model is that it cannot consistently be used to model the peptide fragmentation under HCD, ETD, or electron-transfer and higher-energy collision dissociation (EThcD). PeptideART is a pure machine learning based tool which models the theoretical spectrum prediction as a classification problem, and the probability of the occurrence of each peak is learned by using a shallow feed-forward neural network[**?**, **?**]. Other previous work[**?**] predicts intensity ranks instead of relative intensities using learning-to-rank algorithms. It has been shown that a good prediction method can boost the identification of peptides. However, peptide fragmentation is very complex to predict; Li et al.[**?**] pointed out that the cross experiment correlations of PeptideART based on collision induced dissociation (CID) spectra were significantly lower than within-experiment analyses. To handle the complexity of peptide fragmentation, more powerful algorithms such as deep learning should be considered.

pDeep[**?**], a deep learning-based method based on LSTM model to predict the intensity distribution of product ions of a peptide. pDeep can work well in predicting not only HCD spectra but also ETD and EThcD spectra.pDeep achieved ¿0.9 median PCCs (Pearson correlation coefficient) in predicting HCD, ETD, and EThcD spectra, which is significantly higher than kinetic model-based MassAnalyzer and MS-Simulator as well as the machine learning-based PeptideART. But, similarly like the RT prediction task, LSTM could not learn better than transformer, and this is why we also choose the LSTM + transformer architecture for this task. After pDeep, the modified version of pDeep, called pDeep2[**?**].

pDeep2 is spectrum predictor for modied peptides based on the deep learning model. It use the transfer-learning technique to transfer pDeep model parameters to the prediction of modified peptide's spectrum. It claims that it's accurate model for predicting the spectra of peptides with common PTMs or low-abundance PTMs, even if we only had a limited scale of benchmark modied PSMs. Similarly, our model design make us to predict the spectrum of modified peptide.

3. Methods

3.1. Model

Architecture

It could be formulated as a regression problem from sequence to value: given sequence X, a learned function F

could map the X to the y, either retention time or ion intensity. The mathematical expression is as follows:

$$\hat{y} = F(X; \Theta)$$

Θ is the parameters of the model.we find the optimal parameters $\Theta^\star$ by minimizing the loss function L.

$$\Theta^\star = \arg\min_{\Theta} L(\hat{y}, y)$$

We use the LSTM + Transformer model (illustrated in the Figure) to solve the retention time (RT) and ion intensity prediction task. The long short-term memory (LSTM)[**?**] module comprises of two stacks of bi-directional LSTM. For each stack of LSTM, it has two layers and the dimension of input embedding and hidden state are 256 and 512, respectively. After one stack of LSTM, a combination of LeakyReLU-dropout-linear layer is configured. LSTM is a kind of recurrent neural network(RNN) trying to use gate function to capture the long dependency in the input sequence. The bi-directional LSTM, expecting to learn a good token embedding for the network's downstream layers, is scheduled as the first module of our model.

Transformer[**?**] is the second module composed of n Transformer encoder. Each Transformer encoder has 8 attention head, then a feedforward layer configured after attention head.

Self-attention and position encoding

Transformer is an entirely different model by exploiting the self-attention compared to RNN. Self-attention is an attention mechanism relating different positions of a single sequence to compute a representation of the sequence. Self-attention has been used successfully in a variety of tasks, including reading comprehension, textual entailment, and learning task-independent sentence representations. The Transformer is the first transduction model relying entirely on self-attention to compute its input and output representations without using sequence-aligned RNNs or convolution. It has achieved state-of-the-art performance in multiple natural language tasks, such as language translation, language entailment classification, language modeling, etc. We use the pre-layer form of Transformer, which is proposed in [**?**] and converges much faster.

The position encoding by sine and cosine functions is added to the output of LSTM module then feed into the Transformer module. We take the same way of position encoding as [**?**] which is as follows:

$$PE_{(pos,2i)} = \sin\left(pos/10000^{\frac{2i}{d_{model}}}\right)$$
$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{\frac{2i}{d_{model}}}\right)$$

where pos is the position, i is the dimension and d_{model} have the same dimensions with input embeddings.

3.2. RT prediction

For this task, we implement 5 models with two units of two layers bi-directional LSTM and 4, 5, 6, 7, 8 Transformer encoder layers, correspondingly. We train and select those 5 models independently. After obtaining the best models, those 5 predictions of retention time are averaged as the final prediction for each peptide.

The amino acid tokens are embedded into 256 dimensions to the neural network. Especially since the output of the Transformer is the same length as the input sequence, we need to take it down to one scaler for retention time. By adding the time distributed linear layer to assign varied weights for different amino acids dynamically, we obtain RT prediction. Herein we use the square root of mean squared error (RMSE) as loss function.

3.3. Ion intensity prediction

We implement one model for ion intensity prediction, which has two units of two layers bi-directional LSTM and 8 layers of Transformer encoder. Like the RT prediction task, we first embed the amino acids token, the charge to 192, 64 dimensions separately then concatenate those two vectors, forming the 256 dimension vectors to the neural network. The last layer is a linear layer, which projects the feature from high dimension to 8 dimensions with length unchanged as our prediction of ion intensity.

We use mean squared error (MSE) as our loss function.

For both RT and Ion intensity tasks, we use Adam optimizer[**?**], and the learning rate is 1e-4, the learning rate decay at the milestone epochs during the training. We implement our models by the Python and Pytorch, and train the model on multiple GPUs. The related code could be found in https://github.com/weizhenFrank/DeepPhospho.git.

4. Experiments

4.1. Dataset

Data composition and preprocess

The sequence is represented by the symbol of amino acids such as L, K, M, etc., typically 7-50 in length. Specially, we use 1 to represent the oxidation of methionine (M), and we use 2,3,4 to represent the phosphorylation of serine(S), threonine(T), tyrosine(Y), respectively. And those representation is also used in the Ion Intensity task. Retention time (RT) is a measure of the time taken for a solute to pass through a chromatography column. It is calculated as the time from injection to detection.

The RT datasets are comprised of pairs of $\{X, y\}$. $X := \{< x_1, x_2, x_3, \ldots, x_n >\}$, x_i is amino acid, and y is the retention time. For building the virtual library, we split the dataset into train : validation = 9 : 1, selecting the best

model on validation set; for model comparison, the dataset is split into train : validation : test = 8 : 1 : 1, reporting results on the test set.

As the retention time is distributed in the real-world unit, such as minutes or seconds, we scale each dataset by its max and min of retention time to 0 - 1 by the following formula. To cover all RT dataset distributions, we set the max(RT) as 200, and min(RT) as -100.

In further, we support the peptide with N-terminal acetyl modification. We use the * symbol to indicate modification, @ to indicate no modification. We pad all sequences to the length of the longest sequence in the dataset to form a matrix feeding into the neural network.

Like RT dataset, the ion intensity datasets are comprised of pairs of $\{X, y\}$. $X := \{< x_1, x_2, x_3, \ldots, x_i, \ldots, x_n, +q >\}$, x_i is amino acid, $+q$ is the charge carried by the peptide sequence before it is fragmented in the mass spectrometer. And y is the spectrum of the peptide. Each y is composed of pairs of key and value. The key is the ion's name, such as y2+1, b6+2, and the value is their corresponding raw intensity. We divide each intensity by the maximum of the intensities within a peptide sequence to normalize each intensity into 0-1. As kinds of ions in the dataset is severely imbalanced, we only select the 8 types of ions same as pdeep2[**?**], that is b(y)i+1-noloss, b(y)i+2-noloss, b(y)i+1-1,H3PO4 and b(y)i+2-1,H3PO4, i indicating the site of b(y)ion, to feed the neural network and predict those 8 types of ion intensity. Those ion intensities are formed as the matrix of shape 8 * length of the sequence (illustrated in the supplementary). There are two types of fragment ion intensity values that have no contribution in the loss calculation and are removed in the prediction. One is that related to padding, like y20 for a 7-mer; the other is related to the phosphorylation site. For example, the phosphorylation site is located in the b5, then the b1,b2, b3, and b4 ions cannot lose phosphate so that the ion intensity with phosphate loss must be 0. This ignorance of impossible phosphorylation site potentially help model to learn the implicitly rule of phosphorylation, benefiting the prediction accuracy of intensity of ion with phosphate loss. Otherwise, the only way for model to learn this rule is from the ion intensity data which would be much more inefficient than the injecting the prior knowledge directly to the model learning. The data split, N-terminal acetyl modification indicator and padding operation are the same as the RT dataset.

Metric

RT

The $\Delta t_{95\%}$ metric is used as the main metric, which represents the minimal time window containing the deviations between observed and predicted RTs for 95% of the peptides.

$$\Delta t_{95\%} = 2 * |y - \hat{y}|_{95\%}$$

The subscript 95% means the 95% rank of the deviations. We select the model by $\Delta t_{95\%}$ metric. Pearson Correlation Coefficient (PCC) is also referred.

Model	$\Delta t_{95\%}$	parameters
DeepRT	15.67	3.1M
LSTM	14.73	1.1M
LSTM+transformer(ours)	14.70	1.5M
transformer	17.00	3M

Table 1. Human Dataset results. Ours is better.

Ion Intensity

We compute each peptide's PCC and select the median of those PCCs as the final evaluation metric. Primarily, we follow Prosit[**?**] using normalized spectral angle(SA) as another metric and the median of those SAs is reported as the final evaluation result. SA's formula is as follows:

$$SA = 1 - 2 * \frac{cos^{-1}(\hat{V_a} \cdot \hat{V_b})}{\pi}$$

$\hat{V}$ is a vector whose L2 norm equals 1. We select the model by the median PCC metric.

4.2. RT experiments

For this task, we compare results of two datasets of our model with DeepRT and different model architecture setting, the performance is shown in table **??**, Figure **??** and table **??**, Figure **??**. We could see that we use the half number of parameters of DeepRT but achieve better performance in two datasets. And our model's prediction could fit the real RT distribution very well.

4.3. Ion Intensity experiments

As we have explore the architecture in the RT task, so that we only compare with the pdeep2. The DDA comparison is shown in Figure **??**, Table **??** and Figure **??**, Table **??**. From the results, we could see that in the metric median PCC and SA, we have beaten the SOTA model pdeep2. In further, to explain our model's good generalization ability, we train our model in the DDA dataset and direct test on the DIA18 dataset. Results are show in the Figure **??** and Table **??**. And we could see results that model trained on the DDA dataset, test on DIA18 dataset are comparable to trained on DIA18, and are even similar to pdeep2's results on DIA18.

Model	$\Delta t_{95\%}$	parameters
DeepRT	19.7	3.1M
LSTM	17.6	1.1M
LSTM+transformer(ours)	17.3	1.5M
transformer	20.8	3M

Table 2. Jeff Dataset results. Ours is better.

Model	Median PCC	Median SA
pdeep2	0.954	0.856
LSTM+transformer$\star$	0.975	0.891

Table 3. DDA Dataset results.Ours is better.

Model	Median PCC	Median SA
pdeep2	0.896	0.773
LSTM+transformer$\star$	0.911	0.796

Table 4. DIA18 Dataset results.Ours is better.

Model	Median PCC	Median SA
Direct Test	0.895	0.781
Train then Test	0.911	0.796

Table 5. DIA18 Dataset results. Direct test only drops little compared to training and test

4.4. ablation study

5. Conclusion

In this study, we introduce Dive2Protein, a flexible deep neural network architecture able to predict retention times and tandem mass spectrometry spectra of peptides and that substantially surpass current benchmarks and tools. Although trained on tryptic peptides from human origin, it performed very well with all proteases, organisms, datasets, mass spectrometers and acquisition parameters tested here. This highlights that the learned internal representation of Dive2Protein approximates a chemo-physical model for peptide fragmentation and chromatographic retention time. However, it is also clear that including more non-tryptic data or longer peptides as well as higher charge states would most probably further improve prediction accuracy.

Our collaborator's results demonstrate that predicted spectral libraries can be used for analyzing DIA data. While predicted libraries performed slightly worse than high-quality experimental spectral libraries, replacing lower quality spectral libraries by consistent and high signal-to-noise predicted spectra increased the number of identified peptides by up to 10%. In the future, Dive2Protein might enable the regeneration of libraries on instrument replacement or calibration and potentially supports the consistent addition of new peptide hypothesis without compromising the homogeneity of a library.