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

Weizhen Liu[1]*, Rongjie Li[1]*, Ronghui Lou[2,3,5], Wenqing Shui[2,3]†, Xuming He[1,4]†
[1]School of Information Science and Technology, ShanghaiTech University
[2]School of Life Science and Technology, ShanghaiTech University
[3]iHuman Institute, ShanghaiTech University
[4]Shanghai Engineering Research Center of Intelligent Vision and Imaging
[5]University of Chinese Academy of Sciences
{liuwzh, lirj2, lourh, shuiwq, hexm}@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 the sequence database are crucial for the search. The research proposes we could build a virtual sequence database composed of the peptide sequence with its retention time and spectra by the computational method. Nowadays, the deep learning model has shown 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 DeepPhospho, based on the LSTM-Transformer model, focusing on the phosphorylation peptide data, especially. By our model, researchers 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 [**?**]. In particular, bottom-up proteomic approaches can greatly benefit from high-quality mass spectra prediction from amino acid sequences [**?**]. 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

*Both authors contributed equally to the work.
†Both are conrresponding authors.

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 also release our code(https://github.com/weizhenFrank/DeepPhospho.git).

2. Related Work

2.1. Retention Time Prediction

For retention time prediction, previous efforts to predict peptide RTs are mainly based on retention coefficients (Rc) of amino acids, while SSRCalc [**?**] is the most popular Rc-based RT 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 GP-Time [**?**] 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. However, these tools could not predict RT quite accurately, that it is well recognized that these measurable factors are far not enough to fully illustrate the physicochemical properties of peptides and the complex interactions between peptides and stationary phase. Hence, in the domain of peptide RT prediction, there is still large room for improvement.

Deep learning, an advanced machine learning method, has shown the extraordinary capability to learn complex relationships from large-scale data. Several tools have successfully utilized deep learning in RT prediction, such as DeepRT [**?**], and Prosit [**?**]. DeepRT uses the capsule network (CapsNet) [**?**] model and could foresee the RTs for the peptides at different modification status included as oxidation of methionine, phosphorylation of serine, threonine, tyrosine, and at varied experiment conditions. Prosit utilizes the LTSM model, and like DeepRT, it could predict the retention time given the peptide sequence. However, the DeepRT did not explain very well the choice of CapsNet, and it did not fully consider the sequence's characteristics.

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

2.2. Ion Intensity Prediction

Investigation of the peptide fragmentation is valuable both in theory and in practice, and there is some research effort about 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. But the kinetic model cannot consistently be used to model the peptide fragmentation under different spectrometry conditions. PeptideART is a pure machine learning based tool that transforms the theoretical spectrum prediction as a classification problem, and it uses a shallow feed-forward neural network [**?**, **?**] to learn the probability of the occurrence of each peak. It has been widely recognized that a suitable spectrum prediction method can boost the identification of peptides. However, peptide fragmentation is very complex to predict, and to handle the complexity of peptide fragmentation, more powerful algorithms such as deep learning could 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 kinds of spectra where it achieves >0.9 median PCCs (Pearson correlation coefficient). Furthermore, its performance is significantly better than kinetic model-based MassAnalyzer and MS-Simulator as well as the machine learning-based PeptideART. After pDeep, the modified version of pDeep, called pDeep2 [**?**] is proposed. pDeep2 is spectrum predictor for modified peptides like phosphorylated peptides based on the deep learning model. It uses the transfer-learning technique to transfer pDeep model parameters on non-modified to the prediction of modified peptide's spectrum.

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.

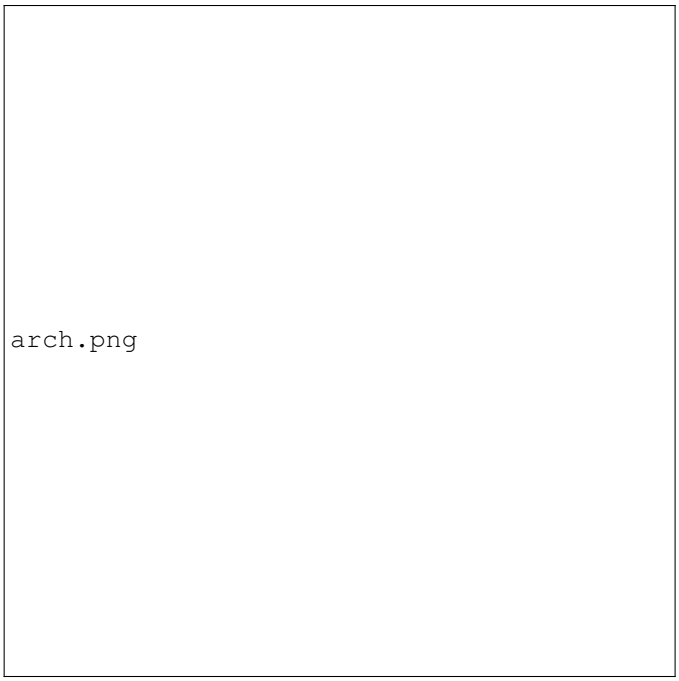

Figure 1. Model architecture

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.

3.2. RT prediction

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.

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.

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. In further, we support the peptide

with N-terminal acetyl modification. We use the * symbol to indicate modification, @ to indicate no modification.

For RT datasets, they are comprised of $\{X, y\}$ pair. $X := \{< x_0, x_1, x_2, x_3, \ldots, x_i, \ldots, x_n >\}$. x_0 is the symbol of * or @. x_i ($i >= 1$) is amino acid. n is the length of peptide. y is the retention time. As the retention time is distributed in the real-world unit, such as minutes or seconds, we scale each dataset by its maximal and minimal of retention time to 0 - 1 by the following formula.

$$RT_{normalized} = \frac{RT - min(RT)}{max(RT) - min(RT)}$$

To train a better model, we sequentially pre-train the models in four datasets called HumanPhosDB [**?**], Jeff [**?**], VeroE6 [**?**] and R2P2 [**?**]. We split those pre-training datasets into training and validation set, selecting the best model on the validation set. The model is initialized by 8the selected model before training on the next pre-training dataset until those four datasets are all trained on. There are three downstream datasets called U2OS-DIA [**?**], RPE1-DIA [**?**] and RPE1-DDA [**?**]. For the three downstream datasets, we manually set the $min(RT)$ and $max(RT)$ equals -100 and 200, respectively. -100 and 200 cold cover all the RTs in the three datasets, and the following researcher could directly use our well-trained model and the fixed $min(RT)$ and $max(RT)$ to predict the unknown RTs of their interested peptides.

Ion intensity datasets are also comprised of $\{X, y\}$ pair. $X := \{< x_0, x_1, x_2, x_3, \ldots, x_i, \ldots, x_n, +q >\}$. $+q$ is the charge carried by the peptide sequence before it is fragmented in the mass spectrometer. 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 train and predict. The shape of ion intensity input is illustrated in the supplementary.

Similarly to the RT task, we also use those ion intensity in three of four pre-training dataset Jeff [**?**], VeroE6 [**?**] and R2P2 [**?**], and fine-tune the pre-trained model on the downstream dataset U2OS-DIA [**?**], RPE1-DIA [**?**] and RPE1-DDA [**?**]. The summary of the datasets used in this work is shown in Table **??**.

Metric For the RT task, the $\Delta t_{95\%}$ metric is used as the primary metric, representing 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. Pearson Correlation Coefficient (PCC) is also referred to, but we select the model by $\Delta t_{95\%}$ metric as the PCC metric could not reflect the difference between different methods as could be seen in the following experimental results.

For the Ion Intensity task, 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. 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

We sequentially train the model on the four pre-training datasets, HumanPhosDB [**?**], Jeff [**?**], VeroE6 [**?**] and R2P2 [**?**], and obtain the best model by the validation sets. Then we fine-tune the model on the three downstream datasets, U2OS-DIA [**?**], RPE1-DIA [**?**] and RPE1-DDA [**?**], respectively. The downstream datasets are split into training : validation : test = 8 : 1 : 1, and we select model on the validation set, reporting the metric on the test set. Correspondingly, we download the DeepRT model provided by the [**?**], and fine-tune the model on those downstream datasets. The detailed result is shown in Table **??**. We could see that ours is the better in all three datasets, which means our method's performance is very stable though these three datasets are quite dissimilar. We improve -4.981%, -7.416% and -6.979% in $\Delta t_{95\%}$, respectively for the three downstream dataset compared to DeepRT.

4.3. Ion Intensity experiments

Similarly to the RT task, we obtain the best pre-trained model from three pre-training datasets, Jeff [**?**], VeroE6 [**?**] and R2P2 [**?**], and fine-tune the model on the downstream three datasets, U2OS-DIA [**?**], RPE1-DIA [**?**] and RPE1-DDA [**?**] with a same split ratio of training, validation, and test. Also, we download the pDeep2 model from [**?**] and fine-tune the pDeep2 on those three downstream datasets. The detailed result is shown in Table **??**. From the table, we could conclude that ours is the better in all three datasets. Although the ion intensity task is relatively more complicated than the RT task, we have significantly boosted ours' performance. We improve 3.060%, 3.987%, and 1.446% in median PCC, and 3.234%, 3.034%, and 2.951% in median SA, respectively, on the three downstream datasets compared to the pDeep2.

data description	no. of peptides	no. of spectra
HumanPhosDB [?]	204,558	-
Jeff [?]	67,552	89,437
VeroE6 [?]	43,405	54,004
R2P2 [?]	35,808	43,312
U2OS-DIA [?]	48,327	58,843
RPE1-DIA [?]	33,576	39,977
RPE1-DDA [?]	129,109	165,719

Table 1. Retention time datasets

method	U2OS-DIA	RPE1-DIA	RPE1-DDA
DeepRT	11.563/0.996	14.752/0.995	16.421/0.983
DeepPhospho	**10.987/0.997**	**13.658/0.996**	**15.275/0.986**

Table 2. RT Dataset results. The left number in cell is $\Delta t_{95\%}$ where the lower is the better, and the right is PCC where the higher is the better.

method	U2OS-DIA	RPE1-DIA	RPE1-DDA
pDeep2	0.887/0.778	0.867/0.767	0.954/0.855
DeepPhospho	**0.915/0.804**	**0.903/0.791**	**0.968/0.881**

Table 3. Ion Intensity Dataset results. The left number in cell is median PCC and the right is median SA where the higher is the better for both metrics.

model	Median PCC	Median SA
w/o LSTM	0.949	0.834
w/o Transformer	0.951	0.835
CNN-Transformer	0.949	0.839
DeepPhospho	**0.955**	**0.844**

Table 4. Ablation study on Jeff dataset

4.4. Ablation study

To illustrate our model design's efficacy, we compare our model architecture with removed LSTM module and removed transformer module. In further, we compare our model with the replacement of LSTM module with convolutional neural network (CNN) module.

The CNN module is built like the ResNet34 [?] except the kernel size of the first convolution layer changed to be 9, and the kernel size in residual block changed to be 7, and it is composed of 3 residual blocks. The comparison experiments are trained on ion intensity data of Jeff [?] dataset. We split the Jeff dataset into training : validation = 9 : 1, reporting the best validation result for each model. The result is seen in Table ??. We could see that once removed any module or replace with CNN module, the performance would decrease.

5. Conclusion

This study introduces DeepPhospho, a flexible deep neural network based on the LSTM-Transformer architecture that can predict retention times and tandem mass spectrometry spectra of peptides. It substantially surpasses current methods in several datasets. We are the first work that introduces the Transformer architecture into peptide chemicophysical properties prediction. Additionally, we design our method target on the phosphorylated peptides in the aspects of additional phosphorylated amino acid symbols, training loss of phosphorylated peptides, and transfer learning of pre-trained phosphorylation peptides datasets.

We believe that a more accurate RT and ion intensity prediction could benefit the downstream proteomics data investigation. For example, the researcher could use our well-trained model to build a predicted spectral library, which could hugely decrease the cost of the wet experiment but no loss of library accuracy. We could also fine-tune the model to adapt to a new type of library where the iteration would be much faster than the previous wet experiment method.