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

1. Prepare Data

1.1. 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.

Ion intensity datasets are comprised of $\{X, y\}$ pair. $X := \{< x_0, x_1, x_2, x_3, \ldots, x_i, \ldots, x_n, +q >\}$. x_0 is the symbol of * or @. $x_i (i >= 1)$ is amino acid. n is the length of peptide. This representation is also used in RT prediction. $+q$ 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 [13], 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.

For RT datasets, they are comprised of $\{X, y\}$ pair. $X := \{< x_0, x_1, x_2, x_3, \ldots, x_i, \ldots, x_n >\}$. 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.

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

2. 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 to solve the retention time (RT) and ion intensity prediction task. The long short-term memory (LSTM) [5] 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 [10] is the second module composed of n Transformer encoder. Each Transformer encoder has 8 attention head, then a feedforward layer configured after attention head.

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

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 [12] 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 [10] 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.

2.1. Ion intensity prediction model

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.

2.2. RT prediction model

For this task, we use model ensemble method and 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.

2.3. Metric

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 [3] 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.

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.

3. Pre-training and transfer learning

3.1. Ion Intensity

For ion intensity prediction, to train a better model, we sequentially pre-train the models in three datasets called Jeff [9], VeroE6 [2] and R2P2 [8] to obtain a good initialization of ion intensity model and fine-tune the pre-trained model on the downstream datasets U2OS-DIA [11], RPE1-DIA [1] and RPE1-DDA [1]. We split those pre-training datasets into training : validation set = 9 : 1, respectively. We tune the hyper-parameters and select the best model on the validation set. The model is initialized by the selected model before training on the next pre-training dataset until those three datasets are all trained on. We select the best model in the pre-traning phase, and fine-tune the model on the downstream datasets.

We use mean squared error (MSE) as our loss function and Adam [6] algorithm to optimize loss with learning rate 1e-3 for the first HumanPhosDB dataset, and 1e-4 for the other datasets. We decay the learning rate by 0.1 once the number of epoch reaches one of the milestones. And the milestones is manually selected by the learning curve during the training.

3.2. RT

For RT prediction, to train a better model, we sequentially pre-train the models in four datasets called HumanPhosDB [7], Jeff [9], VeroE6 [2] and R2P2 [8]. These

pre-training RT dataset use their associated $min(RT)$ and $max(RT)$ to scale the raw RT value to 0-1.

There are three downstream datasets called U2OS-DIA [11], RPE1-DIA [1] and RPE1-DDA [1]. 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. We load the pre-trained model as initialization for downstream dataset training.

Herein we use the square root of mean squared error (RMSE) as loss function and take the same training strategy including the optimizer and learning rate schedule as ion intensity task.

4. Model architecture validation

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 [4] 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. So that there are four models in the ablation study, that is DeepPhospho, LSTM, Transformer and CNN+Transformer.

We do the experiments both on ion intensity dataset and RT dataset. For ion intensity dataset, we use Jeff and R2P2_DDA [8] dataset. For RT, we use R2P2 yeast [8] dataset. We split the dataset into training : validation : test = 8 : 1 : 1, and we tune the hyper-parameter and select the best model in the validation set, reporting the results on the test set.

From the results we could see that, in the test set, our DeepPhospho model achieve median pcc is 0.949 in Jeff, compared to 0.934, 0.901 and 0.937 of LSTM, Transformer and CNN+Transformer. Other datasets also prove that our DeepPhospho's architecture is the best.

Additionally, as Transformer module could capture the long term dependendy better by attention than LSTM [10], we expect the our model has better performance in the longer sequence compared the model based on the LSTM architecture. We select the peptide whose length is more than 40 and could see that the gap between our model and LSTM increase. In the longer sequence, the median pcc of our model is 0.926, but LSTM's is only 0.894. The full comparison results are in supplementary materials. From the result, we could conclude that our model composed of LSTM module and Transformer module is better than any single component, and if we replace the LSTM module with CNN module, we could not obtain the better result.

References

[1] Dorte B Bekker-Jensen, Oliver M Bernhardt, Alexander Hogrebe, Ana Martinez-Val, Lynn Verbeke, Tejas Gandhi, Christian D Kelstrup, Lukas Reiter, and Jesper V Olsen. Rapid and site-specific deep phosphoproteome profiling by data-independent acquisition without the need for spectral libraries. *Nature communications*, 11(1):1–12, 2020. 2, 3

[2] Mehdi Bouhaddou, Danish Memon, Bjoern Meyer, Kris M White, Veronica V Rezelj, Miguel Correa Marrero, Benjamin J Polacco, James E Melnyk, Svenja Ulferts, Robyn M Kaake, et al. The global phosphorylation landscape of sars-cov-2 infection. *Cell*, 182(3):685–712, 2020. 2

[3] Siegfried Gessulat, Tobias Schmidt, Daniel Paul Zolg, Patroklos Samaras, Karsten Schnatbaum, Johannes Zerweck, Tobias Knaute, Julia Rechenberger, Bernard Delanghe, Andreas Huhmer, et al. Prosit: proteome-wide prediction of peptide tandem mass spectra by deep learning. *Nature methods*, 16(6):509–518, 2019. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 3

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 2

[7] Robert T Lawrence, Brian C Searle, Ariadna Llovet, and Judit Villén. Plug-and-play analysis of the human phosphoproteome by targeted high-resolution mass spectrometry. *Nature methods*, 13(5):431–434, 2016. 2

[8] Mario Leutert, Ricard A Rodríguez-Mias, Noelle K Fukuda, and Judit Villén. R2-p2 rapid-robotic phosphoproteomics enables multidimensional cell signaling studies. *Molecular systems biology*, 15(12):e9021, 2019. 2, 3

[9] Jeffrey J Liu, Kirti Sharma, Luca Zangrandi, Chongguang Chen, Sean J Humphrey, Yi-Ting Chiu, Mariana Spetea, Lee-Yuan Liu-Chen, Christoph Schwarzer, and Matthias Mann. In vivo brain gpcr signaling elucidated by phosphoproteomics. *Science*, 360(6395), 2018. 2

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 2, 3

[11] Shisheng Wang, Wenxue Li, Liqiang Hu, Jingqiu Cheng, Hao Yang, and Yansheng Liu. Naguider: performing and prioritizing missing value imputations for consistent bottom-up proteomic analyses. *Nucleic acids research*, 48(14):e83–e83, 2020. 2, 3

[12] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture, 2020. 2

[13] Wen-Feng Zeng, Xie-Xuan Zhou, Wen-Jing Zhou, Hao Chi, Jianfeng Zhan, and Si-Min He. Ms/ms spectrum prediction for modified peptides using pdeep2 trained by transfer learning. *Analytical chemistry*, 91(15):9724–9731, 2019. 1