analytical chemistry

pubs.acs.org/ac Article

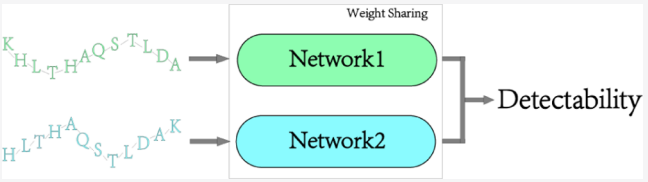
PepFormer: End-to-End Transformer-Based Siamese Network to Predict and Enhance Peptide Detectability Based on Sequence Only

Hao Cheng, Bing Rao, Lei Liu, Lizhen Cui, Guobao Xiao, Ran Su,* and Leyi Wei*

Cite This: *Anal. Chem.* 2021, 93, 6481−6490 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: The detectability of peptides is fundamentally important in shotgun proteomics experiments. At present, there are many computational methods to predict the detectability of peptides based on sequential composition or physicochemical properties, but they all have various shortcomings. Here, we present PepFormer, a novel end-to-end Siamese network coupled with a hybrid architecture of a Transformer and gated recurrent units that is able to predict the peptide detectability based on peptide sequences only. Specially, we, for the first time, use contrastive learning and construct a new loss function for model training, greatly improving the generalization ability of our predictive model. Comparative results demonstrate that our model performs significantly better than state-of-the-art methods on benchmark data sets in two species (*Homo sapiens* and *Mus musculus*). To make the model more interpretable, we further investigate the embedded representations of peptide sequences automatically learnt from our model, and the visualization results indicate that our model can efficiently capture high-latent discriminative information, improving the predictive performance. In addition, our model shows a strong ability of cross-species transfer learning and adaptability, demonstrating that it has great potential in robust prediction of peptides detectability on different species. The source code of our proposed method can be found via https://github.com/WLYLab/PepFormer.



INTRODUCTION

The basic task of proteomics research is the qualitative and quantitative identification of all proteins in a tissue or cell.[1,2] Shotgun proteomics provides an indirect measurement of proteins through peptides derived from the proteolytic digestion of intact proteins. The basic technical route is that proteins are broken down by enzymes into peptide mixtures first, which are analyzed by liquid chromatography-tandem mass spectrometry (LC−MS/MS), and then the database with MS data is retrieved so as to determine the types of proteins and identify hundreds of proteins at the same time.[1,3] Peptide detectability is defined as the probability that a peptide is identified in an LC−MS/MS experiment, so it is useful in providing solutions for protein inference and unlabeled quantification. However, this feature is not considered in most of the bioinformatics pipelines used for proteome analysis. One reason is that there are many variables in the detection of peptides or amino acid sequences by MS, such as the physicochemical properties of peptides, the LC−MS/MS behavior of peptides, and the abundance of their corresponding proteins.[4,5]

To address this problem, several computational methods have been proposed for predicting peptide detectability based on traditional machine learning algorithms and novel deep learning algorithms. For example, Guruceaga et al.[6] developed a Random Forest (RF)-based approach to predict peptide detectability, in which the predictive model is trained based on the physicochemical properties of the peptides stored in the AAIndex resource[7] and other calculable characteristics, such as peptide length and different classes of amino acids. They demonstrate that the RF classifier is the best classification algorithm for the prediction of detectable peptides as compared to their approach with other machine learning algorithms. Li et al.[8] presented a novel iterative algorithm to simultaneously learn standard and effective peptide detectability from complex samples. A total of 292 features were computed solely from peptide sequences and the neighboring residues in proteins from which they were digested as the input and a modular neural network to estimate protein quantities as well as to predict the standard and effective peptide detectability. Beyond that, Zimmer et al.[9] proposed a novel algorithm based on the deep fully connected feed-forward neural network, which allows the informed selection of synthetic prototypic peptides for the successful design of targeted proteomics quantification assays. They used a BioFSharp framework (open-source bioinformatics and computational biology toolbox) to convert a given peptide sequence into a feature vector with 45 entries, which represent a numerical footprint of physicochemical peptide properties as the input of the deep neural network. Later on, owing to the

Received: January 25, 2021
Accepted: March 29, 2021
Published: April 12, 2021

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490

vigorous development of deep learning and the similarity between peptide sequences and natural language, NLP (Natural Language Processing) technology is introduced into the biological sequence model. Guillermo et al.[10] proposed to use the convolution neural network (CNN) to build a predictive method called DeepMSPeptide, where they used the peptide sequence as the input and one-dimensional (1D) convolutional layers to automatically learn and extract sequential features. More recently, Gao et al.[11] considered the significant influence of protein proteolytic digestion on peptide detectability in shotgun proteomics and presented a predictive algorithm namely AP3 based on the RF classifier. In their predictor, they first predicted peptide digestibility and then incorporated it as a new feature with other physicochemical properties to construct a predictive model for peptide detectability prediction.

Although many computational methods have been developed and much progress has been made for the peptide detectability prediction, most existing computational prediction approaches have three shortcomings. First, a majority of current methods highly rely on the physicochemical properties of peptides to construct the predictive model. Due to the lack of direct data on the chemical properties of peptides, 544 physicochemical properties were often calculated for each tryptic peptide from the AAIndex resource. More specifically, the numerical values of the constituent amino acids in a peptide were averaged to produce a single value as the physicochemical property-based feature for peptides. Although this feature representation can fully use the properties of amino acids, in some cases it cannot work well. For example, the peptide sequences "HLTHAQSTLDAK" and "KHLTHAQSTLDA" can be represented with the same features based on physicochemical properties, but actually they have different detectabilities,[12] demonstrating that physicochemical information is not efficient enough. Second, recent methods such as DeepMSPeptide attempt to use NLP techniques (i.e., word2vec) for the prediction of peptide detectability. By analogy with word vector, they considered amino acids as words, and each amino acid is embedded as a fixed feature vector. However, the embedding vector cannot sufficiently capture context-sensitive information, since as for the same amino acid in different peptide sequences the embedding representations are probably different. Third, as the detectability of peptides needs to be verified in experiments, there is no standard measurement to determine if a peptide is detectable or not. Some existing methods trained their models directly based on the MS data of polypeptides. By doing so, it is easy to bring noise into the model training, thus easily generating the evaluation bias and affecting the generalization ability of the model.

To overcome the above shortcomings, we present PepFormer, a novel end-to-end Transformer-based Siamese network to predict peptide detectability. The novelties of the proposed method are concluded as follows.

- We are capable of using peptide sequences only to predict detectability without the need of peptide physicochemical properties as well as other experimental knowledge.
- Our proposed model automatically learns context-sensitive embedding representations from peptides based on the Transformer and gated recurrent unit (GRU) architecture, sufficiently capturing both global and local information to represent the peptide detectability.
- The Siamese network architecture can effectively enhance the representation ability of the embedding features learnt from the Transformer, thus improving the predictive performance.
- We propose a new loss function for model training, which can enhance the generalization ability of our model.

To the best of our knowledge, this is the first attempt to build a unified learning framework with the Siamese network and a Transformer. Comparative studies show that our proposed method outperforms state-of-the-art methods on different data sets, demonstrating the effectiveness of our method in the prediction of peptide detectability.

METHODS AND MATERIALS

Data Sets. Following the same data set collection procedure in the study,[6] we constructed new data sets derived from the GPMDB database,[12] which contains experimentally validated peptide data from a variety of species, including 18 species of eukaryotes, 18 species of prokaryotes, and 17 species of viruses. However, due to the small amount of peptide data for most species, we considered the data samples only from two species: *Homo sapiens* and *Mus musculus*. The format of data downloaded from the GPMDB database is a tab-separated value spreadsheet.

Table 1 shows an example of a peptide sequence in data sets, which contains six columns and one row. As seen in Table 1, the

Table 1. Example of a Peptide Sequence in Data Sets

sequence	z_1	z_2	z_3	z_4	E
DAVLLVFANK	2825	663,899	34,486	28	1.0×10^{-15}

sequence denotes the peptide sequence; z_1−z_4 are the observation numbers of the peptide sequence with a parent ion charge ranging from +1 to +4; and E is the minimum observed E-value for the peptide. Currently, there is no standard threshold to determine whether a peptide sequence is detectable under a certain number of observations. Subsequently, following the same procedure in the study,[6] we ranked the peptide sequences in the data sets according to the observation times and selected the top N data samples as positives and the last N samples as the negatives. Here, we set N = 45,000, which is consistent with the study.[6] In the "Results and Discussion" section, we also discussed how the number of samples affects the predictive performance of the model.

Framework of the Proposed Predictive Method. In this work, we propose a new end-to-end predictive model based on the Siamese network and Transformer architecture. An overview of our model architecture is illustrated in Figure 1. There are four modules in our model, including the sequence preprocessing module, Siamese network module, optimization module, and prediction module. First, in the sequence preprocessing module, we use embedded vectors to represent amino acids of the input peptide sequences. As a result, the peptide sequences can be represented by a feature matrix. Second, the resulting feature matrix is fed into the Siamese network module, generating context-sensitive embedded representations by the Transformer encoder. Afterward, we use Bi-GRU to capture long-distance dependencies of the embedding representations. A FNN (fully connected neural network) is then used to map embedding features to generate unified vector representations. In this way, we can obtain the distance between two peptide sequences. At the same time, the purpose of the model is to predict the detectability of peptides, so a discriminator is needed. In this

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490

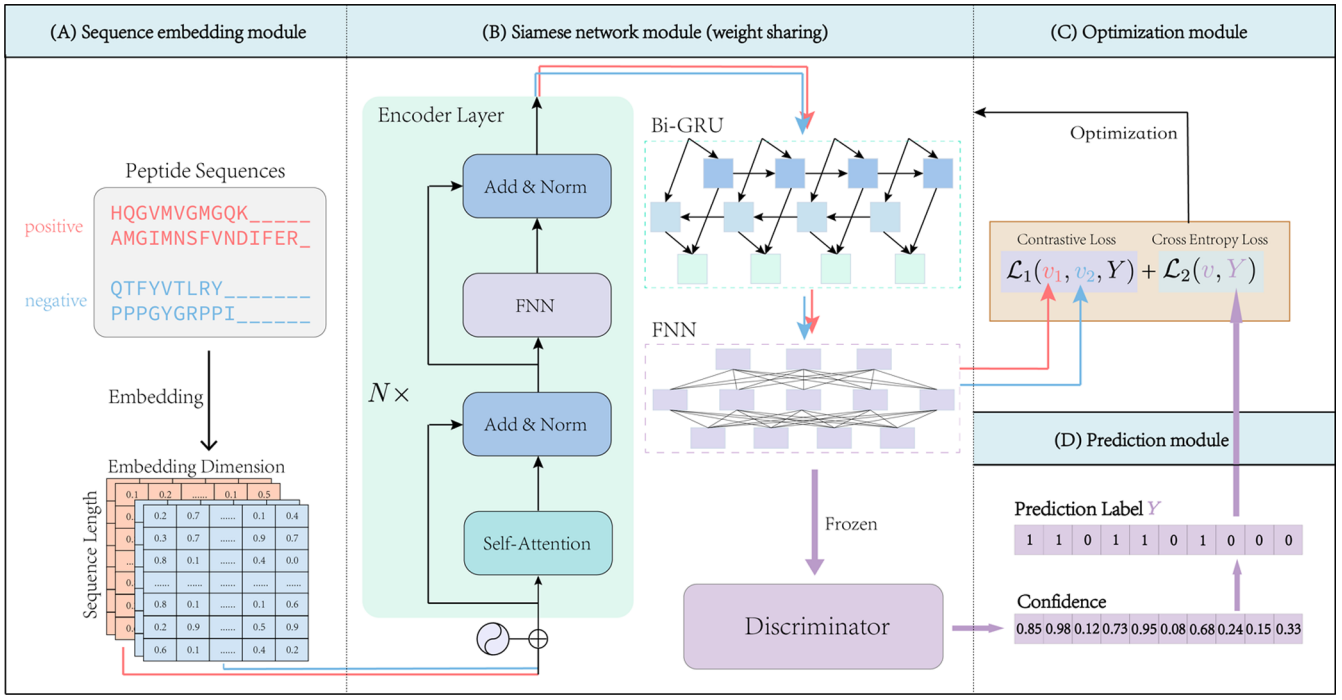


Figure 1. Flowchart of the proposed predictive method—PepFormer. The prediction process can be divided into three parts. (A) The first part is the sequence embedding module, which transforms raw sequence data into embedding feature matrix. (B) Afterward, the resulting matrix is fed into the Siamese network module, which calculates the distance vector and generates the prediction result. (C) The third part is the optimization module. Here, we construct a new loss function to integrate the contrastive loss and the cross-entropy loss from the Siamese network to optimize the predictive performance of the model. (D) Finally, via giving a prediction score, the model can determine if a given peptide is detectable or not. It is detectable if the prediction score is >0.5; otherwise, it is not.

process, we use the Siamese network architecture. In other words, positive samples and negative samples are separately input into the weight-sharing model. Third, in the optimization module, we can calculate the contrastive loss, which is used to make the distance of different representation vectors with the same labels closer and meanwhile to keep the distance of different representation vectors with different labels as far as possible. The cross-entropy loss was computed with the frozen parameters of the previous layers to avoid catastrophic forgetting. The total loss for model optimization is the sum of contrastive loss and cross-entropy loss. Finally, in the prediction module, via giving a prediction score, the model can determine if a given peptide is detectable or not. It is detectable if the prediction score is >0.5; otherwise, it is not.

Sequence Embedding Module. The input of the first encoder layer is a 1D sequence of token embeddings. We first define "token" in peptide sequences as amino acids. Therefore, given a peptide sequence $S = s_1, s_2, ..., s_{|S|}$, where s_i is the ith word and $|S|$ is the length of sequence. Thus, we define a peptide dictionary that maps amino acids to numbers, such as A (alanine) to 1, R (cysteine) to 2, and so on. In this way, we map a peptide sequence to a real-valued vector, which can be denoted as X_i. Notably, because the lengths of peptide sequences are different, we pad real-valued vectors to the same length. We define the maximum length of the peptide sequence as L, so we only consider the length of sequences no more than L. More specifically, given a peptide sequence "LCYVALDFEQEMA-TAASSSSLEK", we convert it to a vector:

[11, 5, 21, 22, 1, 11, 4, 14, 7, 6, 7, 13, 1, 19, 1, 1, 17, 17, 17, 17, 11, 7, 12, 0, ..., 0]

Next, the vector X_i is fed to an embedding layer, transforming into a learnable embedding vector.

Siamese Network Module. *Transformer.* Transformer was proposed by Vaswani et al.[13] for machine translation and has since become the state-of-the-art method in many NLP tasks. In model design, we follow the original Transformer as closely as possible. A Transformer consists of an encoding component and a decoding component. The encoding component is a stack of encoders, which is used to extract sequence features. The decoding component is also a stack of decoders, which is used to generate sequences. Here, we only use a Transformer for feature extraction. The Transformer encoder layer is depicted in Figure 1. An encoding layer consists of a self-attention layer and full connection with the residual connection mechanism.

In order to encode the positional relationship, the Transformer adds a positional vector, which is implemented by sine and cosine functions to each input embedding. In other words, the output $X_{i,\text{embed}}$ of the embedding layer is

$$X_{i,\text{embed}} = \text{Embedding}\,(X_i) + \text{PosEmbedding}\,(X_i) \quad (1)$$

Afterward, $X_{i,\text{embed}}$ is input to a multi-headed self-attention (MSA) block and a MLP block. The layer-normalization (LN) is applied before every block and residual connections after every block. The output of the first encoder layer $X_{i,\text{ouput}}{}^1$ is

$$X_{i,\text{ouput}}{}^1 = \text{MSA}\,(\text{LN}(X_{i,\text{embed}}{}^1)) + X_{i,\text{embed}}{}^1 \quad (2)$$

$$X_{i,\text{ouput}}{}^1 = \text{MLP}\,(\text{LN}(X_{i,\text{ouput}}{}^1)) + X_{i,\text{ouput}}{}^1 \quad (3)$$

Then, $X_{i,\text{output}}{}^1$ is fed into a second encoder layer, generating the output denoted as $X_{i,\text{output}}{}^2$, and so on to the next encoder layers. Suppose there are n encoder layers in the Transformer

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490

encoding component, the output of the last layer is $X_{i,\text{output}}{}^{n}$. Although the shape of $X_{i,\text{output}}{}^{n}$ is the same as $X_{i,\text{embed}}$, the difference between the two is that the latter produces a fixed embedded representation, which is independent of the context of the token, while the former produces a context-dependent embedded representation. The reason for this difference is that the Transformer uses a multi-headed attention mechanism, which can capture not only the local information in a certain size window of the sequence but also the global information of the sequence. In the field of NLP, this method can effectively solve the ambiguity problem of words. In peptide sequences, this mechanism can greatly enrich the representation ability of token and learn context-sensitive embedding representation. In addition, this context-sensitive embedding representation is also helpful for the later GRU model.

Gated Recurrent Unit. A GRU was proposed by Cho et al.[14] to make each recurrent unit to adaptively capture dependencies of different time scales. GRU simplifies long short-term memory (LSTM)[15] a lot but keeps the same effect as LSTM. Similar to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having a separate memory cell. GRU changes the input gate, forgetting gate, and output gate (they are key components of a LSTM cell) into two gates: update gate (z_t) and reset gate (r_t). GRU combines the unit state and output into one state h_t. For each element (time step t) in the input sequence, each layer computes the following function

$$
\begin{aligned}
z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\
\tilde{h}_t &= \tanh(W \cdot [r_t \circ h_{t-1}, x_t]) \\
h &= (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t
\end{aligned} \tag{4}
$$

where W_z, W_r, and W are weights of GRU; h_t is the hidden state at time step t; x_t is the input at time t; $h_{(t-1)}$ is the hidden state of the layer at time $t-1$ or the initial hidden state at time 0; σ is the sigmoid function; and $\circ$ is the Hadamard product. In a multilayer Bi-GRU, the input $x_t^{(l)}$ of the l-th layer ($l \geq 2$) is the hidden state $h_t^{(l-1)}$ of the previous layer. After passing through K layers, we can obtain the final flattened hidden state output h^K ($h^K = [h_1{}^K, h_2{}^K, ..., h_L{}^K]$) of the layer K.

Optimization Module. The Siamese network was first proposed by Lecun et al.,[16] which is originally used to verify the signature on American checks. The purpose of the Siamese network is to measure the similarity of two inputs. More specifically, given two input vectors v_1 and v_2, they are input into the same network in turn. The network will map the inputs to the new space, forming the representation of the input in the new space. Subsequently, the similarity of the two inputs can be evaluated by the distance measure function, denoted as $D(v_1,v_2)$. Here, we use Euclidean distance $D(v_1,v_2)$

$$
D(v_1, v_2) = \|v_1 - v_2\|_2 \tag{5}
$$

After we get the hidden state output h^K, we use a multilayer perceptron to map it to the same representation space. By doing so, we can map an input sequence vector X_i to a representation vector v_i

$$
v_i = \text{MLP}(\text{Bi-GRU}(\text{TransformerEncoder}\ (X_i))) \tag{6}
$$

In order to make the distance of the representation vectors of the same detectability of peptide sequences closer, the distance of the representation vectors with different detectabilities should be as far as possible; we use contrastive loss[17] $\mathcal{L}_1$ as the loss function of the Siamese network

$$
\begin{aligned}
\mathcal{L}_1(v_1, v_2, Y) = &\frac{1}{2}(1 - Y)D(v_1, v_2)^2 \\
&+ \frac{1}{2}Y\{\max(0, m - D(v_1, v_2))^2\}
\end{aligned} \tag{7}
$$

where $Y = 0$ if sequences X_1 and X_2 have the same detectability and $Y = 1$ if they are different. $m > 0$ is a margin. If the distance between v_1 and v_2 is greater than m, no optimization is made; otherwise, the distance between them will be closer to m. In other words, the margin defines a radius around v, and dissimilar pairs contribute to the loss function only if their distance is within this radius.[17]

In this way, we use back propagation algorithm to update network parameters and generate unified pair representation vectors for any pair of inputs. However, we also need a sequence discriminator $D(v)$ to convert the representation vector v to category outputs. We use cross-entropy loss function as the loss of discriminator:

$$
\mathcal{L}_2(v, Y) = -Y \log(D(v)) - (1 - Y)\log(1 - D(v)) \tag{8}
$$

It should be noted that if the multilayer perception and discriminator are connected directly, and the parameters are updated together, the catastrophic forgetting will occur.[18] To avoid this situation, when training the discriminator, we freeze the parameters of the previous layers and only update the parameters of the discriminator. Therefore, in the end, the loss function of our model is

$$
\begin{aligned}
&\mathcal{L}(v_1, v_2, Y, Y_1, Y_2) \\
&\quad = \mathcal{L}_1(v_1, v_2, Y) + \mathcal{L}_2(v_1, Y_1) + \mathcal{L}_2(v_2, Y_2)
\end{aligned} \tag{9}
$$

where Y_1 and Y_2 are the labels of v_1 and v_2 and $Y_1 = 1$ if v_1 is positive, else $Y_1 = 0$.

Experimental Setups. We implemented our model using Pytorch (version 1.6.0), which is an open-source machine learning framework. The data set used in this study consists of 45,000 positive samples and the same number of negative samples, 80% of which is used as the training set and the remaining as the independent test set. The 10-fold cross validation was evaluated on the training set. To avoid the potential evaluation bias generated by cross-validation, our predictive model was determined based on the prediction performance on the test set. The training details of these neural networks are as follows: optimization-ADAM,[19] which is one of the SGD-based algorithms; token vector dimensionality of embedding is 512; and if only one pair of samples are input at a time when training the model, the computational efficiency is too low to make full use of the parallel computing advantage of GPU, so we input one batch samples each time, and the batch size is 128. We use two layers of Transformer encoders and two layers of Bi-GRU. We implemented the discriminator using MLP with the batch-normalization layer[20] with Leaky ReLU[21] as the activation function. In order to avoid overfitting, we adopted the early-stop strategy.

Evaluation Metrics. In this study, we use four metrics, namely, accuracy (ACC), specificity (SP), sensitivity (SN), and Matthews correlation coefficient (MCC), to evaluate our model. The calculation formulas are as follows:

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490

Table 2. Performance of State-of-the-Art Algorithms for Peptide Detectability Prediction

	H. sapiens				*M. musculus*			
models	ACC	SP	SN	MCC	ACC	SP	SN	MCC
DNN	0.5408	0.5514	0.5303	0.0817	0.5333	0.5370	0.5296	0.0666
logistic regression	0.5466	0.4762	0.6167	0.0938	0.5500	0.4903	0.6100	0.1011
AP3	0.6416	0.5949	0.6881	0.2843	0.6462	0.5993	0.6928	0.2934
RF	0.6743	0.5873	0.7610	0.3537	0.6218	0.5047	0.7398	0.2514
DeepMSPeptide	0.7854	0.6875	0.8827	0.5816	0.7228	0.6244	0.8220	0.4552
CNN + LSTM	0.7910	0.7026	0.8790	0.5902	0.7415	0.6062	0.8779	0.5027
PepFormer (this study)	**0.8066**	**0.7213**	**0.8915**	**0.6221**	**0.7521**	**0.6421**	**0.8629**	**0.5176**

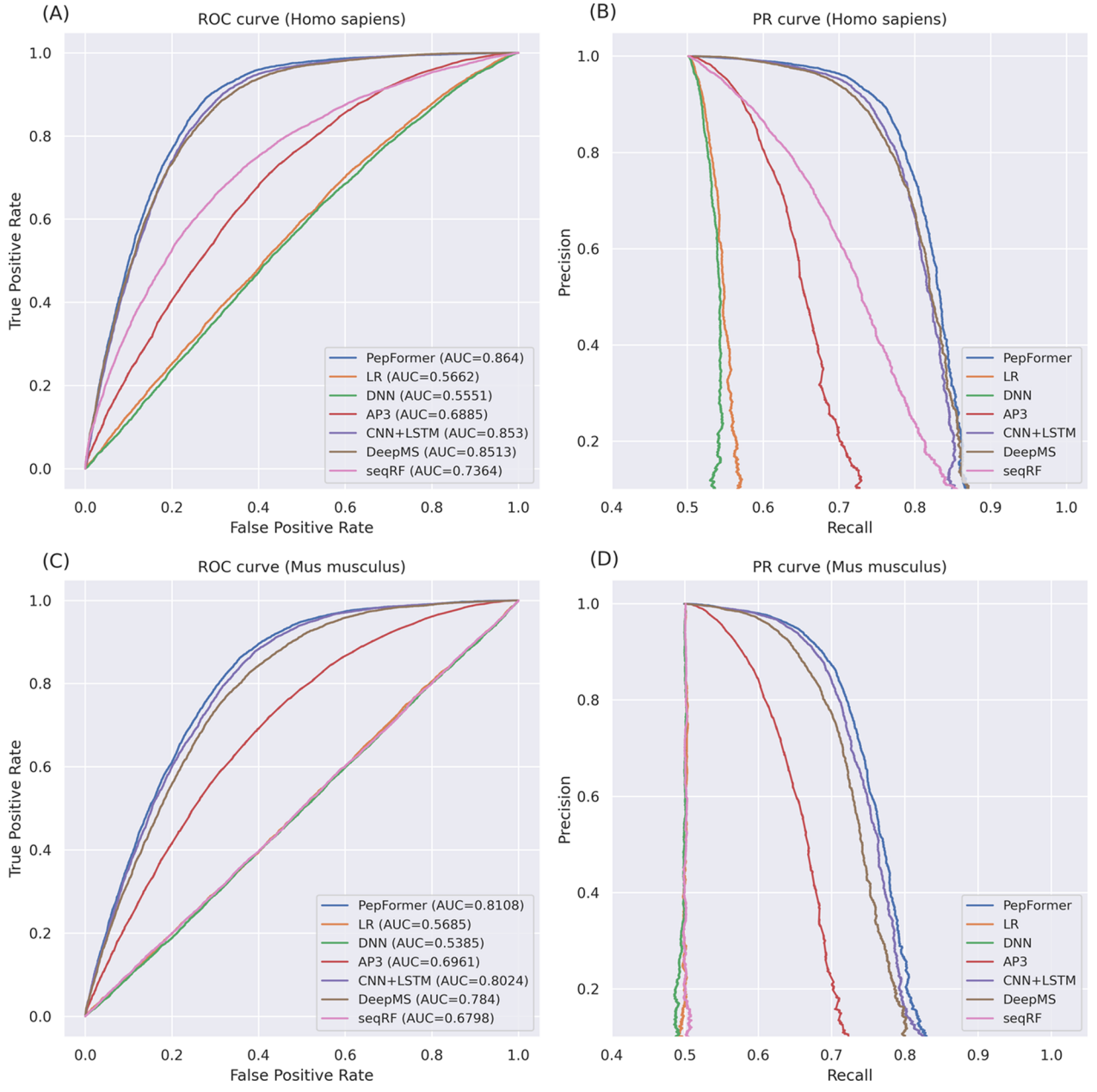


Figure 2. ROC curves and PR curves of our proposed method and existing prediction methods in *Homo* and *Mus* data sets. (A) ROC curves in the *Homo* data set. (B) PR curves in the *Homo* data set. (C) ROC curves in the *Mus* data set. (D) PR curves in the *Mus* data set.

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490

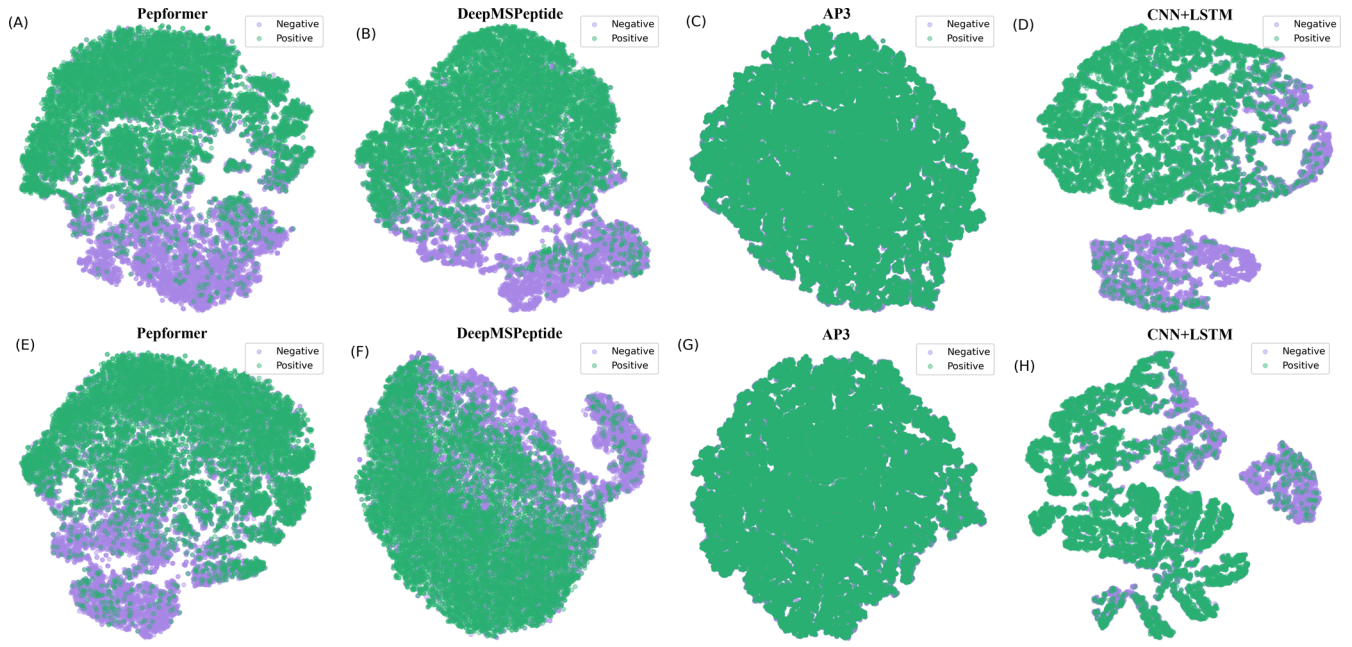


Figure 3. T-SNE visualization of the feature space. (A−D) Visualization results of PepFormer, DeepMSPeptide, AP3, and CNN + LSTM on the *H. sapiens* data set, respectively. (E−H) Visualization results of PepFormer, DeepMSPeptide, AP3, and CNN + LSTM on the *M. musculus* data set, respectively.

$$\begin{cases} \text{accuracy} = \dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \\ \text{specificity} = \dfrac{\text{TN}}{\text{TN} + \text{FP}} \\ \text{sensitivity} = \dfrac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{MCC} = \dfrac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \end{cases} \tag{10}$$

where TP, TN, FP, and FN are true positives, true negatives, false positives, and false negatives, respectively. Specificity measures the proportion of the negatives that are correctly identified, while sensitivity measures the proportion of the positives that are correctly identified. Accuracy is the most intuitive performance measure and it is simply a ratio of the correctly predicted samples to the total samples. AUC represents the probability that a random positive is positioned to the right of a random negative. It is calculated from the area under the ROC (receiver operating characteristic) curve, ranging from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0; one whose predictions are 100% correct has an AUC of 1. MCC, taking into account positives and negatives, is generally regarded as a balanced metric to measure the overall performance of a model. Besides, we also use KL divergence (Kullback−Leibler divergence, also called relative entropy, KLD) to measure the difference between the two distributions.[23] For discrete probability distributions P and Q defined on the same probability space $\mathcal{X}$, the KL divergence from Q to P is defined to be

$$\text{KLD} = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{11}$$

In particular, we investigated the peptide length preference of the proposed model using the KL divergence (see Figure S1 in the Supporting Information for details).

RESULTS AND DISCUSSION

Comparison with Existing Methods. In order to evaluate the effectiveness of our model, we compared our model with state-of-the-art machine learning-based methods that are designed specifically for the peptide detectability prediction:

- DeepMSPeptide:[10] A sequence-based deep learning model, which uses a two-layer 1D convolution to extract features from the peptide sequence.
- AP3:[11] A RF model based on physicochemical features. As we cannot access the peptide digestibility data, here we just used a variant of the AP3 algorithm, in which we used only the physicochemical features from the AAindex and performed the same feature selection method—minimum redundancy maximum relevance (mRMR)[22] to optimize the feature space.
- Deep Natural Network (DNN):[9] It is a FNN that uses the physicochemical features of peptides to train the predictive model.
- CNN + LSTM model: This is a commonly used deep learning architecture, which is a hybrid network of CNN and LSTM. 1D convolution is used to extract sequence features, which are then input into LSTM to capture the long-distance dependence. A discriminator is used for classification.
- Traditional machine learning methods: In addition to the methods mentioned above, we have also implemented other traditional machine learning methods, including Logistic Regression and RF.

It is worth noting that the source codes for all the above methods are not available. For fair comparison, we attempt to implement the existing methods based on the algorithmic detail in their studies. The implementation environment is using

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490

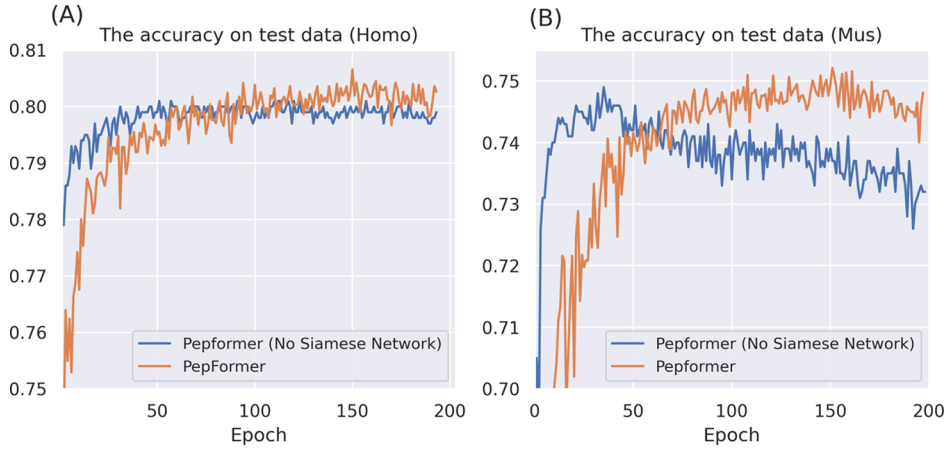


Figure 4. Comparison of the proposed PepFormer trained with and without the Siamese network. (A) Accuracy of the models on the *Homo* test set during training. (B) Accuracy curve of the models on the *Mus* test set during training.

Python (version 3.7). Table 2 presents the results of our proposed PepFormer and the state-of-the-art methods on two benchmark data sets from *H. sapiens* and *M. musculus*. As seen in Table 2, PepFromer remarkably outperforms all other methods on different data sets. As for the *H. sapiens* data set, our PepFromer achieves the highest ACC (0.8066), SP (0.7213), SN (0.8915), and MCC (0.6221), which are higher than those of the runner-up method (CNN + LSTM) by 2, 2.7, 1.4, and 5.4%, respectively. For the *M. musculus* data set, similar results are observed; that is, our method performs better than the existing methods by 1.4−41% and 3−677% in terms of ACC and MCC. Figure 2 illustrates the comparison results based on the ROC curves and PR (Precision and Recall) curves of different methods. As seen, the AUCs of our model for the two test data sets (*H. sapiens* and *M. musculus*) are 0.864 and 0.8108, respectively. Compared to AP3, DeepMSPeptide, and CNN + LSTM, PepFormer increases the AUC by 25.5% (16.5%), 1.5% (3.4%), and 1.3% (1.0%) on average on the *Homo* data set (or *Mus* data set), respectively.

Notably, DNN and AP3 are both based on the physicochemical properties of peptides to train predictive models. As compared to the two methods using physicochemical information, our results demonstrated that peptide sequences themselves contain sufficient information that can distinguish detectable peptides from nondetectable ones. Moreover, it can be seen from Table 2 and Figure 2 that sequence-based deep learning methods (DeepMSPeptide, CNN + LSTM, and our proposed PepFormer) are generally better than the methods based on physicochemical properties, further demonstrating that sequential information might be more discriminative than physicochemical information. More importantly, the deep learning approaches can automatically learn and extract high-latent information, rather than relying on the designing of handcrafted features based on researchers' experience coupled with extra experimental prior knowledge, such as physicochemical information. In conclusion, these results demonstrate that our model retains a better predictive power than the existing methods for the peptide detection prediction.

Visualization Comparison of Embedding Representations of Different Methods. In order to provide more insights regarding the superiority of our method, we present visualizations of feature space of our method and other three outstanding existing methods, including DeepMSPeptide, AP3, and CNN + LSTM. Figure 3A−D illustrates the visualization results on the *H. sapiens* data set, while Figure 3E−H shows the results on the *M. musculus* data set. It should be noted that each point in the figures represents a peptide sequence. The coordinates of the points are determined by T-SNE of the peptide sequence embedding. Different labels are marked by different colors; that is, green and purple represent positive and negative samples, respectively.

As can be seen from Figure 3C, the positive and negative samples in the feature space of the AP3 model are distributed almost together. We cannot see the negative samples, which suggests that the features between the positives and negatives are very similar and shows that physicochemical information is not a distinguishable feature. Figure 3B,D shows the embedding representations of the peptide sequence on DeepMSPeptide and the CNN + LSTM model, respectively. We can see that the positives and negatives have a certain degree of discrimination as compared with the result shown in Figure 3C. One can note that the data samples of different classes are distributed more clearly and compactly in our sequence embedding representation space as compared to other methods. Similarly, the same result can be observed on the *Mus* data set, as shown in Figure 3E−H. The results demonstrate that the superiority of our model is mainly due to the stronger ability of discriminative information capturing. Furthermore, it also increases the interpretability of our model from the perspective of embedding vectors.

Siamese Network Improves the Performance and Generalization Ability of the Model. To see how effective the Siamese network is, we compared the models trained with and without the Siamese network. Figure 4 shows the accuracy curves on the test data with the training epochs. Figure 4A,B represents the test results on two species: *Homo* and *Mus*, respectively. As shown in Figure 4A, at the beginning, the accuracy of PepFormer (without the Siamese network) is higher, but after about 100 epochs, the accuracy of Pepformer is higher than the model without using the Siamese network. A similar trend appears in Figure 4B, after about 60 epochs, the accuracy of PepFormer (without the Siamese network) declines slowly, but the accuracy of Pepformer is increasing. The result demonstrates that using the Siamese network can enhance the generalization ability of the model in different species. Moreover, the model trained with the Siamese network effectively improves the predictive performance as compared to that without using the Siamese network. To this end, it

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490

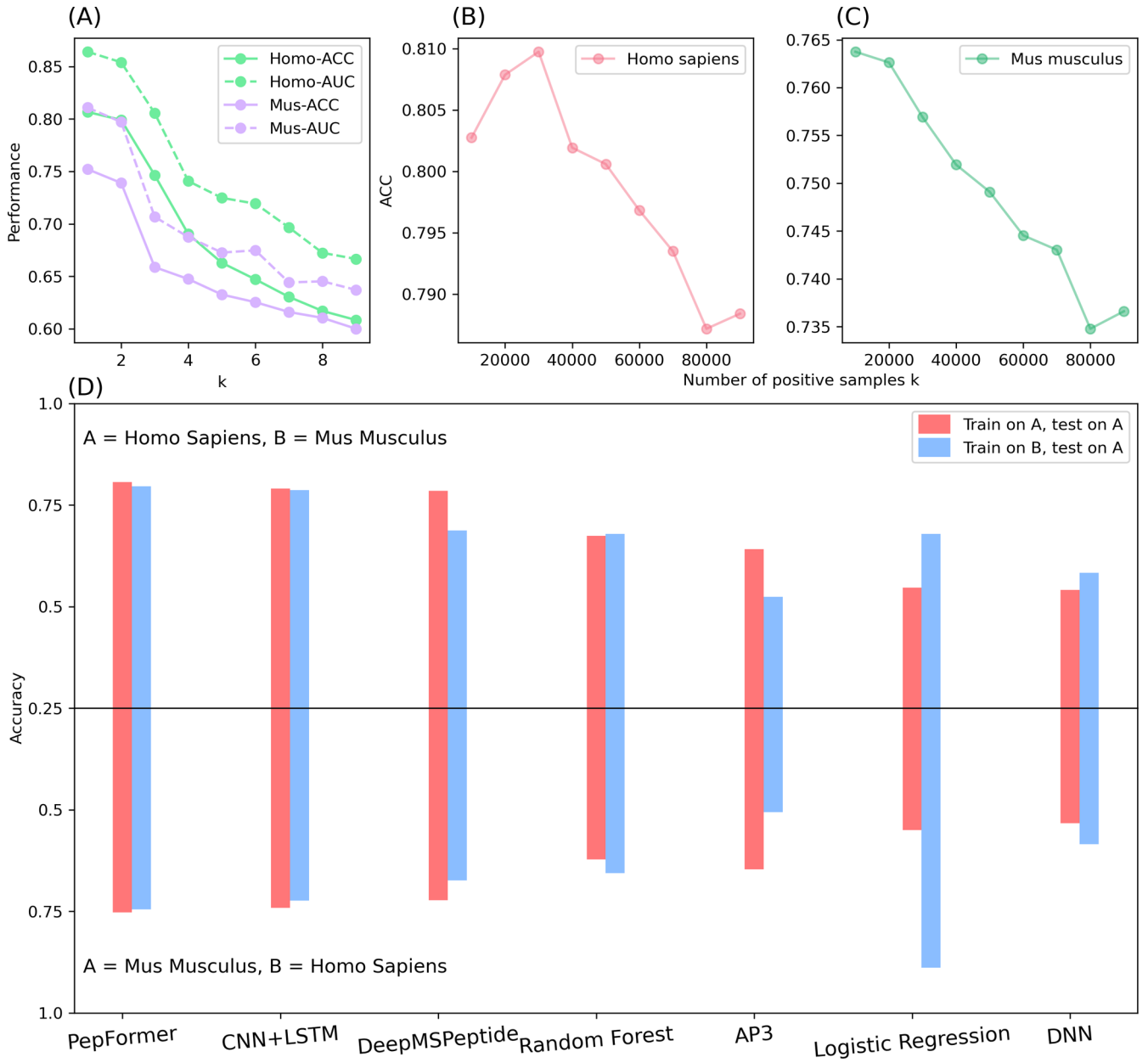


Figure 5. Performance of our predictive model varied with different parameter settings. (A) Relationship between model performance and *k*-mers on different data sets. (B) Relationship between the accuracy and the number of positive samples *k* in the *H. sapiens* data set. (C) Relationship between the accuracy and the number of positive samples *k* in the *M. musculus* data set. (D) Accuracy of different models trained in one species and tested on the other.

provides us a better solution to improve the performance and generalization ability of the model.

Influence of *k*-mers on Model Performance. In our word embedding *k*-mer is a hyperparameter of our model, representing the subsequence of length *k* contained within a peptide sequence. It plays an important role in sequence representation. In order to study the influence of the parameter *k* on model performance, we let $k = 1$ to 9 and generate embedding for each *k*-mer. The results are shown in Figure 5A. As shown in the figure, the performance of the model decreases with the increase of *k* in two data sets. To be specific, with the increase of *k*, the accuracy decreases about 0.05−0.1. The possible reason is that with the increase of *k*, although the ability to capture the global information is enhanced, the ability to exploit local information is lost to some extent, which leads to the decline in performance. Therefore, to make full use of the local information in the data, we set $k = 1$ in our model, which is a reasonable value.

Comparing Prediction Results of Different Thresholds for Data Set Construction. Since we only have the data of the number of peptide sequences detected by MS, with the increase of time, the number of peptides detected increases, so it is unreasonable to divide the detectability of peptides according to a fixed value. However, because the size of the peptide data set is basically fixed, the data set can be divided according to a certain proportion, and the part with higher detection times is called detectable, and the part with lower detection times is called undetectable, so a more reasonable data generation method is given. Since this ratio is also considered to be specified, experiments are needed to verify the influence of different threshold values on the experimental results. We sorted the

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490

peptides according to the detection times of the data set and selected the top k data as positive samples and the last k data as negative samples. The intuitive experimental results are shown in Figure 5B,C. We can see that the performance of the model decreases with the increase of k in two data sets. Since the detection times of peptides in the data set vary with time, the low detection times of peptides in the data set do not mean that they cannot be detected. With the increase of sampling data, the performance of the model gradually declines. The possible reason is that the noise data in the data are also increasing, thus impacting the model performance. However, we can also notice that the results are species-specific, which are slightly different in different data sets. In particular, there is a point higher than the initial point performance in the *Homo* data set, while the accuracy of the initial point is the highest in the *Mus* data set. This shows that a parameter selection process is still needed for a specific data set, and the best performance can be obtained by adjusting the k value.

Evaluations of Cross-Species Transfer Learning Ability. To investigate the cross-species transfer learning ability of the proposed model, we trained our model on one species data set and tested the learning ability of the model on the other species. For comparison, we used the results of our model shown in Table 2 as a benchmark, which is trained and tested on the same species data set. Moreover, following the same procedure, we also evaluated and compared the cross-species prediction ability of existing methods. We illustrated the evaluation performance of different models as a two-sided bar chart shown in Figure 5D, in which horizontal axis coordinates represent various models and vertical axis coordinates represent the accuracy of the models. Notably, for convenience of discussion, we use A and B to denote the data sets from different species, respectively. Above the horizontal axis, A represents the *H. sapiens* data set and B represents the *M. musculus* data set. Below the horizontal axis, the opposite is true.

As can be seen in Figure 5D, when training our model using the Homo data set and evaluating the performance on the test set from the same species, our PepFormer achieved the best results; while training our model using the *Mus* data set and evaluating on the same Homo test set, our performance declined a little bit. This demonstrates that our Pepformer shows very strong cross-species transfer learning ability. Even using different species for model training, our model still can achieve excellent performance. Interestingly, we found that other existing methods, including CNN + LSTM, DeepMSPeptide, and AP3, show the same conclusion with our method. However, the accuracies derived from existing methods are lower than that of our model. The results further confirm that our method is more effective for the peptide detectability prediction, even from the cross-species point of view. On the other hand, it also demonstrates that the models have higher complexity and stronger learning ability and can learn unique characteristics of the data set. Moreover, we also observed that as for RF, logistic regression, and DNN, the accuracy of cross-species transfer learning is better (the blue bar) than those of the models without using cross-species transfer learning. One potential reason is that the learning ability of the model is weak, and it cannot fully learn the unique characteristics of the data set in one species, which makes the effect of the model on the data set of other species slightly better. Finally, despite this, our models achieved excellent performance among the compared methods, which also demonstrates that our model has a strong ability of cross-species transfer learning.

CONCLUSIONS

In this study, we have presented a Transformer-based Siamese network architecture for predicting peptide detectability. For the first time, we use a Transformer to encode the context embedding of the peptide sequence. Our results demonstrate that the Siamese network architecture introduced to the predictive model can effectively increase the accuracy and the generalization ability of the peptide detectability prediction. Moreover, comparing with state-of-the-art methods, our PepFormer exhibits a better overall performance tested on different species, suggesting that our method has potential to be a useful tool for the prediction of peptide detectability. More importantly, given peptide sequences only, we demonstrate that our proposed end-to-end network architecture can automatically learn and explore the discriminative information from the sequences, without any need of prior knowledge and the help of handcrafted feature engineering. Finally, this study may also have a significant effect on improving protein quantification and discovering biological biomarkers for early diagnosis and therapy.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.analchem.1c00354.

Peptide length preference of the proposed model, length distribution of the data set, and model prediction results (PDF)

AUTHOR INFORMATION

Corresponding Authors

Ran Su − *College of Intelligence and Computing, Tianjin University, Tianjin 300384, China*; Email: ran.su@tju.edu.cn

Leyi Wei − *School of Software and Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan 250101, China; Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, College of Computer and Control Engineering, Minjiang University, Fuzhou 350000, China*; orcid.org/0000-0003-1444-190X; Email: weileyi@sdu.edu.cn

Authors

Hao Cheng − *School of Software and Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan 250101, China*

Bing Rao − *School of Mechanical Electronic & Information Engineering, China University of Mining & Technology, Beijing 221008, China*

Lei Liu − *School of Software and Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan 250101, China*

Lizhen Cui − *School of Software and Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan 250101, China*

Guobao Xiao − *Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, College of Computer and Control Engineering, Minjiang University, Fuzhou 350000, China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.analchem.1c00354

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China (nos. 62072329 and 62071278).

■ REFERENCES

(1) Aebersold, R.; Mann, M. *Nature* **2003**, *422*, 198.
(2) He, F. *Mol. Cell. Proteomics* **2005**, *4*, 1841.
(3) McDonald, W. H.; Yates, J. R. *Curr. Opin. Mol. Therapeut.* **2003**, *5*, 302.
(4) Jarnuczak, A. F.; Lee, D. C. H.; Lawless, C.; Holman, S. W.; Eyers, C. E.; Hubbard, S. J. *J. Proteome Res.* **2016**, *15*, 2945.
(5) Tang, H.; Arnold, R. J.; Alves, P.; Xun, Z.; Clemmer, D. E.; Novotny, M. V.; Reilly, J. P.; Radivojac, P. *Bioinformatics* **2006**, *22*, No. e481.
(6) Guruceaga, E.; Garin-Muga, A.; Prieto, G.; Bejarano, B.; Marcilla, M.; Marín-Vicente, C.; Perez-Riverol, Y.; Casal, J. I.; Vizcaíno, J. A.; Corrales, F. J.; Segura, V. *J. Proteome Res.* **2017**, *16*, 4374.
(7) Kawashima, S.; Pokarowski, P.; Pokarowska, M.; Kolinski, A.; Katayama, T.; Kanehisa, M. *Nucleic Acids Res* **2007**, *36*, D202.
(8) Li, Y. F.; Arnold, R. J.; Tang, H.; Radivojac, P. *J. Proteome Res.* **2010**, *9*, 6288.
(9) Zimmer, D.; Schneider, K.; Sommer, F.; Schroda, M.; Mühlhaus, T. *Front. Plant Sci.* **2018**, *9*, 1559.
(10) Serrano, G.; Guruceaga, E.; Segura, V. *Bioinformatics* **2019**, *39*, 1279.
(11) Gao, Z.; Chang, C.; Yang, J.; Zhu, Y.; Fu, Y. *Anal. Chem.* **2019**, *91*, 8705.
(12) Craig, R.; Cortens, J. P.; Beavis, R. C. *J. Proteome Res.* **2004**, *3*, 1234.
(13) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 2017.
(14) Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *EMNLP 2014−2014 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics, 2014.
(15) Hochreiter, S.; Schmidhuber, J. *Neural Comput.* **1997**, *9*, 1735.
(16) Bromley, J.; Bentz, J. W.; Bottou, L.; Guyon, I.; Lecun, Y.; Moore, C.; Säckinger, E.; Shah, R. *Int. J. Pattern Recogn. Artif. Intell.* **1993**, *07*, 669.
(17) Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality Reduction by Learning an Invariant Mapping. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; IEEE, 2006.
(18) McCloskey, M.; Cohen, N. J. *Psychol. Learn. Motiv.* **1989**, *24*, 109.
(19) Kingma, D. P.; Ba, J. L. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations ICLR 2015—Conference Track Proceedings*; ICLR, 2015.
(20) Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *32nd International Conference on Machine Learning*; ICML, 2015.
(21) Maas, A. L.; Hannun, A. Y.; Ng, A. Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models. *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*; ICML, 2013.
(22) Ding, C.; Peng, H. *J. Bioinf. Comput. Biol.* **2005**, *03*, 185.
(23) Kullback, S.; Leibler, R. A. *Ann. Math. Stat.* **1951**, *22*, 79.

https://doi.org/10.1021/acs.analchem.1c00354
Anal. Chem. 2021, 93, 6481−6490