Bioinformatics, 38(6), 2022, 1514–1524
https://doi.org/10.1093/bioinformatics/btac006
Advance Access Publication Date: 6 January 2022
Original Paper

Sequence analysis

ToxIBTL: prediction of peptide toxicity based on information bottleneck and transfer learning

Lesong Wei [1], Xiucai Ye[1,*], Tetsuya Sakurai[1], Zengchao Mu[2] and Leyi Wei[3,*]

[1]Department of Computer Science, University of Tsukuba, Tsukuba 3058577, Japan, [2]School of Mathematics and Statistics, Shandong University, Weihai, China and [3]School of Software, Shandong University, Jinan, China

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on October 29, 2021; revised on November 29, 2021; editorial decision on December 9, 2021; accepted on January 4, 2022

Abstract

Motivation: Recently, peptides have emerged as a promising class of pharmaceuticals for various diseases treatment poised between traditional small molecule drugs and therapeutic proteins. However, one of the key bottlenecks preventing them from therapeutic peptides is their toxicity toward human cells, and few available algorithms for predicting toxicity are specially designed for short-length peptides.

Results: We present ToxIBTL, a novel deep learning framework by utilizing the information bottleneck principle and transfer learning to predict the toxicity of peptides as well as proteins. Specifically, we use evolutionary information and physicochemical properties of peptide sequences and integrate the information bottleneck principle into a feature representation learning scheme, by which relevant information is retained and the redundant information is minimized in the obtained features. Moreover, transfer learning is introduced to transfer the common knowledge contained in proteins to peptides, which aims to improve the feature representation capability. Extensive experimental results demonstrate that ToxIBTL not only achieves a higher prediction performance than state-of-the-art methods on the peptide dataset, but also has a competitive performance on the protein dataset. Furthermore, a user-friendly online web server is established as the implementation of the proposed ToxIBTL.

Availability and implementation: The proposed ToxIBTL and data can be freely accessible at http://server.wei-group.net/ToxIBTL. Our source code is available at https://github.com/WLYLab/ToxIBTL.

Contact: yexiucai@cs.tsukuba.ac.jp or weileyi@sdu.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The therapeutic potential of peptide-based drugs has gained unprecedented interest during the past decade and their development is both strong and rapid. In general, peptides are defined as short chains with 50 or fewer amino acids, and it is one of the biomolecules that determine and control biological functions, thereby playing a crucial role in treating various pathological conditions (Fosgerau and Hoffmann, 2015; Gohil and Thirugnanasambandan, 2021). When compared with small molecules, peptides offer higher biological activity and selectivity, which result in relatively fewer side effects. Simultaneously, compared with proteins, peptides have smaller sizes and are more accessible and cheaper to manufacture using chemical methods (Craik *et al.*, 2013; Haggag *et al.*, 2018). Therefore, these characteristics make peptides evolve as promising therapeutic agents. Up to now, more than 80 peptide-based drugs have been approved in the market for treating a variety of diseases, including diabetes, cancer, chronic pain, osteoporosis, infectious diseases and multiple sclerosis (Muttenthaler *et al.*, 2021).

The importance of peptides as key biological mediators, along with their remarkable potency, selectivity and moderate production costs, was established (Craik *et al.*, 2013; Muttenthaler *et al.*, 2021). However, peptides as a type of bio drugs have several intrinsic weaknesses, which prevent them from being directly used as convenient therapeutics, such as toxicity, immunogenicity and stability (Gupta *et al.*, 2013, 2015). Great efforts have been made to overcome the limitations on the immunogenicity and stability (Ansari and Raghava, 2010; Chen *et al.*, 2012; El-Manzalawy *et al.*, 2008; Gentilucci *et al.*, 2010; Saha and Raghava, 2006). However, for the toxicity of peptides, to date, there are fewer proposed methods considering this limitation, which is essential for facilitating their therapeutic application.

The most direct and effective way to identify the toxicity of a peptide is through biological experiments in a wet lab, which are expensive, labor-intensive and time-consuming, particularly with the great growth of potential therapeutic peptides. Moreover, the results obtained from animal trials generally offer little guidance to human toxicity reactions (Mumtaz and Pohl, 2012). Instead of the biological experiments, computation-assisted identification methods have

been proposed to analyze the toxicity of peptides. These methods can be roughly divided into two categories: similarity-based methods and machine learning-based methods. The similarity-based methods use alignment search tools to measure the local and global pairwise sequence similarity, such as BLAST (Altschul *et al.*, 1997), or infer sequence toxicity from homologous ones (Negi *et al.*, 2017). However, there are several drawbacks in such methods. First, peptides of interest need to have homologous toxic ones. Second, the performance will drop dramatically when processing large amounts of data. Third, *e*-value cutoff and an arbitrary sequence similarity need to be set, which can affect prediction performance.

In contrast to the similarity-based methods, the machine learning-based methods focus on capturing discriminative information related to toxicity, using both positive and negative samples, to predict the peptide toxicity (Manavalan *et al.*, 2019a, b; Su, *et al.*, 2020). For example, ClanTox employs the 545-dimension features derived from primary sequences as inputs, and trains a predictor based on boosted stump classifiers for analyzing animal toxins (Naamati *et al.*, 2009). ToxinPred combines a support vector machine (SVM) and various statistical features of peptide sequences to discriminate toxic peptides from non-toxic ones (Gupta *et al.*, 2013). These two methods have made great contributions to the development of peptide toxicity prediction. However, capturing discriminative features to represent the inherent characteristics of peptides is the key to construct a powerful computational predictor, which is still the challenge for the performance improvement of machine learning-based methods (Li and Liu, 2020; Wei *et al.*, 2021). Additionally, in the process of extracting features, these two methods do not consider the sequence-order information and the position dependency in a sequence, which are critical for amino acid sequences analysis and nucleic acid sequences analysis (Li *et al.*, 2017; Liu *et al.*, 2015). Therefore, it is highly desirable to incorporate them into computational methods to extract more discriminative features for amino acid sequences.

Recently, compared with other machine learning methods, deep learning methods have achieved remarkable performance in the field of bioinformatics (Ye *et al.*, 2020), such as drug-target interaction prediction (Chu *et al.*, 2021; Zeng *et al.*, 2020), RNA secondary structure prediction (Sato *et al.*, 2021; Singh *et al.*, 2021) and protein fold recognition (Li and Liu, 2020; Liu *et al.*, 2020). In our previous work (Wei *et al.*, 2021), we proposed a deep learning-based method, called ATSE, which used structural and evolutionary information based on graph neural networks and the attention mechanism for peptide toxicity prediction and achieved high prediction accuracy (ACC). However, in this method, position-specific scoring matrixes (PSSMs) containing evolutionary information need to be searched by using PSI-BLAST in a big database (Altschul *et al.*, 1997), which is time-consuming. Searching PSSMs in different databases with different sizes will obtain different results, which is also a limitation of ATSE. Moreover, Pan *et al.* (2020) propose ToxDL, a deep learning-based model that is effective to classify toxic and non-toxic proteins with diverse lengths by employing both sequence information and protein domain knowledge. However, this method is not specially designed for peptides and needs to search protein domains of a given protein for embeddings.

Despite the great progress made so far, there is still a need to develop a more accurate method for peptide toxicity prediction to reduce the number of misclassified samples and thus improve the confidence of the predicted toxic peptides. To overcome the aforementioned shortcomings, we propose, in this study, a novel deep learning-based model, called ToxIBTL, to effectively predict the toxicity of both peptides and proteins by adopting both information bottleneck principle and transfer learning technique. We make several significant contributions which can be summarized as follows:

1. Our model automatically learns the latent evolutionary information embedding in the BLOSUM62 (BLOcks SUbstitution Matrix) matrix of a peptide (or protein) by using a hybrid network CNN_BiGRU composing of convolutional neural network (CNN) and bidirectional gated recurrent unit (BiGRU), which can sufficiently capture both local and long-distance correlations in a sequence. Simultaneously, we use FEGS (Feature Extraction based on Graphical and Statistical features; Mu *et al.*, 2021) model to generate physicochemical features for a sequence. These features are employed together to represent a peptide (or protein) sequence.
2. We adopt the information bottleneck principle to supervise the feature learning for finding the better concise latent representation from the obtained features, which tends to retain as much as relevant information for predicting the label while removing noisy information.
3. To address the problem of peptide data scarcity, transfer learning is used to transfer common toxic information learned from the large protein dataset to the small peptide dataset, thereby improving the prediction performance.
4. Our model can predict the toxicity of peptides with short lengths as well as proteins with long lengths.

Comparative experimental results show that our proposed ToxIBTL leads to a new state-of-the-art performance on the peptide dataset and generates a competitive performance on the protein dataset compared with several existing methods, indicating the effectiveness of our model in predicting the toxicity of peptides and proteins. Finally, an online web server of ToxIBTL is implemented and made publicly accessible at http://server.wei-group.net/ToxIBTL.

2 Materials and methods

2.1 Benchmark dataset

In this study, we use two benchmark datasets to evaluate the performance of our proposed model for predicting the toxicity of proteins and peptides. The first dataset established by Pan *et al.* (2020) is employed to build models for predicting protein toxicity. It contains 4472 toxic animal proteins used as positive samples and 6341 non-toxic animal proteins used as negative samples. Each sequence in the testing set has a similarity $< 40\%$ to that in the training set, meanwhile, there are no protein sequences with the same domain from the Pfam clans (El-Gebali *et al.*, 2019) between these two sets.

The second benchmark dataset created in our previous work (Wei *et al.*, 2021) is used to build models for peptide toxicity prediction, which consists of 3864 samples with a range of 10–50 residues. The positive samples in this dataset are toxic peptide sequences, which are experimentally validated. Similarly, the negative samples are non-toxic peptide sequences, which have the same number as the positive ones. The sequence similarity between any two peptide sequences is less than 90%, which can avoid the evaluation bias introduced by sequence similarity. For training, about 85% of toxic and non-toxic peptides are randomly selected to fine-tune our model for predicting the toxicity of the peptide, and the remaining peptides are adopted as testing set to evaluate the performance of the fine-tuned model. A statistical summary of these two datasets used in this study is shown in Table 1.

In addition, to better understand the difference intuitively between the training set and the testing set, we visualize the data distribution from two aspects. One is the number of each amino acid type shown in Figure 1 and Supplementary Figure S1, the other is the

Table 1. Overview of the two benchmark datasets

	Dataset	Number of positives	Number of negatives
Protein	Training set	4413	5671
	Testing set	59	670
Peptide	Training set	1642	1642
	Testing set	290	290

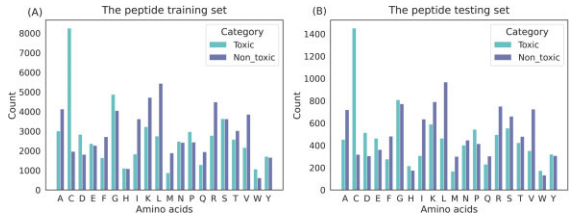

Fig. 1. The amino acids distribution of toxic and non-toxic peptides in the training set and the testing set. (**A**) Distribution map in the peptide training set. (**B**) Distribution map in the peptide testing set

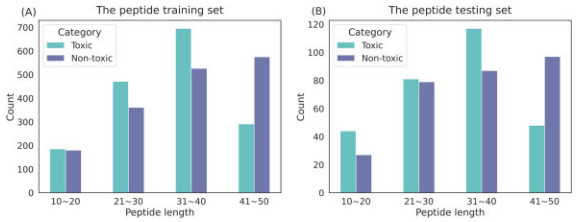

Fig. 2. The length distribution of toxic and non-toxic peptides in the training set and testing set. (**A**) Distribution map in the peptide training set. (**B**) Distribution map in the peptide testing set

distribution of sequence length shown in Figure 2 and Supplementary Figure S2.

2.2 The proposed method architecture

In this section, we introduce the proposed ToxIBTL architecture for predicting peptide toxicity based on the information bottleneck principle and transfer learning, as shown in Figure 3. The workflow of ToxIBTL mainly contains three steps, including sequence encoding, optimization and classification. In the first step, to encode the evolutionary information, we convert raw sequences to evolutionary profiles and feed them into the hybrid network CNN_BiGRU to automatically capture latent local and global information; simultaneously, to capture physicochemical information, raw sequences are sent into the FEGS model to obtain graphical features and statistical features. In the second step, we directly concatenate the evolutionary and physicochemical features and use the information bottleneck principle to optimize the concatenated features. In the third step, the optimized features are used to determine the sequence as toxic or non-toxic one. Through the proposed ToxIBTL, we first train a model on the protein dataset; then, the pre-trained model is transferred to fine-tune a new model on the peptide set. The hyperparameters of ToxIBTL are given in Supplementary Material.

2.3 Sequence encoding

2.3.1 CNN_BiGRU network

Previous studies illustrate that extracting the evolutionary information of proteins contributes to protein sequence analysis, especially to analyzing proteins with low sequence similarities (Liu *et al.*, 2020). As shown in Figure 3B, the standard BLOSUM62 scoring matrix is adopted to encode peptide (or protein) sequences, which is developed by analyzing the frequencies of amino acid substitutions in clusters of related proteins. Within each cluster, the amino acid sequences are at least 62% identical when two proteins are aligned. The score in BLOSUM62 matrix reflects the chance that one amino acid is substituted for another in a cluster. Given a peptide (or protein) sequence $\boldsymbol{P}$, each residue is encoded by the corresponding row of this matrix. Therefore, a sequence $\boldsymbol{P}$ can be represented as the following matrix:

$$\text{BLOSUM62} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,20} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,20} \\ \vdots & \vdots & \ddots & \vdots \\ p_{l,1} & p_{l,2} & \cdots & p_{l,20} \end{bmatrix}, \quad (1)$$

where l is the length of sequence $\boldsymbol{P}$ and 20 is the number of standard amino acids.

In order to extract the latent discriminative features hiding in evolutionary information derived from the BLOSUM62 matrix, a hybrid network is designed, called CNN_BiGRU that consists of CNN and BiGRU, to effectively capture the contextual and semantic information of peptide (or protein) sequences. Specifically, the BLOSUM62 matrix of a peptide (or protein) sequence is fed into a 2D convolutional layer with a non-linear activation function (e.g. *relu*) to extract the local correlation between amino acids through the local perceptual domain. Afterward, the output of the convolutional layer is taken as the input of the BiGRU layer to obtain the long and short dependency information amongst extracted local correlation and capture sequence-order effects (Li *et al.*, 2017).

Notably, due to different lengths among sequences, following the same protocol in the study (Pan *et al.*, 2020), all input sequences are truncated to a maximum length of 1002 to meet the input requirement of neural networks. For sequences < 1002 in length, zero-padding is performed. It's worth noting that in the BiGRU, the *pack_padded_sequence*, a function in Pytorch, is used to make sure BiGRU will not perform the calculation on padding elements, and then the hidden vector of the last valid input is picked up as the representation for a sequence. So, it not only reduces unnecessary calculations, but also avoids the impact of padding bits on the sequence feature representation. Moreover, the dropout technique is introduced to reduce the risk of overfitting.

2.3.2 FEGS model

Feature representation is a critical step for developing a precise prediction model (Li and Liu, 2020; Tan *et al.*, 2019). In order to better represent each peptide (or protein) sequence to encompass the perspective of its biophysical and biochemical properties, FEGS, a feature extraction model of protein sequence using the physicochemical properties of amino acids and statistical information of protein sequences, is introduced to extract the graphical and statistical features of peptide (or protein) sequences. As shown in Figure 3C, we can get a 578-dimensional feature vector to represent a peptide (or protein) sequence.

For the graphical feature encoding, 158 physicochemical properties of amino acids are effectively used to transform a peptide (or protein) sequence into a 158-dimensional numerical vector, which are selected from the AAindex database (Kawashima and Kanehisa, 2000). First, the 20 amino acids are ranked in ascending order according to their physicochemical indices. Second, the ranked 20 amino acids are sequentially positioned on the circumference of the bottom of a right circular cone of height 1 according to the following formula:

$$\psi(\alpha_i) = \left(\cos\frac{2\pi i}{20}, \sin\frac{2\pi i}{20}, 1\right), \ i = 1, \ 2, \ldots, \ 20, \quad (2)$$

where α_i denotes one of the ranked 20 amnio acids. Subsequently, 400 amino acid pairs are arranged on the underside of the right circular cone according to the following formula:

$$\omega(\alpha_i\alpha_j) = \psi(\alpha_i) + \frac{1}{4}\left(\psi(\alpha_j) - \psi(\alpha_i)\right), \ i, \ j = 1, 2, \ldots, 20, \quad (3)$$

where $\psi(\alpha_i\alpha_j)$ represents one of the 400 amino acid pairs.

Based on the above right circular cone, the 3D graphical curve of a given peptide (or protein) sequence $P = p_1p_2 \ldots p_l$ can be constructed. Extend the origin $S_0(0, 0, 0)$ to the point $S_1(x_1, y_1, z_1)$ corresponding to the first amino acid p_1, and then the point S_1 is extended to the point $S_2(x_2, y_2, z_2)$ corresponding to the second amino acid p_2, and so on. The coordinate of the point $S_i(x_i, y_i, z_i)$ is determined by the following formula:

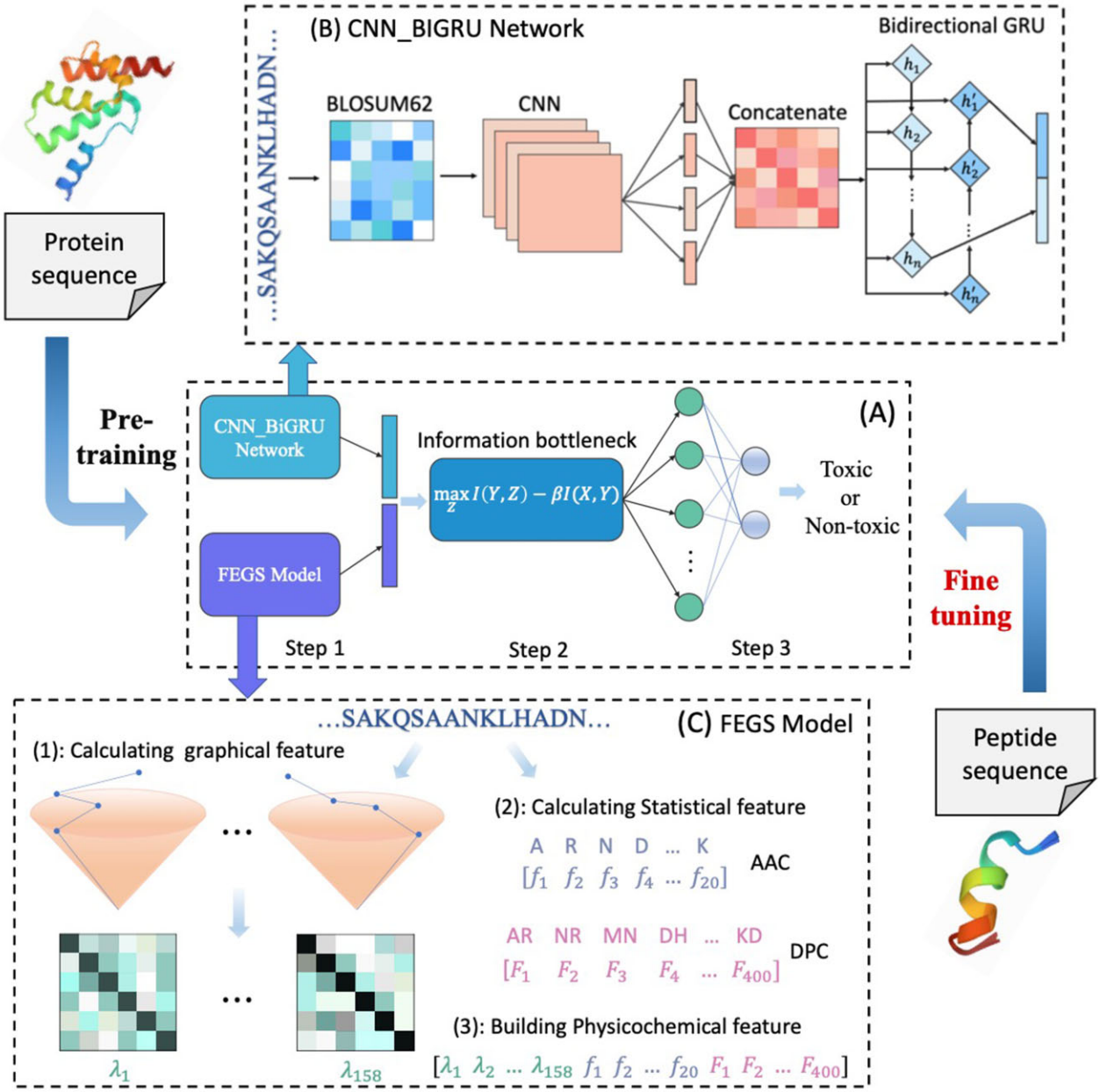

Fig. 3. The flowchart describes the overall implementation of the proposed transfer learning approach for predicting peptide toxicity. (**A**) The architecture of proposed ToxIBTL. We first train a model from scratch on the protein set, then fine-tune the pre-trained model on the peptide set to classify toxic and non-toxic peptides. (**B**) The architecture of the CNN_BiGRU network. (**C**) The pipeline of the FEGS model

$$\phi(p_i) = \phi(p_{i-1}) + \psi(p_i) + \sum_{\alpha_1,\alpha_2 \in \{A,\ C,\ D,\ldots,Y\}} f_{\alpha_1\alpha_2}\omega(\alpha_1\alpha_2), \quad (4)$$

where $\phi(p_0) = (0,0,0)$, p_i is the ith amino acid, and $f_{\alpha_1\alpha_2}$ represents the frequency of the amino acid pair $\alpha_1\alpha_2$ occurring in the sequence P. Then, the 3D graphical curve S of the sequence P is obtained. Afterward, a nonnegative symmetric matrix M is computed for the graphical curve, whose each off-diagonal entry $m_{i,j}(i \neq j)$ is a quotient of the Euclidean distance between points S_i and S_j in the graphical curve and the sum of geometrical lengths of edges between S_i and S_j along the graphical curve, and the diagonal elements are set to zero. The largest eigenvalue of the matrix M is computed and divided by the length of the corresponding sequence to characterize the corresponding graphical curve. Since each physicochemical property corresponds to a graphical curve, a 158-dimensional vector can be generated as the graphical features for a peptide (or protein).

The statistical features consist of two classical and interpretable feature descriptors as follows:

1. Amino acid composition reflects the frequency of 20 different amino acids occurring in a peptide (or protein) sequence. It can generate a 20-dimensional feature vector and be computed as follows:

$$f(t) = \frac{N(t)}{L},\ t \in \{A,\ C,\ D,\ E, \ldots, Y\}, \quad (5)$$

where $N(t)$ is the number of amino acid type t and L is the length of a peptide (or protein) sequence.

2. Dipeptide composition calculates the frequency of each type of amino acid pair in a peptide (or protein) sequence, which describes the fraction of amino acids and their local order. It has a 400-dimensional feature vector and is defined as follows:

$$f(a,\ b) = \frac{N_{ab}}{L-1},\ a,\ b \in \{A,\ C,\ D,\ E, \ldots, Y\}, \tag{6}$$

where N_{ab} is the number of amino acid pairs formed by amino acid types a and b.

2.4 Optimization

To represent a peptide (or protein) sequence, we linearly combine the evolutionary features extracted from CNN-BiGRU network with the physicochemical features from FEGS model. However, the combined features from different views may contain noisy and redundant information thereby causing poor predictive results. To extract more discriminative features, the information bottleneck (Tishby *et al.*, 2000), an approach based on information theory, is introduced to maximize the mutual information between the label Y and the learned representation Z to make sure the learned Z is maximally informative about Y, while simultaneously minimizing the mutual information between the learned Z and the original sequence representation X to filter out irrelevant information as much as possible. Note that X denotes the combined features; Y is the label of X; Z is learned from X by the information bottleneck principle. Following the study (Alemi *et al.*, 2016), our goal can be formulated as the following function:

$$\max_{Z} I(Y,\ Z) - \beta \mathrm{I}(X,\ Z), \tag{7}$$

where β is a parameter to control the tradeoff between accuracy and complexity. Low β corresponds to high mutual information between Z and Y, and low compression. The function I is defined as the mutual information between two random variables A and B as follows:

$$I(A,\ B) = \int \mathrm{d}a\ \mathrm{d}b\ p(a,\ b) \log \frac{p(a,\ b)}{p(a)p(b)}, \tag{8}$$

where a and b are instances of random variables A and B, respectively; $p(a,\ b)$ is the joint probability distribution of A and B; $p(a)$ and $p(b)$ are the marginal probability distributions of A and B, respectively.

Following the solution for mutual information terms described in the study (Alemi *et al.*, 2016), we assume $p(Z|X,\ Y) = p(Z|X)$ according to the Markov chain $Y \leftrightarrow X \leftrightarrow Z$. So, the first term in Equation (7) can be expressed as:

$$I(Y,\ Z) = \int \mathrm{d}y\ \mathrm{d}z p(y,\ z) \log \frac{p(y,\ z)}{p(y)p(z)} = \int \mathrm{d}y\ \mathrm{d}z p(y,\ z) \log \frac{p(y|z)}{p(y)}\ . \tag{9}$$

Since $p(y|z)$ is intractable, we use $q(y|z)$, which can be learned by our network, to approximate $p(y|z)$. Since KL-divergence between $p(y|z)$ and $q(y|z)$ is always positive, we have:

$$\begin{aligned} I(Y,Z) &\geq \int \mathrm{d}y\ \mathrm{d}z\, p(y,z) \log \frac{q(y|z)}{p(y)} \\ &= \int \mathrm{d}y\ \mathrm{d}z\, p(y,z) \log q(y|z) - \int dy\, p(y) \log p(y). \end{aligned} \tag{10}$$

The term $-\int \mathrm{d}y p(y) \log p(y)$ denotes the entropy of the label Y, which does not affect our optimization, so it can be ignored. Employing the Markov assumption, $p(y,\ z)$ can be rewritten as $p(y,\ z) = \int \mathrm{d}x\ p(x,\ y,\ z) = \int \mathrm{d}x\ p(x)p(y|x)p(z|x)$, and then the first term of our objective becomes:

$$I(Z, Y) \geq \int \mathrm{d}x\ \mathrm{d}y\ \mathrm{d}z\ p(x)p(y|x)p(z|x) \log q(y|z). \tag{11}$$

For the term $I(X,\ Z)$, it can be written as:

$$\begin{aligned} I(Z,X) &= \int \mathrm{d}z\ \mathrm{d}x\ p(z,x) \log \frac{p(z|x)}{p(z)} \\ &= \int \mathrm{d}z\ \mathrm{d}x\ p(z,x) \log p(z|x) - \int \mathrm{d}z\ p(z) \log p(z). \end{aligned} \tag{12}$$

Similarly, we use $r(z)$ to approximate $p(z)$ and use the property of KL-divergence between $p(z)$ and $r(z)$, then we can have:

$$I(Z,\ X) \leq \int \mathrm{d}z\ \mathrm{d}x p(x) p(z|x) \log \frac{p(z|x)}{r(z)}. \tag{13}$$

Combine Equations (7), (11) and (13), we can obtain:

$$\begin{aligned} \mathrm{I}(Z, Y) - \beta \mathrm{I}(Z, X) \geq & \int \mathrm{d}x\ \mathrm{d}y\ \mathrm{d}z\ p(x)p(y|x)p(z|x) \log q(y|z) \\ & - \int \mathrm{d}z\ \mathrm{d}x p(x) p(z|x) \log \frac{p(z|x)}{r(z)} = L\ . \end{aligned} \tag{14}$$

Here, the Monte Carlo sampling method (Shapiro, 2003) is used to approximate the integral over x and y. Therefore, L can be approximated as:

$$L \approx \frac{1}{N} \sum_{i=1}^{N} \left[\int \mathrm{d}z\ p(z|x_i) \log q(y_i|z) - \beta p(z|x_i) \log \frac{\mathrm{p}(z|x_i)}{r(z)} \right], \tag{15}$$

where N is the number of the sampled data. Assume $p(z|x) = \mathcal{N}(z|f^{\mu}(x), f^{\sigma}(x))$ as an encoder, where f^{μ} and f^{σ} are two multilayer perceptions (MLPs) that learn the mean μ and the covariance σ (after a *softplus* function transform) for , respectively. Especially, each MLP has k neurons and k is the dimension of μ and σ. Then, the reparameterization trick (Kingma and Welling, 2013) is used to make $z = f^{\mu}(x) + f^{\sigma}(x)\varepsilon$, where ε is a Gaussian random variable and z is our final latent representation to characterize a sequence, whose dimension is k. Next, supposing that the KL-divergence can be computed between $p(z|x)$ and $r(z)$, put all the above together and our objective function (i.e. Equation 15) can be rewritten as follows:

$$L_{IB} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left[\mathbb{E}_{\epsilon \sim p(\epsilon)} (-\log q(y_i|z_i)) + \beta \mathrm{KL}\big((p(z|x_i), r(z)) \big) \right]. \tag{16}$$

Note that the first term in Equation (16) is the cross-entropy between z and y, and then backpropagation algorithm can be directly applied to L_{IB} to update network parameters, which let us get a compressive and accurate latent representation that contained more relevant information and less superfluous information. Therefore, we take L_{IB} as the loss function of our model to obtain the latent feature vector z for improving prediction performance.

2.5 Classification

This step mainly employs a fully connected layer and a *sigmoid* layer. The latent feature vector z learned by information bottleneck principle is forwarded into the fully connected layer with a *relu* activation function, and then a *sigmiod* layer is employed to perform classification as follows:

$$\text{output} = \ \text{sigmoid}(\text{relu}(wz + b)), \tag{17}$$

$$\text{sigmoid}(\text{s}) = \frac{1}{1 + e^{-s}}, \tag{18}$$

where w stands for the weights of the fully connected layer and b stands for the corresponding bias. The values of output are probabilities ranging from 0 to 1. If the probability value is > 0.5, the sequence belongs to the toxic peptide (or protein) class, and vice versa.

2.6 Transfer learning

Transfer learning is a machine learning methodology that aims at storing the knowledge learned from a task with a large number of available labeled data and applying it to a different but related task with a small set of data (Zhuang *et al.*, 2021). Therefore, for a new task, instead of starting the learning process from scratch, we can start with the patterns learned from a related task. Here, the protein dataset is used to pre-train the neural network architecture shown in Figure 3A. Next, transfer learning is performed by further fine-tuning the pre-trained model on the peptide dataset as shown in Figure 3. During transfer learning, we change the number

Table 2. Predictive performance of various methods on protein dataset

Method	F1_score	MCC	auROC	auPRC
BLAST	0.800	0.801	—	—
BLAST-score	0.789	0.775	0.868	0.818
InterProScan	0.347	0.402	—	—
Hmmsearch	0.185	0.307	—	—
ClanTox	0.620	0.604	0.903	0.612
ToxinPred-RF	0.667	0.638	0.948	0.716
ToxinPred-SVM	0.677	0.648	0.938	0.712
ToxDL	0.809 (±0.022)	0.793 (±0.024)	**0.989 (±0.002)**	**0.913 (±0.014)**
ToxIBTL (this study)	**0.830 (±0.007)**	**0.816 (±0.008)**	0.953 (±0.001)	0.847 (±0.002)

Note: For a fair comparison, we report the average after experimenting 10 times for ToxIBTL. The best performance amongst all methods is denoted as boldface.

Table 3. Predictive performance of various methods on peptide dataset

Methods	SN	SP	FDR	ACC	MCC
ClanTox	0.855	0.888	0.132	0.872	0.743
ToxinPred-RF	0.918 (±0.016)	0.904 (±0.018)	0.094 (±0.017)	0.911 (±0.010)	0.823 (±0.021)
ToxinPred-SVM	0.893 (±0.012)	0.924 (±0.016)	0.077 (±0.015)	0.909 (±0.011)	0.817 (±0.023)
Only-GNN	0.869 (±0.008)	0.898 (±0.006)	0.112 (±0.005)	0.885 (±0.002)	0.778 (±0.003)
Only-CNN_BiLSTM	0.947 (±0.012)	0.895 (±0.019)	0.114 (±0.017)	0.919 (±0.007)	0.840 (±0.012)
ATSE	**0.965 (±0.003)**	0.940 (±0.003)	0.068 (±0.003)	0.952 (±0.002)	0.903 (±0.004)
ToxIBTL (This study)	0.963 (±0.003)	**0.954 (±0.002)**	**0.046 (±0.002)**	**0.960 (±0.002)**	**0.921 (±0.003)**

Note: For a fair comparison, we report the average after experimenting 10 times for ToxIBTL. The best performance amongst all methods is denoted as boldface.

of neurons in the penultimate fully connected layer and the parameter β in lost function, whereas other hyperparameters in networks remain unchanged as changing them does not yield better predictive performance. In addition, all the weights in network layers are fine-tuned without freezing any weights of certain layers because using all weights to fine-tune the pre-trained model performs better than freezing certain layers.

3 Model performance assessment

3.1 Evaluation metrics

In this study, the protein toxicity prediction task suffers from severe class imbalance, which is not capable to be evaluated with regular ACC. Therefore, we use the same four metrics in the study (Pan *et al.*, 2020) to evaluate the performance of our model, including the F1_score, the Matthews correlation coefficient (MCC), the area under the receiver operating characteristic curve (auROC) and the are under the precision-recall curve (auPRC). For the peptide toxicity prediction task without the class-imbalanced issue, seven sets of commonly used metrics are employed to evaluate the prediction results, including sensitivity (SN), specificity (SP), false discovery rate (FDR), ACC, MCC, auROC and auPRC. The metrics mentioned above are defined as follows:

$$\begin{cases} \mathrm{SN} = \dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\ \mathrm{SP} = \dfrac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \\ \mathrm{FDR} = \dfrac{\mathrm{FP}}{\mathrm{TP}+\mathrm{FP}} \\ \mathrm{ACC} = \dfrac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \\ \mathrm{F1\_score} = 1 + \dfrac{2\mathrm{TP}}{\mathrm{FP}+\mathrm{FN}} \\ \mathrm{MCC} = \dfrac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})\times(\mathrm{TP}+\mathrm{FN})\times(\mathrm{TN}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN})}} \end{cases}, \tag{19}$$

where TP (true positive) and TN (true negative) represent the numbers of correctly predicted positives and negatives, respectively. FP (false positive) and FN (false negative) are the numbers of wrongly predicted positives and negatives, respectively. SN and SP are employed to measure the positive and negative predictive ability of a classifier. FDR reflects the percentage of FPs among all the predicted positives. MCC, F1_score and ACC are adopted to evaluate the overall prediction performance of a classifier. In addition, the ranges of auROC and auPRC are from 0.5 to 1. The higher the score, the better the prediction performance of the model.

3.2 Performance comparison with existing methods on protein dataset

To evaluate the effectiveness of our proposed model, we compare it with existing methods on the protein dataset, including alignment-based methods [BLAST (Altschul *et al.*, 1997), BLAST-score, InterProScan (Jones *et al.*, 2014), Hmmsearch (Potter *et al.*, 2018)], machine learning-based methods [ClanTox (Naamati, *et al.*, 2009), ToxinPred (i.e. ToxinPred-RF and ToxinPred-SVM) (Gupta *et al.*, 2013)] and deep learning-based methods [ToxDL (Pan *et al.*, 2020)]. They employed different features for identifying toxic proteins, including multiple sequence alignment-based features, sequence-based features and protein domain-based features. More details about them can be found in Supplementary Material. Their predictive results are listed in Table 2.

As can be seen from Table 2, except for ToxDL, ToxIBTL achieves the best performance with the F1_score, MCC, auROC and auPRC of 0.830, 0.816, 0.953 and 0.847, respectively. Specifically, the F1_score, MCC, auROC and auPRC of the proposed predictor are 3–64.5%, 1.5–50.9%, 0.5–8.5% and 2.9–23.5% higher than other predictors. This result demonstrates that ToxIBTL can capture more effective information related to toxicity than the alignment-based methods and the machine learning-based methods. When compared with ToxDL, ToxIBTL improves upon the F1_score from 0.809 to 0.830 (a relative improvement of 2.6%) and improves upon the MCC from 0.793 to 0.816 (a relative improvement of 2.9%), whereas the values of auROC and auPRC of ToxIBTL are lower, which indicates that ToxIBTL can achieve competitive performance to ToxDL. In addition, the standard deviation of each

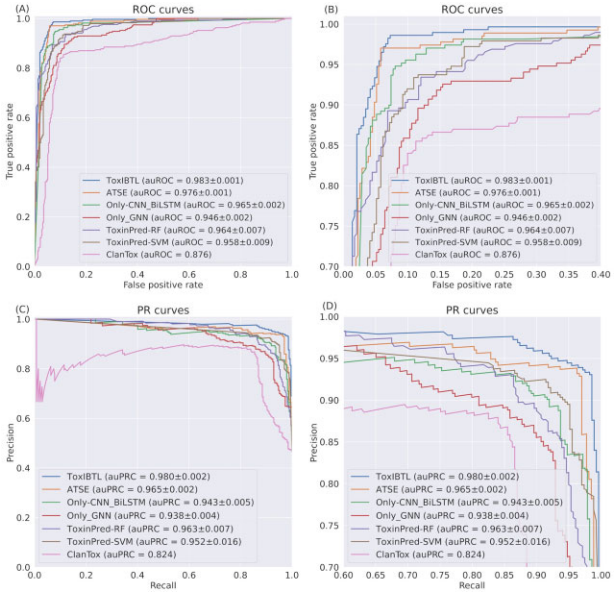

Fig. 4. ROC and PR curves of different methods. (**A**) ROC curves of the proposed ToxIBTL and competing methods. (**B**) It shows the same as (A), but zoomed-in on an interesting region. (**C**) PR curves of proposed ToxIBTL and competing methods. (**D**) It shows the same as (C), but zoomed-in on an interesting region

metric of the proposed method is lower than ToxDL, indicating that ToxIBTL can achieve more stable performance.

3.3 Performance comparison with existing methods on peptide dataset

In this section, we compare the proposed ToxIBTL with several state-of-the-art methods specially designed for peptide toxicity prediction, including ClanTox, ToxinPred (ToxinPred-RF and ToxinPred-SVM), ATSE (Wei *et al.*, 2021) and ATSE's variants (Only-GNN and Only-CNN_BiLSTM). More details about them can be found in Supporting Information. The comparative results are presented in Table 3. From Table 3, we can see that ToxIBTL outperforms all the competing methods in terms of SP, FDR, ACC and MCC. To be specific, ToxIBTL achieves an SP of 0.954, an ACC of 0.960 and an MCC of 921, which generates a relative improvement over the runner-up ATSE of 1.49%, 0.84% and 1.99%, respectively, and reduces FDR from 0.068 to 0.046 (a relative reduction of 32.35%). For SN, ToxIBTL is 0.963, whereas ATSE obtains 0.965. The difference between them is very small, indicating the abilities of these two methods to predict positive samples are very close.

These results discussed above demonstrate that ToxIBTL can extract more accurate and effective toxicity-specific features to characterize peptide sequences for distinguishing both toxic peptides and non-toxic ones. There are two main reasons for the outstanding performance of our model. First, compared with machine learning-based methods using handcrafted features to build predictors, which depend on the limited knowledge of peptides, ToxIBTL can not only automatically capture high-latent features by data driving, but also learn context-sensitive information of sequence by fuzing both CNN and BiGRU, which is critical for sequence analysis. Second, compared with ATSE and its variants, on one hand, ToxIBTL introduces transfer learning to apply common and generalizable knowledge learned from proteins to peptides, which addresses the problem of small data on toxic peptides, on the other hand, ToxIBTL adopts the information bottleneck principle to find a better representation that contains more relevant information for predicting toxicity and less superfluous information that interferes with predicting results. Therefore, there is no surprise that ToxIBTL achieves the best performance when combining transfer learning strategy and information bottleneck principle.

To intuitively illustrate the superiority of ToxIBTL, we display the ROC and PR curves obtained from different methods in Figure 4. Figure 4B and D shows a partial zoomed-in view of Figure 4A and C, respectively. We can observe that the auROC and auPRC of the proposed method are 0.983 and 0.980, respectively, which are 0.7–10.7% and 1.5–15.6% higher than other methods. These comparative results further suggest its ability to perform better with unknown both toxic and non-toxic peptides when compared with the existing methods.

3.4 Effectiveness analysis of information bottleneck principle

To explore the effectiveness of introducing information bottleneck principle in our feature learning scheme, on the peptide dataset, we perform comparison experiments on four feature representations, including the output of the CNN_BiGRU, the output of FEGS model, the combination of the above two feature representations, and the feature representation obtained after optimization. For convenience discussion, we denote these four feature representations in turn as CNN_BiGRU_F, FEGS_F, CNN_BiGRU+FEGS_F and OptimaizedF. We use six common classifiers, including random forest (RF), SVM (with RBF kernel), Gaussian naive Bayes (GNB), LightGBM (Ke *et al.*, 2017), logistic regression (LR) and k-nearest neighbors (KNN), to carry out comparison experiments. Notably, we employ the 10-fold cross-validation technique to perform comparison experiments. The results are presented in Figure 5.

From Figure 5, it can be seen that compared with the other feature representations, OptimizedF exhibits the best overall predictive performance on each classifier in terms of SN, SP, ACC and MCC, especially on the classifiers RF and GNB, which indicates the effectiveness of using information bottleneck principle to integrate different features, leading to a more effective and discriminative feature representation that better suits to most common classifiers. Note that CNN_BiGRU+FEGS_F, obtained by concatenating two features directly, achieves slightly better performance than CNN_BiGRU_F, even though containing evolutionary and physicochemical information. The possible reason is that the directly concatenated features contain more irrelevant and noisy information, which poses a bad effect on classification and hinders the generation of stable and significant performance. Therefore, we apply the information bottleneck principle to effectively fuze evolutionary information and physiochemical information, containing in CNN_BiGRU+FEGS_F, to maintain relevant information as much as possible and filter out superfluous information, which can lead to a more accurate and discriminative feature representation (i.e.

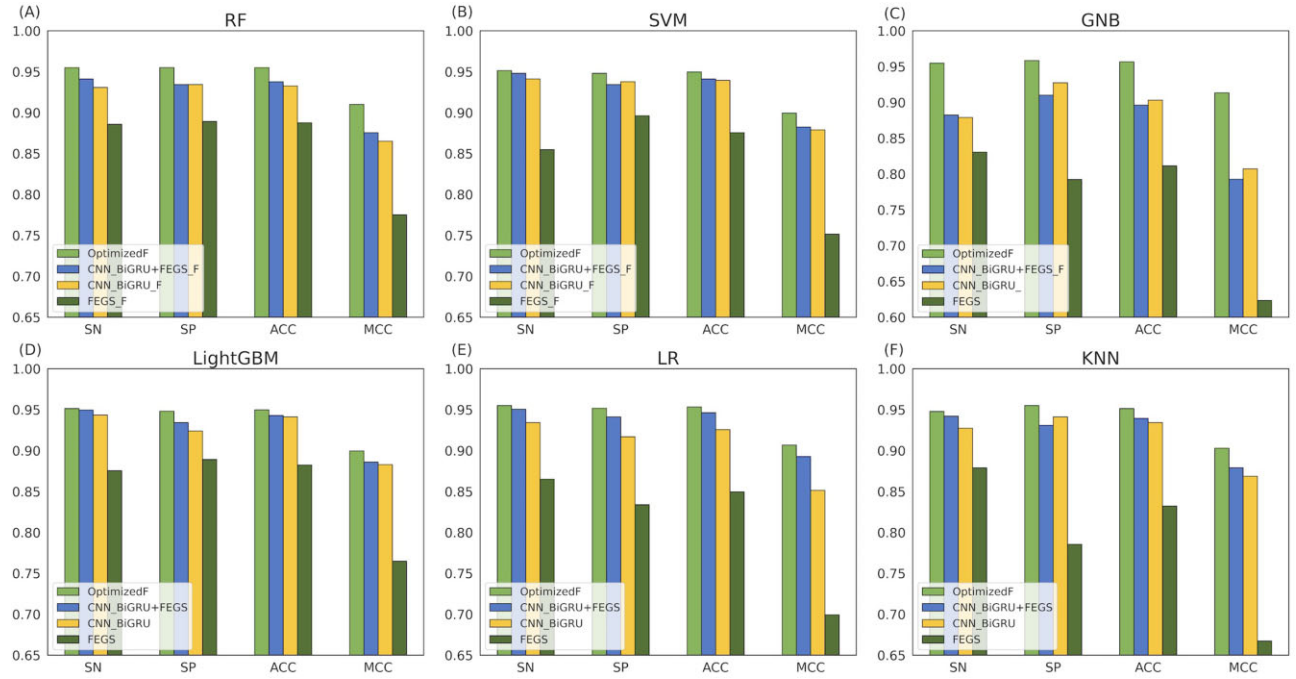

Fig. 5. Ten-fold cross-validation metrics comparison results of CNN_BiGRU_F, FEGS_F, CNN_BiGRU+FEGS_F and OptimaizedF based on six common classifiers. (**A**) Results on RF. (**B**) Results on SVM. (**C**) Results on GNB. (**D**) Results on LightGBM. (**E**) Results on LR. (**F**) Results on KNN

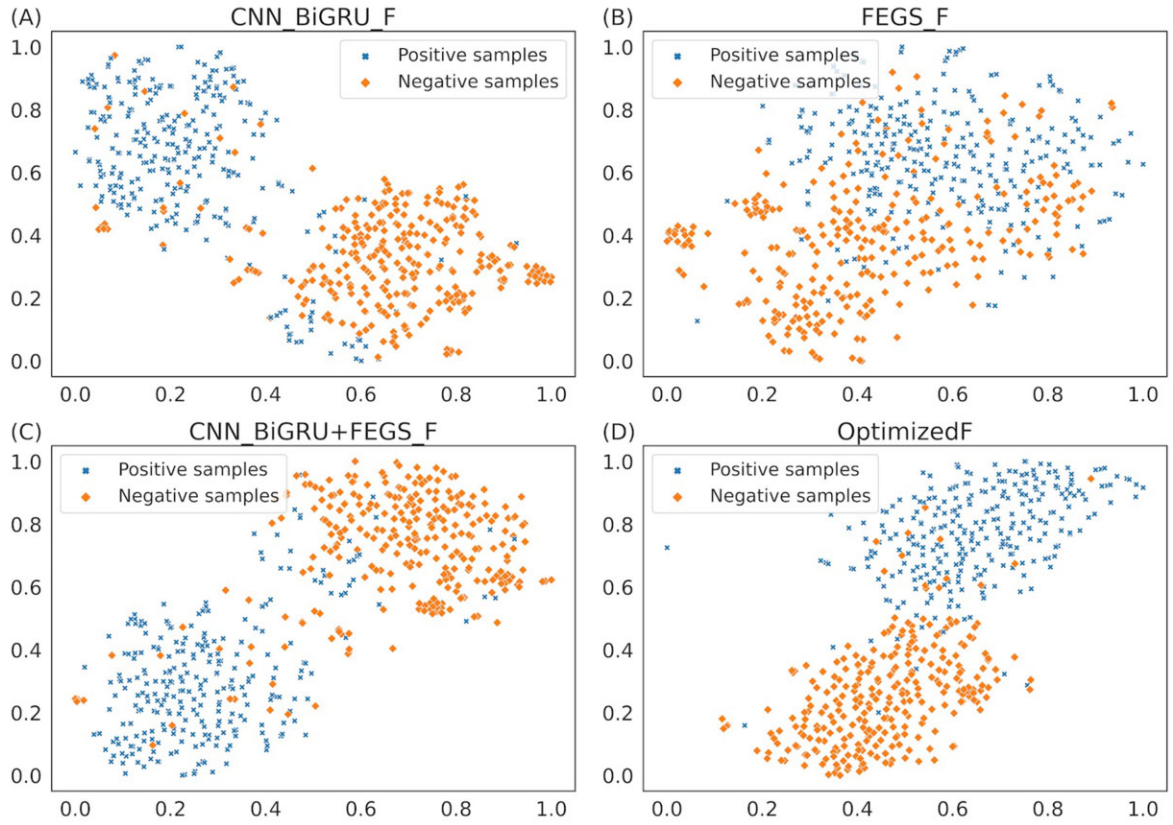

Fig. 6. t-SNE visualization of the feature space. (**A**) Visualization of CNN_BiGRU_F. (**B**) Visualization of FEGS_F. (**C**) Visualization of CNN_BiGRU+FEGS_F. (**D**) Visualization of OptimaizedF. The positive samples and negative samples are shown in blue and yellow colors, respectively

OptimizedF) to better classify both toxic peptides and non-toxic ones. Moreover, after using the information bottleneck principle, the dimension of OptimizedF is smaller than that of CNN_BiGRU+FEGS_F, which not only reduces the training time but also avoids the 'curse of dimensionality'. To summarize, the comparison experiments on common classifiers highlight the importance and requisite to employ the information bottleneck principle to integrate comprehensive feature encodings into a consolidated framework for further enhancing the model performance.

To more intuitively explain the effectiveness of the information bottleneck principle, we use t-SNE (Van der Maaten and Hinton, 2008) to project the four representations of each sequence into a 2D space for visualization. The results are illustrated in Figure 6. As shown in Figure 6B, although the clustering effect is the worst, the distribution of positive and negative samples can still be observed roughly, demonstrating that FEGS_F also captures some relevant information. From Figure 6A and C, we can see that after concatenating CNN_BiGRU_F and FEGS_F directly, there is a slight improvement in the clustering effect. With the optimization of the information bottleneck principle, as shown in Figure 6D, two categories distribute more clearly and compactly, which indicates the strong ability of the information bottleneck principle to retain relevant information for predicting peptide toxicity while discarding the redundant information in the representation. Furthermore, it also increases the interpretability of our model from the perspective of feature learning strategy.

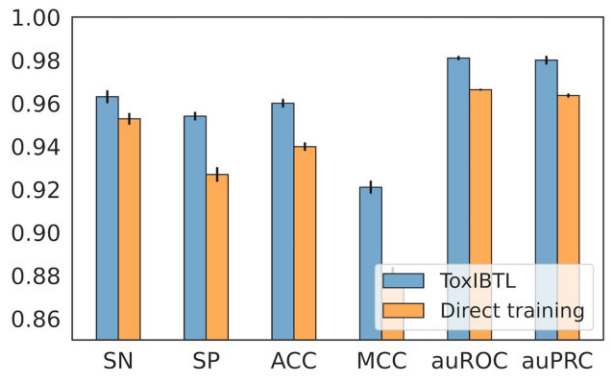

Fig. 7. Performance comparison between ToxIBTL and the direct training method on peptide dataset

3.5 Effectiveness of transfer learning

To investigate whether transfer learning can improve predictive performance with small data, we compare ToxIBTL with the direct training method that directly utilizes our designed neural network to train the predictive model on the peptide dataset. The comparative results are displayed in Figure 7. As shown in Figure 7, we can see that compared with the direct training method, ToxIBTL achieves an obvious improvement in terms of six metrics (i.e. SP, SN, ACC, MCC, auROC and auPRC). Specifically, the ACC and MCC of ToxIBTL are 0.960 and 0.921, which are 2% and 4.1% higher than the direct training method. The results demonstrate that transfer learning is a useful strategy and can increase the model power for identifying toxic and non-toxic peptides. The possible reason is that transfer learning can effectively transfer common knowledge from proteins with a large amount of data to peptides and obtain a more discriminative feature representation to represent intrinsic characteristics of peptides. Therefore, due to the limited data, the identifying power of the direct training method is relatively lower.

3.6 Analysis of the impact of peptide length on performance

To study the impact of peptide length on the performance of predictors, we compare the misclassification rate of various predictors, including ClanTox, ToxinPred-RF, ToxinPred-SVM, Only-GNN, Only-CNN_BiLSTM, ATSE and ToxIBTL, toward toxic peptides and non-toxic peptides of diverse length (length $\in$ [10, 20], [21, 30], [31, 40], [41, 50]). This study is conducted on the peptide testing set. The results are displayed in Figure 8. From Figure 8A, we can see that ClanTox, ToxinPred-RF, ToxinPred-SVM and Only-GNN have relatively higher rates of misclassification for shorter toxic peptides, which indicates they are highly biased toward longer toxic peptides. As a result, they classify more short-length peptides as non-toxic ones. For Only-GNN and Only-CNN_BiLSTM, apart from the length$\in$ [21, 30], they all achieve great performance in other length intervals. For length $\in$ [10, 20], the proposed ToxIBTL can predict the toxicity of peptides accurately, and for other length intervals, it also achieves relatively lower misclassification rates compared with other predictors.

From Figure 8B, it can be seen that ClanTox and ToxinPred-RF are biased toward non-toxic peptides for length $\in$ [10, 20]. In the case of ToxinPred-SVM and Only-GNN, they all perform worse for all length intervals. For Only-CNN_BiLSTM, it is biased toward non-toxic peptides for a longer length and gets a higher misclassification rate for short-length non-toxic ones. For ATSE, it performs

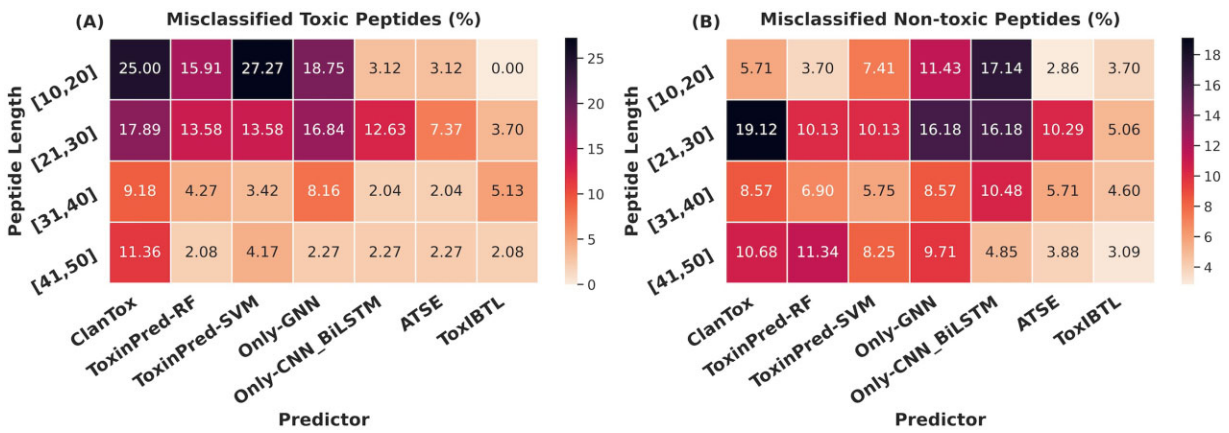

Fig. 8. Heat maps showing misclassified both toxic peptides and non-toxic peptides of varying lengths by different predictors. (**A**) Heat map of misclassified toxic peptides. (**B**) Heat map of misclassified non-toxic peptides

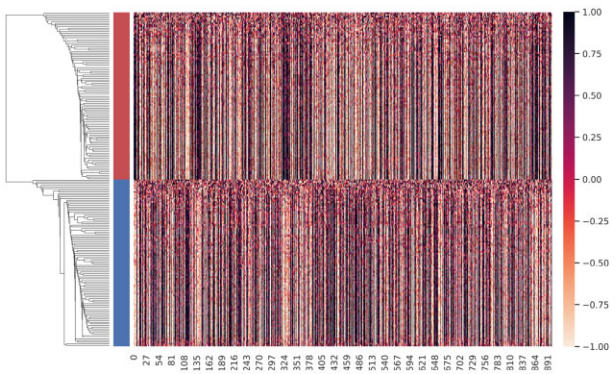

Fig. 9. The clustering analysis chart of features obtained by ToxIBTL on peptide dataset

worse in length $\in [10, 20]$ compared with other length intervals. For our proposed ToxIBTL, we don't observe the obvious bias toward a certain length due to the relatively balanced misclassification rates.

To sum up the above, the proposed ToxIBTL does not have a distinct bias toward toxic or non-toxic peptides and can accurately identify both toxic and non-toxic peptides for diverse lengths. Therefore, our model can be taken as a promising predictor for predicting peptides of any length with great confidence.

3.7 Feature analysis

Discriminative features are the key to build a powerful computational classifier. When compared with other existing methods, the proposed ToxIBTL has two main advantages: (i) integrates evolutionary information and physiochemical information to represent peptides, and (ii) uses the information bottleneck principle and transfer learning to extract effective features containing more relevant information and less noisy information. The latent and toxicity-related features, i.e. toxic and non-toxic peptides have distinctly different characteristics, are constructed with 900 neurons in our neural networks, which correspond to toxic or non-toxic peptides. To highlight the discriminative power of toxicity-related features extracted by the proposed ToxIBTL, for comparison, we randomly select 100 toxic peptides and 100 non-toxic ones and perform a clustering analysis of the features on the randomly selected peptides. The clustering results are presented in Figure 9. As shown in Figure 9, we can easily see that: (i) toxic and non-toxic peptides are clustered into two distinct sub-trees, and (ii) the peptides of the same type tend to show similar values of toxicity-related features. These results demonstrate that the features extracted by the proposed ToxIBTL can accurately capture the toxicity characteristics of peptides.

4 Conclusion

Accurate identification of the toxicity of peptides plays a vital role in the discovery and development of peptide-based drugs. Therefore, in this study, we propose a novel deep learning method, called ToxIBTL, for improving the prediction of peptide toxicity from unknown peptides, which learns the informative features from multiple aspects including evolutionary, graphical and statistical information. To develop an efficient prediction model, the information bottleneck principle is coupled with transfer learning to extract more discriminative features for peptide sequences. We first pre-train our model on the large protein dataset, and then fine-tune the pre-trained model on the small peptide dataset, which can effectively transfer common knowledge learned from proteins to peptides. Experimental results demonstrate that the proposed ToxIBTL achieves a significant prediction performance on the peptide dataset, and it is superior to other state-of-the-art methods. In addition, we also conduct a comparative study to evaluate the performance of our model on the protein dataset. Experimental results show that our model also can obtain competitive performance compared with other existing methods. In general, ToxIBTL is an efficient model for predicting the toxicity of both peptides and proteins. We anticipate that our predictor can help to select the desired peptides or proteins in a cost-effective and high-throughput way to accelerate drug discovery and development.

Although our model achieves a promising performance, there is still room for further improvement. For example, ToxIBTL trains the classifiers on the protein dataset and the peptide dataset independently. However, the model can be trained on these two datasets together to efficiently learn the common characteristic space from both proteins and peptides by simultaneously minimizing classification losses on them, which may further improve the prediction performance. In addition, there is a lack of interpretability in our model. The toxicity of peptides may result from certain residues located at the C- or N-terminal in a peptide sequence or concerted actions of multiple residues. Extracting such intrinsic relationships from the predictor can not only enrich the explanation, but also enhance our understanding of the mechanism of peptide toxicity. However, the solutions to solve these problems still need to be further explored.

Funding

This study was supported in part by the New Energy and Industrial Technology Development Organization (NEDO) grant [AJD30064], JST COI-NEXT, Grants-in-Aid for Scientific Research under grant [18H03250] and the Natural Science Foundation of China. [No. 62071278].

Conflict of Interest: The authors declare no competing financial interest.

References

Alemi,A.A. *et al.* (2016) Deep variational information bottleneck. Proceedings of the International Conference on Learning Representations (ICLR) 2017, Toulon, France, April 24-26, 2017.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Ansari,H.R. and Raghava,G.P. (2010) Identification of conformational B-cell Epitopes in an antigen from its primary sequence. *Immunome Res.*, **6**, 1–9.

Chen,F. *et al.* (2012) Extraordinary metabolic stability of peptides containing α-aminoxy acids. *Amino Acids*, **43**, 499–503.

Chu,Y. *et al.* (2021) DTI-CDF: a cascade deep forest model towards the prediction of drug-target interactions based on hybrid features. *Brief. Bioinform.*, **22**, 451–462.

Craik,D.J. *et al.* (2013) The future of peptide-based drugs. *Chem. Biol. Drug Des.*, **81**, 136–147.

El-Gebali,S. *et al.* (2019) The Pfam protein families database in 2019. *Nucleic Acids Res.*, **47**, D427–D432.

El-Manzalawy,Y. *et al.* (2008) Predicting linear B-cell epitopes using string kernels. *J. Mol. Recognit.*, **21**, 243–255.

Fosgerau,K. and Hoffmann,T. (2015) Peptide therapeutics: current status and future directions. *Drug Discov. Today*, **20**, 122–128.

Gentilucci,L. *et al.* (2010) Chemical modifications designed to improve peptide stability: incorporation of non-natural amino acids, pseudo-peptide bonds, and cyclization. *Curr. Pharm. Des.*, **16**, 3185–3203.

Gohil,D. and Thirugnanasambandan,T. Nanocarriers in protein and peptide drug delivery. In: Shah,N. (ed.) *Nanocarriers: Drug Delivery System*. Springer, Singapore, 2021. pp. 349–365.

Gupta,S. *et al.* (2013) In silico approach for predicting toxicity of peptides and proteins. *PLoS One*, **8**, e73957.

Gupta,S. *et al.* (2015) Peptide toxicity prediction. In: Zhou,P. and Huang,J. (eds) *Computational Peptidology*. Springer, New York, pp. 143–157.

Haggag,Y.A. *et al.* (2018) Peptides as drug candidates: limitations and recent development perspectives. *Biomed. J.*, **1**, 3.

Jones,P. *et al.* (2014) InterProScan 5: genome-scale protein function classification. *Bioinformatics*, **30**, 1236–1240.

Kawashima,S. and Kanehisa,M. (2000) AAindex: amino acid index database. *Nucleic Acids Res.*, **28**, 374.

Ke,G. *et al.* (2017) Lightgbm: a highly efficient gradient boosting decision tree. In: *Advances in Neural Information Processing Systems*. Curran Associates Inc., USA. Vol. **30**, pp. 3146–3154.

Kingma,D.P. and Welling,M. (2013) Auto-encoding variational Bayes. *Proceedings of the International Conference on Learning Representations (ICLR) 2014, Banff, AB, Canada, April 14-16, 2014*.

Li,C.-C. and Liu,B. (2020) MotifCNN-fold: protein fold recognition based on fold-specific features extracted by motif-based convolutional neural networks. *Brief Bioinform*., **21**, 2133–2141.

Li,D. *et al.* (2017) Protein remote homology detection based on bidirectional long short-term memory. *BMC Bioinformatics*, **18**, 1–8.

Liu,B. *et al.* (2015) Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res*., **43**, W65–W71.

Liu,B. *et al.* (2020) DeepSVM-fold: protein fold recognition by combining support vector machines and pairwise sequence similarity scores generated by deep learning networks. *Brief Bioinform*., **21**, 1733–1741.

Manavalan,B. *et al.* (2019a) AtbPpred: a robust sequence-based prediction of anti-tubercular peptides using extremely randomized trees. *Comput. Struct. Biotechnol. J*., **17**, 972–981.

Manavalan,B. *et al.* (2019b) mAHTPred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. *Bioinformatics*, **35**, 2757–2765.

Mu,Z. *et al.* (2021) FEGS: a novel feature extraction model for protein sequences and its applications. *BMC Bioinformatics*, **22**, 1–15.

Mumtaz,M.M. and Pohl,H.R. (2012) Interspecies uncertainty in molecular responses and toxicity of mixtures. In: Andreas,A. (ed.) *Molecular, Clinical and Environmental Toxicology*. Springer, Basel, pp. 361–379.

Muttenthaler,M. *et al.* (2021) Trends in peptide drug discovery. *Nat. Rev. Drug Discov*., **20**, 309–325.

Naamati,G. *et al.* (2009) ClanTox: a classifier of short animal toxins. *Nucleic Acids Res*., **37**, W363–W368.

Negi,S.S. *et al.* (2017) Functional classification of protein toxins as a basis for bioinformatic screening. *Sci. Rep*., **7**, 1–11.

Pan,X. *et al.* (2020) ToxDL: deep learning using primary structure and domain embeddings for assessing protein toxicity. *Bioinformatics*, **36**, 5159–5168.

Potter,S.C. *et al.* (2018) HMMER web server: 2018 update. *Nucleic Acids Res*., **46**, W200–W204.

Saha,S. and Raghava,G.P.S. (2006) Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. *Proteins*, **65**, 40–48.

Sato,K. *et al.* (2021) RNA secondary structure prediction using deep learning with thermodynamic integration. *Nat. Commun*., **12**, 1–9.

Shapiro,A. (2003) Monte Carlo sampling methods. In: Ruszczynski,A. and Shapiro,A. (eds) *Handbooks in Operations Research and Management Science*. North Holland Publishing Co., Amsterdam. vol. **10**, pp. 353–425.

Singh,J. *et al.* (2021) Improved RNA secondary structure and tertiary base-pairing prediction using evolutionary profile, mutational coupling and two-dimensional transfer learning. *Bioinformatics*, **37**, 2589–2600.

Su,R. *et al.* (2020) Empirical comparison and analysis of web-based cell-penetrating peptide prediction tools. *Brief. Bioinform*., **21**, 408–420.

Tan,J.-X. *et al.* (2019) Identification of hormone binding proteins based on machine learning methods. *Math. Biosci. Eng*., **16**, 2466–2480.

Tishby,N. *et al.* (2000) The information bottleneck method. *arXiv preprint physics/0004057*.

Van der Maaten,L. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res*., **9**, 2579–2605.

Wei,L. *et al.* (2021) ATSE: a peptide toxicity predictor by exploiting structural and evolutionary information based on graph neural network and attention mechanism. *Brief Bioinform*., **22**, bbab041.

Ye,X. *et al.* (2020) Detecting interactive gene groups for single-cell RNA-Seq data based on co-expression network analysis and subgraph learning. *Cells*, **9**, 1938.

Zeng,X. *et al.* (2020) Network-based prediction of drug–target interactions using an arbitrary-order proximity embedded deep forest. *Bioinformatics*, **36**, 2805–2812.

Zhuang,F. *et al.* (2021) A comprehensive survey on transfer learning. *Proc. IEEE*, **109**, 43–76.