OXFORD

Briefings in Bioinformatics, 22(6), 2021, 1–13

https://doi.org/10.1093/bib/bbab203
Problem Solving Protocol

PSSP-MVIRT: peptide secondary structure prediction based on a multi-view deep learning architecture

Xiao Cao, Wenjia He, Zitan Chen, Yifan Li, Kexin Wang, Hongbo Zhang, Lesong Wei, Lizhen Cui, Ran Su and Leyi Wei

Corresponding authors: Leyi Wei, School of Software, Shandong University, Jinan, China; Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan, China; E-mail: weileyi@sdu.edu.cn; Ran Su, College of Intelligence and Computing, Tianjin University, Tianjin, China. E-mail: ran.su@tju.edu.cn

Abstract

The prediction of peptide secondary structures is fundamentally important to reveal the functional mechanisms of peptides with potential applications as therapeutic molecules. In this study, we propose a multi-view deep learning method named **P**eptide **S**econdary **S**tructure **P**rediction based on **M**ulti-**V**iew **I**nformation, **R**estriction and **T**ransfer learning (PSSP-MVIRT) for peptide secondary structure prediction. To sufficiently exploit discriminative information, we introduce a multi-view fusion strategy to integrate different information from multiple perspectives, including sequential information, evolutionary information and hidden state information, respectively, and generate a unified feature space. Moreover, we construct a hybrid network architecture of Convolutional Neural Network and Bi-directional Gated Recurrent Unit to extract global and local features of peptides. Furthermore, we utilize transfer learning to effectively alleviate the lack of training samples (peptides with experimentally validated structures). Comparative results on independent tests demonstrate that our proposed method significantly outperforms state-of-the-art methods. In particular, our method exhibits better performance at the segment level, suggesting the strong ability of our model in capturing local discriminative information. The case study also shows that our PSSP-MVIRT achieves promising and robust performance in the prediction of new peptide secondary structures. Importantly, we establish a webserver to implement the proposed method, which is currently accessible via http://server.malab.cn/PSSP-MVIRT. We expect it can be a useful tool for the researchers of interest, facilitating the wide use of our method.

Xiao Cao is currently an undergraduate student in the School of Software at Shandong University, China. His interests are machine learning, bioinformatics and computer vision.

Wenjia He is currently an undergraduate student in the School of Software at Shandong University, China. His interests are deep learning, meta-learning, few-shot learning and their application in bioinformatics.

Zitan Chen is currently an undergraduate student in the School of Software at Shandong University, China. Her interests include machine learning, bioinformatics and virtual reality.

Yifan Li is currently an undergraduate student in the School of Software at Shandong University, China. His interests are machine learning and bioinformatics.

Kexin Wang is currently an undergraduate student in the School of Software at Shandong University, China. Her interests are image processing, computer vision and machine learning.

Hongbo Zhang is currently an undergraduate student in the School of Software at Shandong University, China. His interests are machine learning and virtual reality.

Lesong Wei received the BS degree in Computer Science and Technology from the Taiyuan University of Technology, China, in 2017, the MS degree in Computer Technology from Fuzhou University, China, in 2021. He currently is a PhD student at the University of Tsukuba, Japan. His research interests are bioinformatics and machine learning.

Lizhen Cui is currently a Professor with School of Software, Shandong University, the Deputy Director of the E-Commerce Research Center and the Director of the Research Center of Software and Data Engineering, Jinan. He is currently involved in big data integration and intelligent analytics. He is an Academic Leader of the Outstanding Innovation Team. He has received nearly 20 research grants at national, provincial and ministerial levels. He has published more than 60 high-level academic articles in TPDS, TSC, TCC, Chinese Journal of Computers, Scientific Data, AAAI, SIGIR, CIKM, BIBM, ICDCS, DASFAA, ICWS, ICSOC and AAMAS.

Ran Su is an Associate Professor at the College of Intelligence and Computing, Tianjin University, Tianjin, China. Her research interests are bioinformatics and machine learning.

Leyi Wei received his PhD in Computer Science from Xiamen University, China. He is currently a Professor in School of Software at Shandong University, China. His research interests include machine learning and its applications to bioinformatics.

Submitted: 28 February 2021; **Received (in revised form):** 21 April 2021

Key words: peptide secondary structure; transfer learning; multi-view feature fusion; feature representation

Introduction

Peptides have recently emerged as potential therapeutic molecules against various diseases for their high specificity, high tolerance, high penetration, few side effects, low production cost and ease in manufacturing and modifications [1]. The biological functions of peptides are closely related to their structures. Therefore, understanding the structures of bioactive peptides is not only helpful in further understanding of peptide functions but also guides the designing of peptides with desired functions [2]. Secondary structure refers to the 3D local segments of the protein macromolecule that forms after the amino acid residues join in a sequence and before the protein folds into its tertiary structure. The secondary structure involves hydrogen bonds along the backbone that cause the long chain to fold into local shapes, mainly helices (H), strands (E) and coils (C) [3]. Subsequently, before predicting peptide tertiary structures, an important step is to determine the secondary structures of peptides, which can provide information regarding binding characteristics and backbone that are useful for tertiary structure prediction.

In the past few years, several computational methods have been proposed for predicting protein secondary structures based on machine learning. For instance, Jones [4] designed a two-stage neural network trained with evolutionary features derived from position-specific scoring matrices (PSSM), which is a kind of profile containing sufficient evolutionary conservation information. Zhou and Troyanskaya [5] proposed a new supervised generative stochastic network-based method that learns a Markov chain from a conditional distribution and applied it to protein structure prediction. Later on, Wang *et al*. [6] proposed Deep Convolutional Neural Fields, a deep learning model that can explore not only the complex sequence–structure relationship but also interdependency between adjacent property. Particularly, unlike other previous methods, it can provide more accurate secondary structure prediction for proteins without close homolog or with little evolutionary information. Similarly, Li and Yu [7] presented Diffusion Convolutional Recurrent Neural Network (DCRNN), an end-to-end deep neural network that focuses on using both global features and local features and utilizes multi-task learning to predict the secondary structure and amino acid solvent accessibility simultaneously. To capture the long-distance dependency along with proteins, Heffernan *et al*. [8] designed Spider3, a bidirectional recurrent neural network with a long short-term memory mechanism, aiming to extract the global features. They demonstrated that Spider3 outperforms other previous methods. Busia and Jaitly [9] proposed the next-step conditioned deep Convolutional Neural Network (CNN), which improved upon state-of-art by using a novel chained prediction approach. The neural network frames the secondary structure prediction as a next-step prediction problem. More recently, Fang *et al*. [10] developed a deep inception-inside-inception network (namely Deep3I) that integrates various information like physiochemical properties of amino acids, and evolutionary information derived from the PSI-BLAST profile (PSSM) to train the predictive model. Deep3I enables effective processing of local and global interactions between each residue in making accurate predictions. Besides the methods introduced above, there are other outstanding protein secondary structure prediction methods, such as PSIPRED [11], Jpred [12], RaptorX [13], PHD [14], PROTEUS2* [15], etc.

However, the methods mentioned above are designed specifically for protein secondary structure prediction, there are many differences in secondary structure between protein and peptide. On the one hand, previous studies have demonstrated that by comparing secondary structure composition of peptides and proteins, their secondary structures are different [2] for some identical segments of residues in proteins and peptides. On the other hand, lacking accurate peptide secondary structures also limits the prediction of peptide functions, like anti-cancer activity [16], which heavily relies on sequential information. Thus, it is reasonable to expect an improvement by integrating extra secondary structure information. To deal with this problem, Singh *et al*. [2] first proposed a Random Forest-based method namely PEP2D, which predicts the peptide secondary structures with sequential and evolutionary information and gained a lot of improvement by exploiting secondary structure information. In summary, secondary structure prediction of peptides is of great significance for downstream structural or functional prediction.

In this study, we proposed a novel deep learning neural network called **P**eptide **S**econdary **S**tructure **P**rediction based on **M**ulti-**V**iew **I**nformation, **R**estriction and **T**ransfer Learning (PSSP-MVIRT), which is designed specifically for peptide secondary structure prediction. The novelty of the proposed PSSP-MVIRT can be concluded as the following three aspects. First, to sufficiently exploit discriminative information, we used a multi-view fusion strategy to integrate the information from multiple perspectives, including sequential information, evolutionary information and hidden state information, respectively. Second, to extract global and local features of peptides, we used a hybrid network architecture of CNN [28] and Bi-directional Gated Recurrent Unit (BGRU). Particularly, we introduce an additional restriction mechanism that can capture high-latent feature representations and improve the representation ability. Third, due to the lack of training samples with experimentally validated structures, we here utilized transfer learning to train our model on a large-scale protein dataset first and then fine-tuned the model for peptide secondary structure prediction. Extensive comparative experiments on benchmark datasets demonstrate that our proposed method significantly outperforms state-of-the-art methods on the independent test. More importantly, via comparative analysis, we show that our method can capture more local informative characteristics of peptides, which can effectively help to improve the predictive performance.

Methods and materials

Datasets

Initial dataset collection

In this study, we used the same benchmark dataset, namely SCRATCH-1D, which is commonly used for performance evaluation in several studies [17]. This dataset consists of 5772 primary and corresponding secondary structures of protein data with three structural states (H, E and C). In SCRATCH-1D, the protein structures are derived with X-ray crystallography with

Table 1. Summary of the datasets used in this work

Datasets	Structural states			Sequence number
	H	E	C	
Segmented protein training set	321,476	206,758	340,032	9262
Peptide training set	38,749	18,020	32,910	1285
Peptide testing set	5294	1119	3733	257

a resolution of at least 2.5 angstroms, with no chain breaks, with less than five unknown amino acids and of length at least 30 residues. Notably, the sequence identity in the dataset is reduced to 25% to avoid the bias of performance evaluation. However, we found that there are some proteins with unnatural residues represented by the symbol X. After removing these peptides, 4542 protein and peptide sequences are retained in our dataset.

Training and testing dataset

Since our task is to predict the secondary structures of peptides, whose samples are normally <100 residues long, the protein sequences with the length >100 residues long in the dataset are segmented to 100 residues long, rather than using full-length protein sequences. By doing so, we yielded 9262 segmented protein subsequences in total. The reason for doing so is to better capture the characteristics of short peptide-like sequences so as to achieve better performance. All the segmented protein subsequences are used for pretraining an initial deep learning-based predictive model, whereas the peptide sequences are used for model fine-tuning to generate a task-specific model. For peptide model training phase, we randomly selected 1028 out of 1285 peptide sequences as training dataset, of which the number of the three structural states H, E and C are 38 749, 18 020 and 32 910, respectively (Table 1). The remaining 257 peptide sequences (with H of 7450, E of 4199 and C of 6957) are chosen as our test set used for model performance evaluation. The sequence length of each peptide is between 30 and 100 residues that are labeled with three-state secondary structure. The statistics of the three-state secondary structure and peptide sequence are shown in Figure 1A and C, in which the length of each color in each FASTA file, respectively, represents the amount of helix (H), strand (E) or coil (C). Figure 1B shows the number of corresponding peptide sequences in response to the specified amount of each state in the dataset. The detailed information of the datasets used in this work can be seen in Table 1. Figure 1 also illustrates the statistics of the datasets.

The architecture of the proposed PSSP-MVIRT

Figure 2 illustrates the architecture of the proposed neural network, namely, PSSP-MVIRT. The method contains four major modules: (1) multi-view feature embedding, (2) feature extraction, (3) feature representation ability enhancement and (4) prediction module. The prediction procedure is described as follows. In Module (1), given a peptide sequence, it is first encoded into three feature metrics, which represent sequential information, evolutionary information and hidden state information, respectively. Afterward, to learn a unified feature embedding, we use Cosine Similarity based on a multi-view fusion strategy to measure how similar two embedded features are. In Module (2), to further exploit more discriminative information, we utilized a hybrid neural network of CNN and BGRU, capturing the local features and global features. In Module (3), we employed the Transformer Encoder [18], a widely used Natural Language Processing technique, to enhance the feature representation derived from the last step. Finally, in Module (4), resulting features are fed into our model to predict each position of the peptide belonging to which structural state: C, H or E. The four modules are introduced in detail below.

Multi-view feature fusion module

In this section, we introduce how to preprocess our raw peptide sequences into numeric feature representations, which can be trained with a machine learning algorithm. Below, we first introduce the embedding approaches from the following four feature views: evolutionary information, sequential information, hidden state information and similarity information. Next, to generate a unified feature space, we adopt a multi-view feature fusion and learning strategy.

Feature View 1—sequential information embedding

The sequential information here is generated by word2vec [19] from a list of indices. Comparing with one-hot coded sequential information, it can be used to learn high-quality residue vectors with latent semantic and prevent the zero-redundant expression.

Feature View 2—evolutionary information embedding

PSSM is an $m*n$ matrix, where m is the length of each protein sequence and n is the number of standard residues. PSSM scores are usually shown as positive or negative integers. In this way, we can compute the position-specific scores of the 20 amino acids in a specific position of the sequence. The lower-scored amino acids have a great tendency to evolve into the higher-scored amino acids, which maintain a stable state. In this study, the PSSM of each peptide sequence was generated by three iterations of Position-Specific Iterated-Basic Local Alignment Search Tool (PSI-BLAST)+ [20] against the SwissProt database [21] (version updated on 5 September 2020) with default parameters.

Feature View 3—hidden state information embedding

The Hidden Markov Model (HMM) serves as a type of stochastic model. And it is widely used for predicting protein secondary structure. In peptide secondary structure prediction, structures such as H (helices), E (strands) and C (coils) are learned by HMMs, and these HMMs are applied to new peptide sequences whose secondary structures remain unknown. The output of probabilities from the HMMs is used to predict the secondary structures of sequences [22]. In this study, the explicit sequences are peptide sequences and the hidden states are their secondary structures. The HMM profiles we used in the study were generated from HMMER3.0 [23].

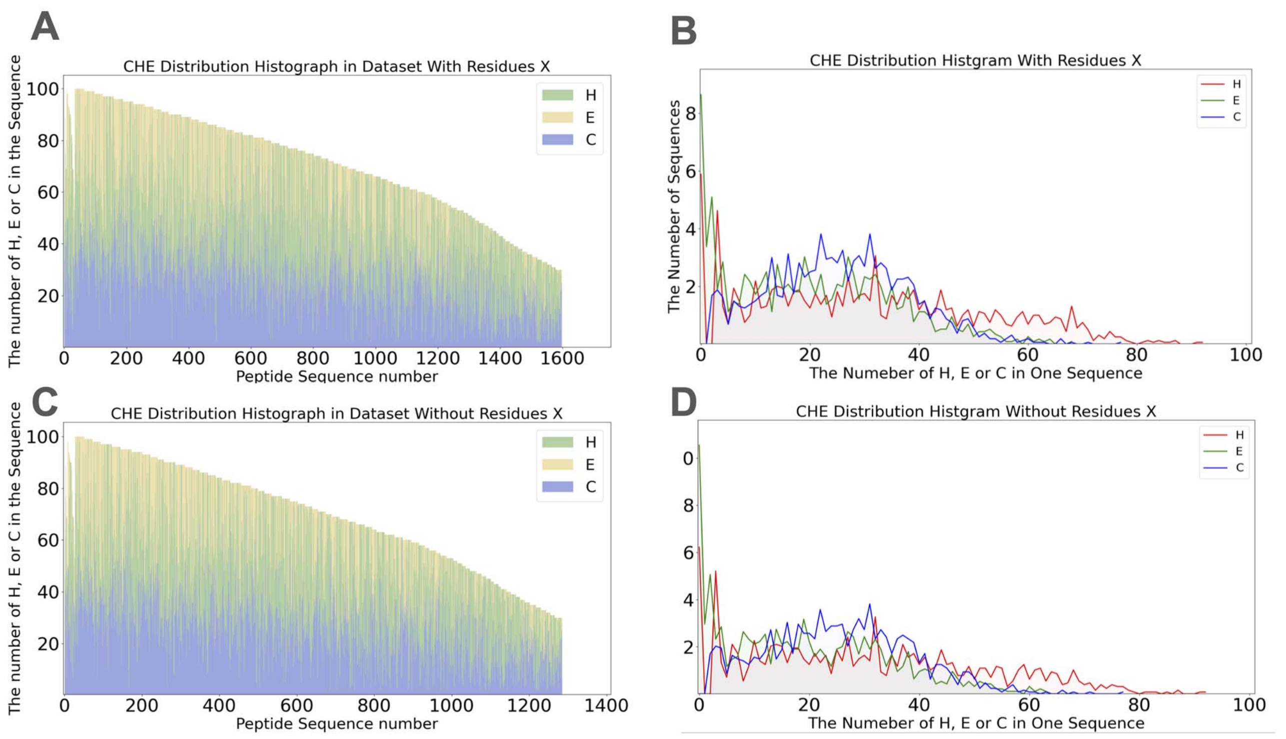


Figure 1. Statistics of peptide dataset. (**A**) The number of each secondary structure in each peptide sequences with residues *X*; (**B**) the number of corresponding peptide sequences with residues *X* in response to the specified amount of HEC; (**C**) the number of each secondary structure in each peptide sequences without residues *X*; (**D**) the number of corresponding peptide sequences without residues *X* in response to the specified amount of HEC.

Multi-view feature fusion strategy

We here used the cosine similarity to generate a unified feature representation space by fusing the information from the above three feature views. Given a benchmark dataset with n sequences $\{P, E\}$, where P represents **PSSM** and E represents embedded sequential information. For each given peptide, its feature can be denoted as a matrix X as below:

$$\begin{cases} x = \text{cosine}\,(P, E) & (1) \\ \text{cosine}\,(P, H) = \frac{P \cdot E}{\|P\|\,\|E\|} & (2) \\ \|P\|^2 = Tr\left(P^T P\right) & \\ P^T P = \begin{pmatrix} p_{11}^T p_{11} & K & p_{1n}^T p_{11} \\ M & O & M \\ p_{m1}^T p_{11} & L & p_{mn}^T p_{11} \end{pmatrix} & (3) \\ Tr\left(P^T P\right) = p_{11}^T p_{11} + K + p_{nn}^T p_{nn} & (4) \end{cases}$$

To simplify the integration process, the matrix 1-norm is replaced by infinite-norm, as shown below. Moreover, we also integrate some supplementary information, which consists of two parts: (1) similarity information generated by PSSM information and embedded sequential information; and (2) similarity information generated by HMM information and embedded sequential information. The process of generating unified feature representation space of Hidden Markov Model (**HMM**) and embedded sequential information is the same as above.

$$\begin{cases} \text{cosine}\,(P, E) = \frac{P \cdot E}{\|P\|_\infty \|E\|_\infty} & (5) \\ \|P\|_\infty = \max\left(\sum_{i=1}^n \left|p_{1j}\right|, \sum_{i=1}^n \left|p_{2j}\right|, \cdots\cdots, \sum_{i=1}^n \left|p_{1nj}\right|\right) & (6) \end{cases}$$

where $\|P\|_\infty$ is the infinite-norm of matrix P, and $\|H\|_\infty$ is the infinite-norm of matrix H.

Finally, HMM, PSSM and the two unified feature representation spaces are concatenated to an $m \times w$ matrix as the high-latent input feature, where m is the length of peptide length and w is the sum of width of HMM, PSSM and the two supplementary information.

The high-latent feature extraction module

For high-latent feature extraction, we utilized a hybrid neural network of CNN and BGRU, in which the CNN is to extract local features and the BGRU is to extract global features.

Local feature extraction using CNN with novel padding techniques

Here, we leverage CNN to learn and extract local features. Each convolutional neuron processes data only for its receptive field. Thus, CNN is employed here to extract local information in peptide feature representations. Notably, padding techniques (Cyclic padding and Reflection padding) that are discussed in support information are used before each 2D-convolutional layer, as shown in Figure 3. By using the padding techniques, we can effectively solve the boundary information extraction problem for peptide chains, improving the predictive performance on the terminals of each peptide chain. More details of the padding techniques are introduced in Supplementary Material.

Global feature extraction using BGRU

The global feature extraction can be divided into two parts by the additional restriction. In the first global feature extraction part, the fully connected layers are used as the transition layers

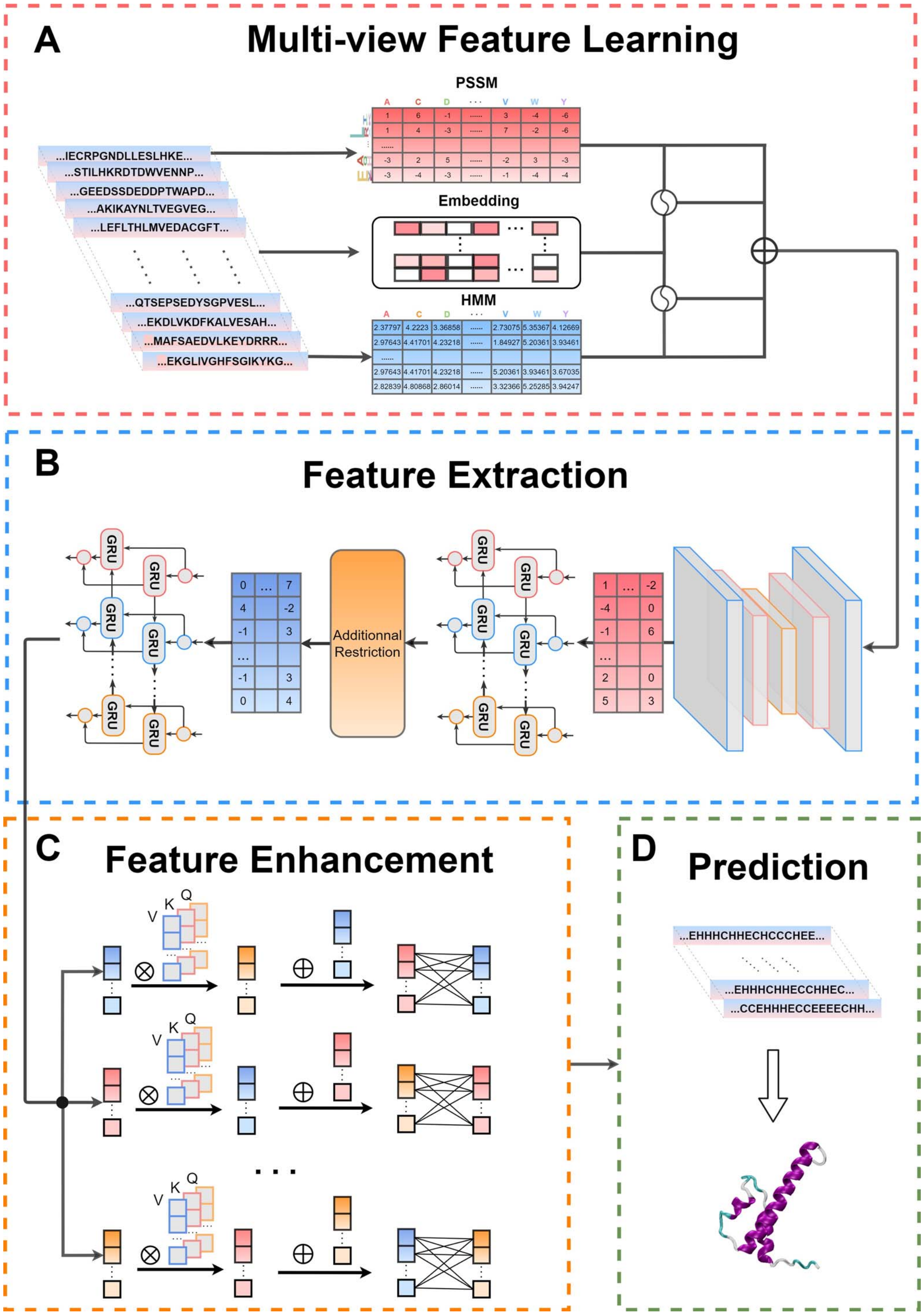


Figure 2. Architecture of PSSP-MVIRT. (A) Peptides are first encoded by four kinds of feature representation approaches to explore different sequential information, which are then integrated into a feature matrix by concatenation (B) CNN with padding techniques to extract local features and Parallel BGRU to extract local–global features at segmental level; (C) the resulting features are enhanced by multi-head attention mechanism; (D) the secondary structures of the peptides are predicted by our well-trained model and visualized by PyMol, a tool specific for secondary structure visualization.

between the local feature extraction part and the global feature extraction part. Then, the BGRU receives a more effective feature matrix and extracts the long-distance dependency further. In the second global feature extraction part, the fully connected layers are inserted behind the BGRU layer, as shown in Figure 2. In this paper, we also explored whether it performs better to divided peptides into multi-subsequences as input of BGRUs, which was named Parallel BGRU, whose architecture is shown in Figure 4. The comparison experiment about different-levels of Parallel BGRU architectures is discussed in Section 'Determination of the optimal network architecture of our model'.

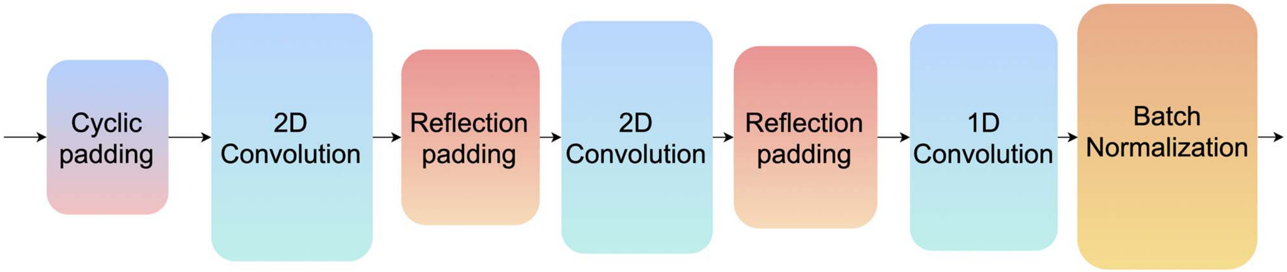


Figure 3. The local features extraction part in Feature Extraction Module.

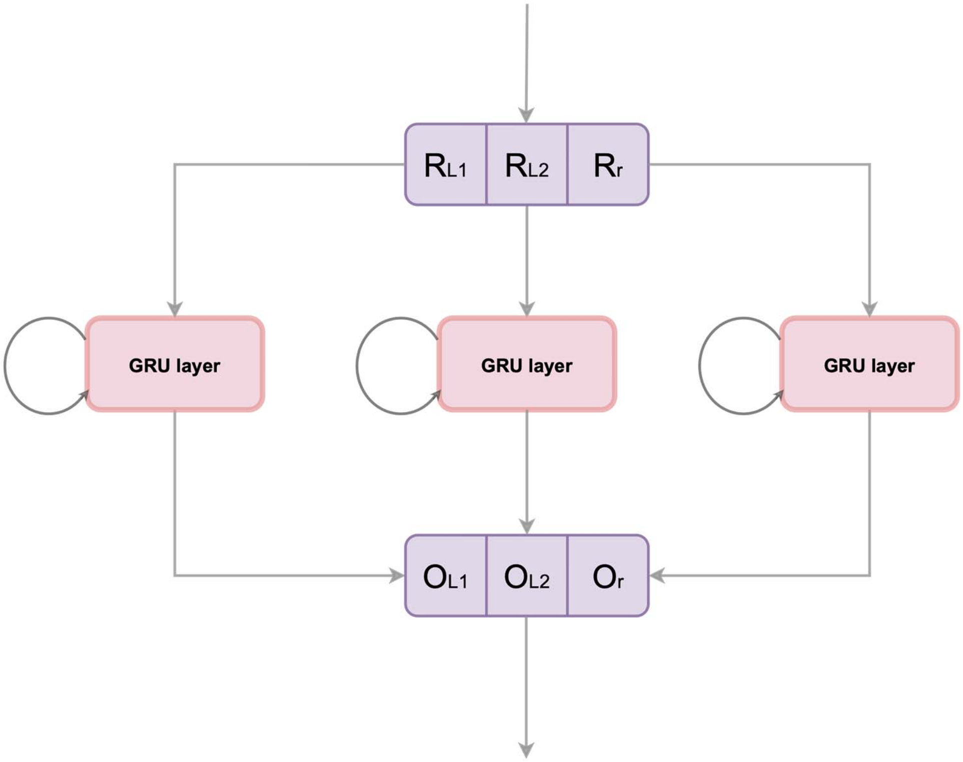


Figure 4. The architecture of Parallel BGRU.

Gated recurrent unit (GRU)

GRU [24] performs well in solving the vanishing gradient problem of standard Recurrent Neural Network (RNN). GRU allows each recurrent unit to capture the dependency of different time scales adaptively, as shown in Figure 5. It is easier for each unit to remember the existence of a specific feature in the input stream over a long series of time steps with GRUs.

$$\begin{cases} zt = \sigma\left(Wz \cdot \left[h\left(t-1\right), xt\right]\right) & (7)\\ rt = \sigma\left(Wr \cdot \left[h\left(t-1\right), xt\right]\right) & (8)\\ \tilde{ht} = \left(W \cdot \left[rt * h\left(t-1\right), xt\right]\right) & (9)\\ ht = (1-zt) * h\left(t-1\right) + zt * \tilde{ht} & (10)\\ \sigma(x) = \frac{1}{1+e^{-x}} & (11) \end{cases}$$

where *zt* is defined as an update gate controlling the degree to which the state information of the previous time is brought into the current state; *rt* is defined as a reset gate, which controls how much information is written to the candidate activation $\tilde{ht}$ from the previous state.

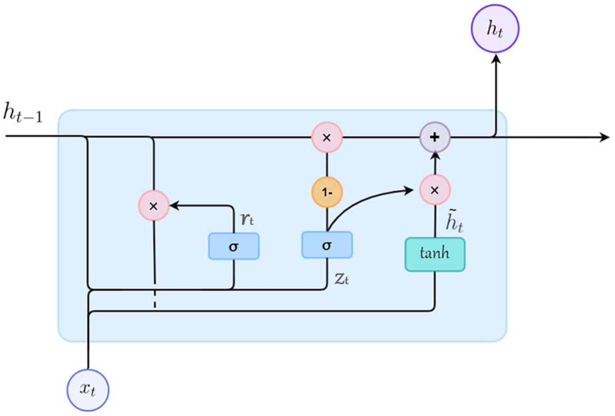


Figure 5. The architecture of BGRU.

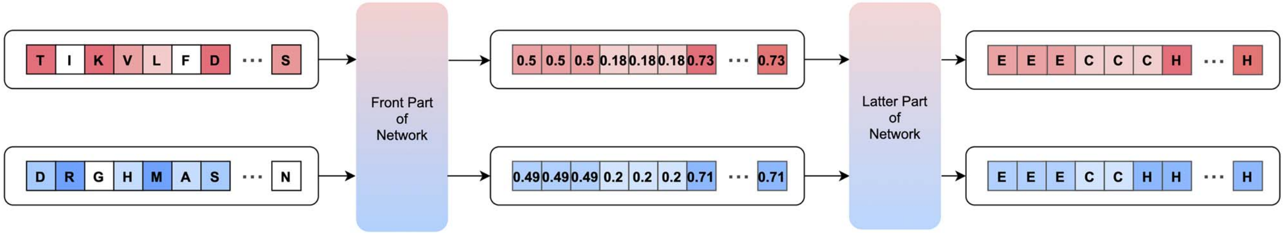


Figure 6. Example of the relationship between the feature representations in the middle of neural network and the prediction results.

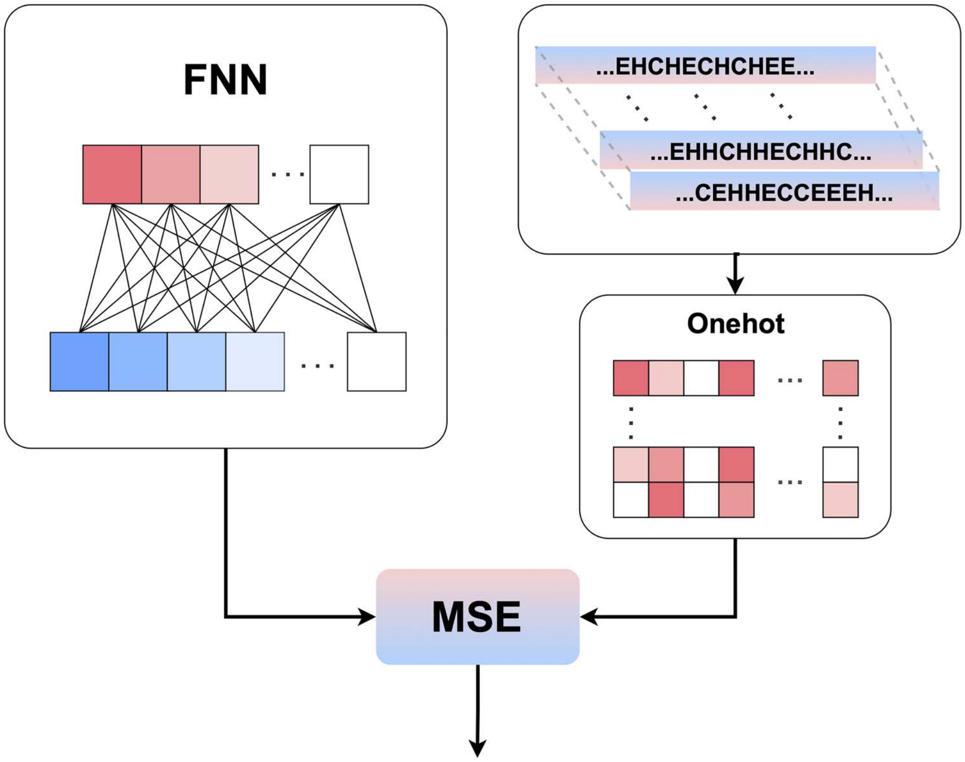


Figure 7. Additional restriction mechanism in PSSP-MVIRT.

Additional restriction mechanism

Inspired by Wang *et al.* [25], the additional restriction mechanism is employed in our model. As shown in Figure 6, when the secondary structure of pepide sequences is the same, it should have similar representations in the middle of the neural network if it receives different peptide sequences. For that reason, an additional restriction mechanism is inserted between the global features extraction parts. The additional restriction consists of fully connected layers that are used to reshape the transient state. It receives the output of the first global features extraction part as an input feature. After the fully connected layers, an additional loss is calculated by the mean square error as the cost function using the secondary structure labels and the outputs of the fully connected layer, as shown in Figure 7.

Feature representation ability enhancement module

This part mainly consists of the six-stacked eight-head Transformer Encoder [18]. The output of the feature extraction part is received as the embedded input, which could be a more effective representation way than the word embedding. It processes the high-level feature by feeding these vectors into a self-attention layer and then into a feed-forward neural network, and finally sends out the output to the next Transformer Encoder block. After the processing of the Transformer Encoder, it comes to two fully connected layers, which receive the attention feature and output the secondary structure labels.

Multi-head attention mechanism

The concept of ‘attention’ has gained popularity recently in training neural networks, especially in translating and aligning words, which is similar to peptide secondary structure prediction, for it can flexibly catch global and local dependency. In model design, we follow the original Transformer Encoder part as closely as possible, which works as the main feature enhancement part.

It is proved that linearly projecting of the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively, outperform a single attention function with d_{model}-dimensional keys, values and queries. Each Transformer Encoder block includes a Scaled Dot-Product Attention layer and full connection with residual connection mechanism. The overall multi-head attention mechanism is shown as:

$$\begin{cases} \text{softmax}(x) = \frac{\exp(x)}{\sum_t \exp(x)} & (12) \\ \text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK}{\sqrt{d_k}}\right) V & (13) \\ \text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) & (14) \\ \text{MultiHead} = \text{Concat}\,(\text{head}_1, \ldots, \text{head}_i) & (15) \end{cases}$$

where Q denotes queries matrices; K denotes key matrices; V denotes value matrices; and W^Q, W^K, W^V denote the trained weight matrices, respectively.

Prediction module

For training a robust predictive model, we here construct a new loss function, which is composed of the following two cost functions: (1) the mean square error middle restriction function and (2) the weighted mean square error loss function, as shown below. To balance the two cost functions, a balance coefficient γ is used to calculate a final cost for the optimizer, as shown below:

$$\begin{cases} \text{Loss}\,(\text{MSE}_1) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(m_i^j - l_i^j\right) & (16) \\ \text{Loss}\,(\text{MSE}_2) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(y_i^j - w_i^j\right) & (17) \\ \mathbf{Loss}\,(\mathbf{Final}) = \boldsymbol{\gamma}\,\mathbf{Loss}\,(\text{MSE}_1) + \mathbf{Loss}\,(\text{MSE}_2) & (18) \end{cases}$$

where MSE is the acronym of mean square error, m is defined as the output of the additional restriction part, l is defined as the secondary structure label coded by the one-hot encoder, y is defined as the output of the feature extraction module, w is defined as the weighted coded label, which the weight of state E is 1.25 and others are 1, m is the sample number and n is the peptide sample length without zero paddings.

Performance metrics

In this study, the performance of the PSSP-MVIRT is measured by the accuracy of prediction in each structural state Acc_i (Acc_H, Acc_E, Acc_C), the accuracy of prediction in all states namely Acc, precision in each structural state and segment overlap measure [26] (Sov). The metrics are computed as follows:

$$\begin{cases} \text{Acc}_i = \frac{A_{ii}}{A_i} & (19) \\ \text{Acc} = \sum \alpha_i \frac{\sum_{i \in \{H,E,C\}} A_{ii}}{\sum_{i \in \{H,E,C\}} A_i} & (20) \\ \text{Sov} = \frac{\sum_{i \in \{H,E,C\}} \sum_{si} \frac{\min \text{ov}(s1,s2) + \delta(s1,s2)}{\max \text{ov}(s1,s2)} \times \text{len}(s1)}{N} & (21) \end{cases}$$

where i is any secondary structure element (Helix, Sheet or Coil); A_i is the total number of correctly predicted residues in each state; A_{ii} is the number of correctly predicted residues in the state i; α_i is the proportion of state i in the whole test set; $s1$ and $s2$ are segments corresponding to actual and predicted secondary structure; len($s1$) corresponds to the number of residues defining the segment $s1$; min ov($s1, s2$) corresponds to the length of overlapping $s1$ and $s2$ segments; max ov($s1, s2$) is the maximum overlap of s1 and $s2$ segments for which either of the segments has a residue in state i; $\delta(s1, s2)$ is computed below:

$$\delta\,(s1, s2) = \min \begin{cases} \left(\max \text{ov}\,(s1, s2) - \min \text{ov}\,(s1, s2)\right) & (22) \\ \left(\min \text{ov}\,(s1, s2)\right) & (23) \\ \left(\frac{\text{int}(\text{len}(s1))}{2}\right) & (24) \\ \left(\frac{\text{int}(\text{len}(s2))}{2}\right) & (25) \end{cases}$$

Experimental settings

To get better performance and accelerate the training of the network, the batch normalization and dropout techniques are employed in PSSP-MVIRT. The dropout layers with ignoring rate p of 0.25 are inserted between each layer, except for the additional restriction part. As for batch normalization, it is inserted between (1) the input and features fusion part, (2) features fusion part and local features extraction part, (3) local features extraction part and the first global features extraction part, (4) the first global features extraction part and the second global features extraction part, and (5) the first global features extraction part and additional restriction part. It is thought to have the ability to reduce the internal covariate shift by adding network layers that control the means and variances of the layer inputs [27].

When it comes to the additional cost function, the secondary structure labels are encoded by the one-hot encoder and the balance coefficient γ is set to 0.1. In PSSP-MVIRT neural network, the Rectified Linear Unit (ReLU) activation function is used for all the convolutional layers, BGRU layers and some of the fully connected layers. The activation function sigmoid is used before the final fully connected layer in the additional restriction part, and the activation function softmax is used before the final fully connected layer in the second global features extraction part.

Our deep learning models with 31 744 430 parameters overall were trained globally by *Adam* algorithm with learning rate $l = 1e - 4$ to minimize the cost function Loss (Final). The training epoch is set to 250 and it performs best in the 97 epoch (Supplementary Figure S4). All the training and testing procedures were performed based on Nvidia Titan RTX GPUs and were implemented by python based on PyTorch.

Results and discussion

Comparison with existing secondary structure prediction methods

To evaluate the effectiveness of our proposed PSSP-MIRVT, we compared it with existing popular protein secondary structure prediction methods like PHD [16] and Jpred [14] on the same independent test set for fair. Notably, as for our prediction method, we trained three different weighted models with different structural state weights to avoid the data imbalance problem. The evaluation results are listed in Table 2. It shows that different weighted prediction models of our PSSP-MIRVT perform well among the five models, and the one whose E-state weight of 1.25 achieves the best performance with Acc of 78.50%, Acc_H of 90.16%, Acc_E of 56.84%, Acc_C of 68.47% andSov of 75.81%, respectively. We observed that our peptide-specific methods perform significantly better than those protein designed methods, especially on the Sov in peptide secondary prediction, which indicates that the methods designed for protein secondary structure prediction cannot

Table 2. Evaluation results of our proposed PSSP-MVIRT and existing methods on independent test set

Methods	Observedj	Predicted			ACCj	ACC	SOV
		H	E	C	(%)	(%)	(%)
PHD	H	4282	225	787	80.88	76.31	57.89
	E	72	566	481	50.58		
	C	478	355	2900	77.68		
JPRED	H	4195	146	953	79.24	78.05	60.62
	E	64	588	467	52.54		
	C	337	259	3136	84.03		
Our(1.00)	H	4760	62	472	89.91	77.64	75.06
	E	174	489	456	43.70		
	C	847	256	2630	70.45		
Our (1.25)	**H**	**4773**	**99**	**422**	**90.16**	**78.50**	**75.81**
	E	**139**	**636**	**344**	**56.84**		
	C	**836**	**341**	**2556**	**68.47**		
Our(1.50)	H	4608	167	519	87.04	77.40	84.01
	E	89	670	360	59.87		
	C	755	402	2576	69.01		

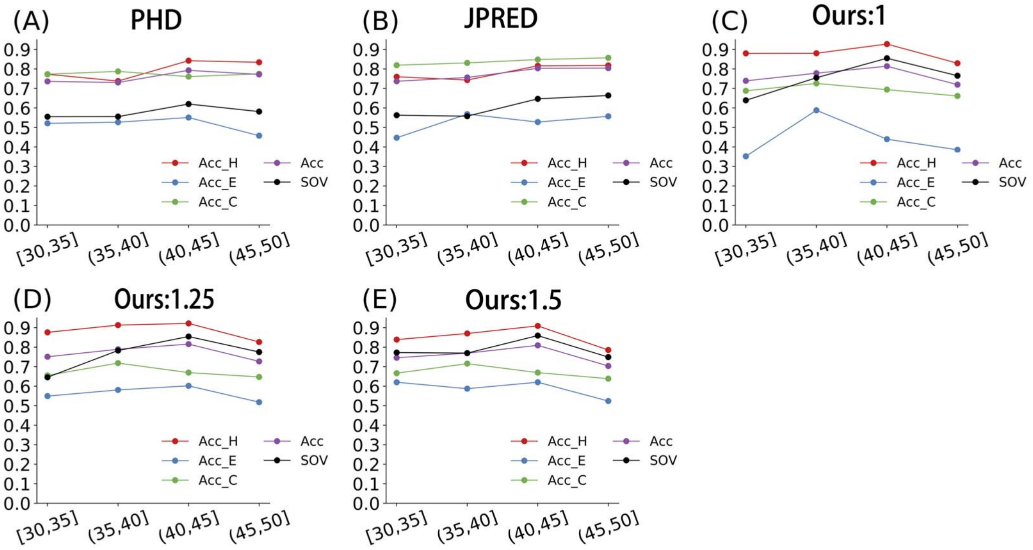


Figure 8. The performances of our method and existing methods on three test subsets with different length intervals: (A) performance of PHD; (B) performance of Jpred; (C) performance of our model with E-state weight 1; (D) performance of our model with E-state weight 1.25; (E) performance of our model with E-state weight 1.5.

sufficiently capture the discriminative information of short peptide sequences and proves the necessity of PSSP-MIRVT designed specifically for peptides. As compared with the PHD, our model is superior to PHD in almost all metrics, achieving 2.19, 9.28, 6.26 and 17.92% higher performance in terms of Acc, Acc_H, Acc_E, Sov, respectively. As compared with Jpred, our model outperforms Jpred in almost all metrics with 0.45, 10.92, 4.30 and 15.19% higher in terms of Acc, Acc_H, Acc_E and Sov, respectively. As seen, unlike Jpred and PHD having good Acc but poor Acc_H and Acc_E, our model not only achieves competitive Acc, Acc_E, Acc_C, but also reaches a pretty good Acc_H with >10% higher than existing methods, proving the advantage of PSSP-MIRVT to deal with the label imbalanced difficulty. In addition, we used Sov, another important metric to provide the measurement at the segment level, to evaluate the overall performance of methods. As seen in Table 2, our PSSP-MIRVT can achieve a greatly outstanding performance on Sov with >15% surpassing previous methods. We speculate that convolution layers after features fusion enable our model to better capture the information at the local regions of peptides. Thus, it exhibits better performance than existing methods at the segment level of peptides. It is noteworthy that, our method is an end-to-end deep learning approach that can learn and extract features from sequence only and make predictions, without any professional feature engineering like traditional machine learning-based methods. In conclusion, it can be concluded that our model (E-state weighted 1.25) is more effective than Jpred and PHD in the prediction of peptide secondary structure prediction, especially for Acc_H and Sov.

Length preference investigation for peptide secondary structure prediction

To further investigate if our model has the length preference for peptide secondary structure prediction, we divided the test set into four subsets with different length intervals: [30, 35), [35,

Table 3. Results of the models with different input features

Methods	Observedj	Predicted			ACCj	ACC	SOV
		H	E	C	(%)	(%)	(%)
Word2Vec	H	4571	109	614	86.34	73.72	69.80
	E	189	368	562	32.88		
	C	925	264	2544	68.15		
PSSM	H	4532	81	681	85.61	76.41	74.09
	E	154	463	502	41.38		
	C	755	220	2758	73.88		
HMM	H	5280	1	13	99.73	53.44	26.25
	E	1119	0	0	0		
	C	3591	0	142	3.80		
Multi-view feature fusion	**H**	**4773**	**99**	**422**	**90.16**	**78.50**	**75.81**
	E	**139**	**636**	**344**	**56.84**		
	C	**836**	**341**	**2556**	**68.47**		

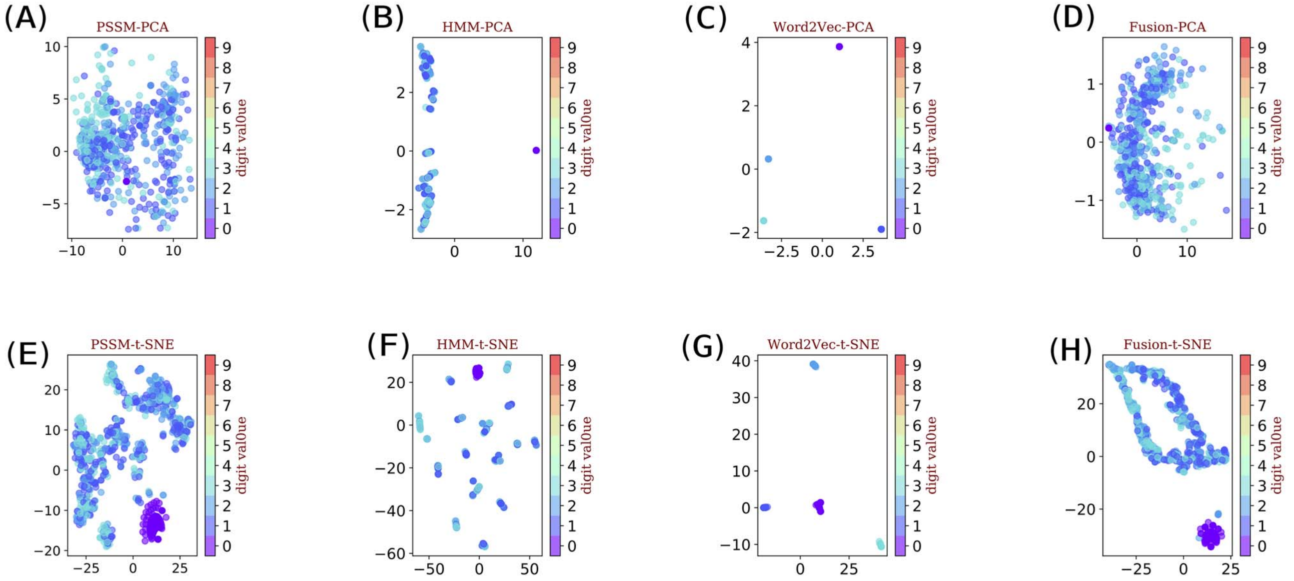


Figure 9. Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization of different input features: (**A–D**) represent PCA visualization results of PSSM, HMM, Word2vec and the feature fusion, respectively; (**E–H**) represent t-SNE visualization results of PSSM, HMM, Word2vec and the feature fusion, respectively.

40), [40, 45] and (45, 50] residues long, respectively. The details of the four subsets can be found in Supplementary Table S2. PSSP-MIRVT is evaluated on the four test subsets, and the results are presented in Supplementary Table S1. Figure 8 describes that our method better performs at the [30, 35), [35, 40), [40, 45] than that the (45, 50] interval, giving the highest Acc of 75.10, 78.93 and 81.58%, Sov of 64.58, 78.30 and 77.53%, respectively. By comparison, our performances lead by 1.41 and 8.31% in terms of Acc and Sov at the [30, 35], lead by 3.3 and 22.52% in terms of Acc and Sov at the (35, 40], and lead by 1.26 and 20.80% in terms of Acc and Sov at the (40, 45]. Interestingly, we found that the performances show a clear downtrend as the peptide length increases (Figure 8), which demonstrates that our method can achieve the best performance to predict shorter peptides. The observation implies that our model is superior in the prediction of the peptides even shorter than 30 residues long, while existing methods are not good at. Moreover, we also evaluated other existing methods on the three subsets with different length intervals, and the results are presented in Supplementary Table S2. Unfortunately, there is no clear trend observed as did in our model.

Determination of the optimal network architecture of our model

To determine the optimal network architecture of our model and achieve the best performance, we optimized the two major hyperparameters of our model, one of which is the number of convolutional layers, and the other is the segment numbers of the proposed BGRU. For the determination of the optimal convolutional layer number, we analyzed different layer numbers from 1 to 4. The results are reported in Supplementary Figure S5A, which shows that our model achieved the peak, giving the highest performances with the Acc of 78.50%, Acc$_H$ of 90.16%, Acc$_E$ of 56.84% and Acc$_C$ of 68.47%, respectively, when the layer number reaches 3. Specifically, the model with the three convolutional layers improves the Acc and Sov by 0.93 and 1.07% compared with one convolutional layer, indicating

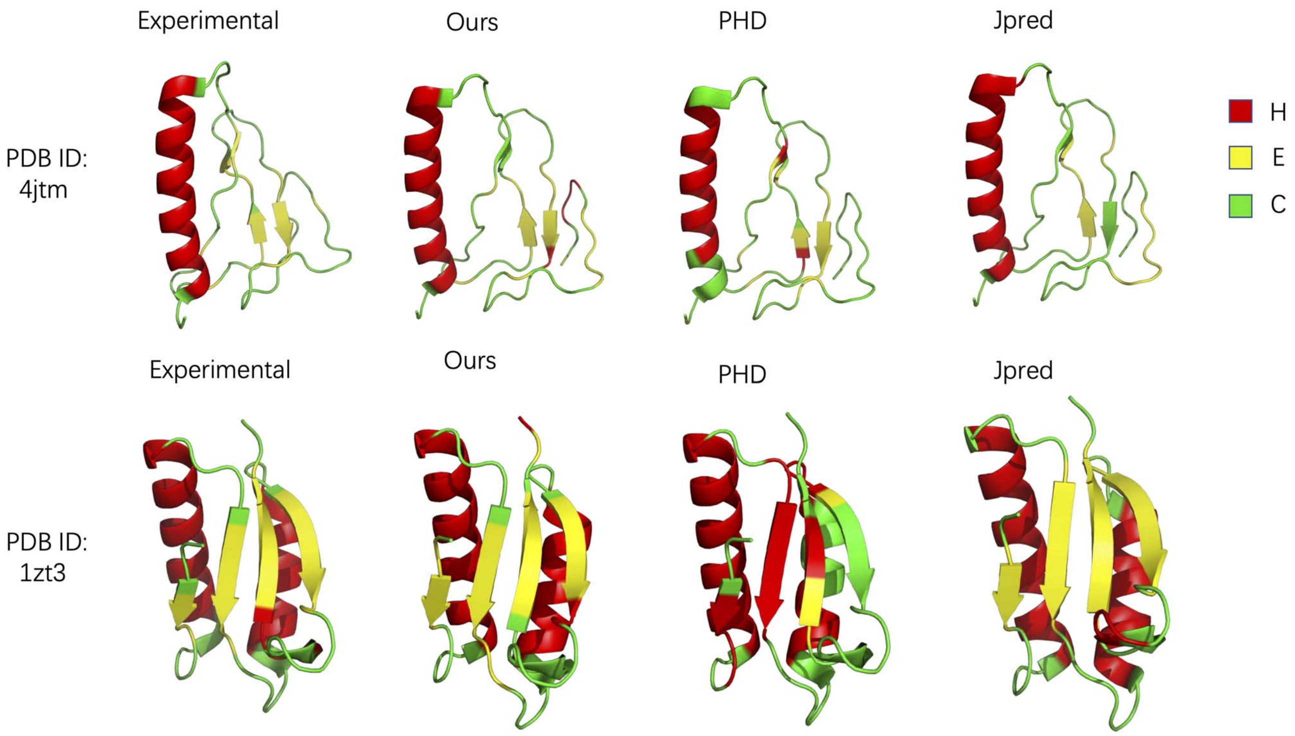


Figure 10. Visualization of secondary structures mapped into tertiary structures for our method and existing methods including PHD and Jpred.

that we can capture the most sufficient information with three convolutional layers. The possible reason for lower performance with one or two convolution layers is ought to be the lack of local feature information extraction, whereas the reason for four 2D-convolution layers might be too many training parameters, resulting in overfitting.

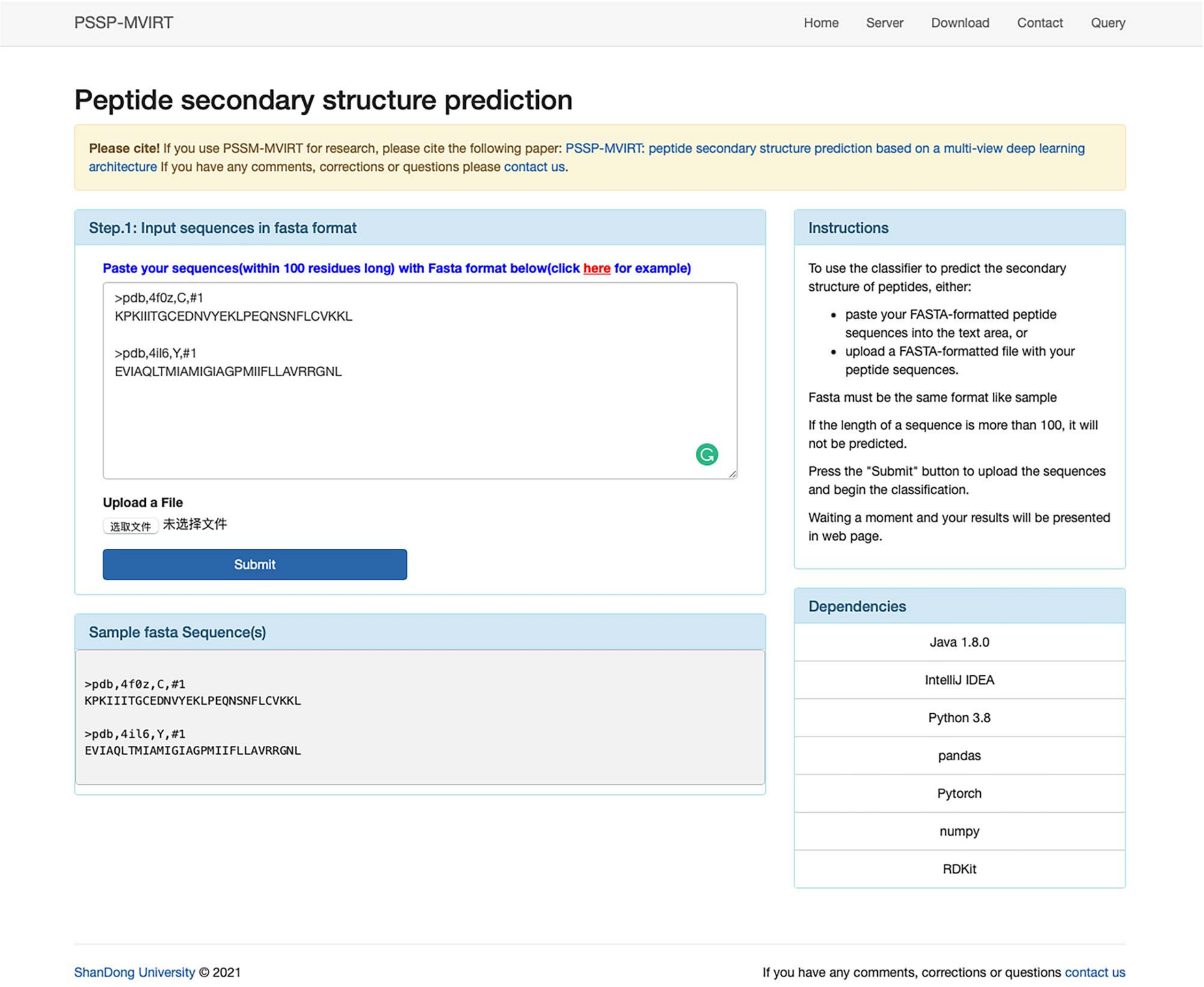
Similarly, to optimize the proposed Parallel BGRU architecture, we investigated different segment numbers with a range from 1 to 4 and illustrated the results in Supplementary Figure S5B. It is worth noting that if the segment number is set to 1, it is the original peptide without any segmentations. As expected, we achieved the best performance when the segment number is equal to 1 since segmentation results in loss of global information. To be specific, the best Acc and Sov are 78.50 and 75.81%, respectively, which are 0.94 and 2.59% higher than the models with second best architecture. Besides, our models with 1, 2 and 3 segments perform well, which proves that the features extracted at the peptide segment level might be a new method to keep exploring. However, if the peptide is segmented into too many subsequences, like four or more segments, as shown in Supplementary Figure S5B, the performance decreased significantly, whose potential reason might be that the local structure pattern is broken. Moreover, we also investigated the impact of the learning rate of our model. The detailed results can be found in Supplementary Material.

Impact of our multi-view feature fusion strategy

To analyze the impact of our multi-view feature fusion strategy, we compared our fused features with three individual features, including the features extracted from PSSM profile, HMM profile and Word2vec, respectively. To simplify the discussion, the three features are denoted as PSSM, HMM and Word2vec, respectively. The results of different features are presented in Table 3. Among the three individual features, the PSSM outperforms the other two, demonstrating that the evolutionary information is more effective for the prediction of peptide secondary structures. After fusing the features from PSSM profiles, HMM profiles and the embedding of peptide sequences with our multi-view learning strategy, the model performs best as compared with the individual features, which indicates the different information is complementary to each other, effectively improving the predictive performance. To understand the features intuitively, we also further visualized the feature space distribution of different feature representations, as shown in Figure 9. As seen, our proposed multi-view feature fusion strategy can create a better feature space, in which different structural states are more clearly separated, which further demonstrates that the multi-view feature fusion strategy is effective to improve the feature representation ability.

Case study

To intuitively compare the performance between our method and existing methods, we randomly selected two peptide chains with Protein Data Bank Identity (PDB ID)—4jtm and 1zt3, performing different methods for the secondary structure prediction on the two peptides. We illustrate the prediction results in Figure 10, in which we present the known experimental structures and the predicted structures of our method, PHD and Jpred, respectively. The secondary structures are mapped into the tertiary structures in which the red area represents helix (H), the yellow area represents strand (E) and the green area represents coil (C). It depicts that the predicted structures by our method are more similar to the experimental ones as compared with other methods. In particular, our method performs better on the local consecutive sequence regions than other methods, further confirming that our model can capture more discriminative local



Peptide secondary structure prediction

Please cite! If you use PSSM-MVIRT for research, please cite the following paper: PSSP-MVIRT: peptide secondary structure prediction based on a multi-view deep learning architecture If you have any comments, corrections or questions please contact us.

Step.1: Input sequences in fasta format

Paste your sequences(within 100 residues long) with Fasta format below(click here for example)

```
>pdb,4f0z,C,#1
KPKIIITGCEDNVYEKLPEQNSNFLCVKKL

>pdb,4il6,Y,#1
EVIAQLTMIAMIGIAGPMIIFLLAVRRGNL
```

Upload a File

Instructions

To use the classifier to predict the secondary structure of peptides, either:

- paste your FASTA-formatted peptide sequences into the text area, or
- upload a FASTA-formatted file with your peptide sequences.

Fasta must be the same format like sample

If the length of a sequence is more than 100, it will not be predicted.

Press the "Submit" button to upload the sequences and begin the classification.

Waiting a moment and your results will be presented in web page.

Sample fasta Sequence(s)

```
>pdb,4f0z,C,#1
KPKIIITGCEDNVYEKLPEQNSNFLCVKKL

>pdb,4il6,Y,#1
EVIAQLTMIAMIGIAGPMIIFLLAVRRGNL
```

Dependencies
Java 1.8.0
IntelliJ IDEA
Python 3.8
pandas
Pytorch
numpy
RDKit

ShanDong University © 2021

If you have any comments, corrections or questions contact us

Figure 11. PSSP-MVIRT server.

information. To be this end, we can conclude that our methods are better than existing methods.

Webserver

A user-friendly web server has been developed for readers to better predict peptide secondary structure using our best model (E-state weighted 1.25). The server was developed using HTML, JavaScript and Java as the front and installed on a Ubuntu Enterprise Linux server environment. The server takes FASTA sequences as input and presents secondary structure in text format. In addition, our server can take multi-sequences once, as shown in Figure 11. Also, our code and dataset can be downloaded at https://github.com/massyzs/PSSP-MVIRT for free. Up to now, the PSSP-MVIRT server can be reached at http://server.malab.cn/PSSP-MVIRT.

Conclusion

In this study, we have developed an end-to-end deep learning-based method named PSSP-MVIRT for peptide secondary structure prediction. Benchmarking comparisons demonstrate that our predictive model significantly outperforms existing methods especially on Acc_H and Sov. Moreover, we have also investigated the length preference of our model in the prediction of peptide secondary structures and demonstrated that our model shows better performance when predicting shorter peptides. Besides, we found that our proposed multi-view feature fusion learning strategy can enhance the feature representation ability, thus improving the predictive performance. The PSSP-MVIRT server can provide a potential way to improve the performance of the method for the research community.

Key Points

- In this study, we propose a multi-view deep learning-based method called PSSP-MVIRT for the prediction of peptide secondary structure prediction.
- Unlike existing methods trained based on hand-crafted features, we introduce a multi-view fusion strategy to integrate different information from multiple perspectives and generate a unified feature space that greatly improves the feature representation ability.

- Comparative studies show that the proposed PSSP-MVIRT significantly outperforms existing predictors, especially at the peptide segment level, which demonstrates that our method has a strong capability to capture the local discriminative information.
- To facilitate the use of our method, we establish a web server for the implementation of the proposed PSSP-MVIRT, which can provide a high-throughput prediction of peptide secondary structures. It is publicly accessible at http://server.malab.cn/PSSP-MVIRT.

Supplementary Data

Supplementary data are available online at *Briefings in Bioinformatics*.

Funding

Natural Science Foundation of China (Nos. 62072329 and 62071278).

References

1. Craik DJ, Fairlie DP, Liras S, *et al*. The future of peptide-based drugs. *Chem Biol Drug Des* 2013;**81**(1):136–47.
2. Singh H, Singh S, Raghava GPS. Peptide secondary structure prediction using evolutionary information. *bioRxiv* 2019; 558791. doi: 10.1101/558791.
3. Cheng J, Liu Y, Ma Y, *et al*. Protein secondary structure prediction based on integration of CNN and LSTM model. *Journal of Visual Communication and Image Representation* 2020;**71**:102844.
4. Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol* 1999; **292**(2):195–202.
5. Zhou J, Troyanskaya OG. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction 2014;**32**(1):745–53.
6. Wang S, Peng J, Ma J, *et al*. Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports* 2016;**6**(1):18962.
7. Li Z, Yu Y. Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.
8. Heffernan R, Yang Y, Paliwal K, *et al*. Capturing non-local interactions by long short term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers, and solvent accessibility. 2017;**33**(18):2842–49.
9. Busia A, Jaitly N. Next-step conditioned deep convolutional neural networks improve protein secondary structure prediction. 2017. arXiv preprint arXiv:1702.03865.
10. Fang C, Shang Y, Xu D. MUFOLD-SS: new deep inception-inside-inception networks for protein secondary structure prediction. *Proteins* 2018;**86**(5):592–8.
11. Mcguffin LJ, Bryson K, Jones DT. The PSIPRED protein structure prediction server. *Bioinformatics* 2000;**16**(4): 404–5.
12. Cole C, Barber JD, Barton GJ. The Jpred 3 secondary structure prediction server. *Nucleic Acids Res* 2008;**36**(Web Server issue):W197–201.
13. Wang S, Li W, Liu S, *et al*. RaptorX-Property: a web server for protein structure property prediction. *Nucleic Acids Res* 2016;**44**(W1):W430–5.
14. Rost B, Sander C, Schneider R. PHD–an automatic mail server for protein secondary structure prediction. *Comput Appl Biosci* 1994;**10**(1):53–60.
15. Montgomerie S, Scott M, Cruz JA, Shrivastava S, *et al*. PROTEUS2: a web server for comprehensive protein structure prediction and structure-based annotation. *Nucleic acids research* 2008;**36**(suppl_2):W202–W9.
16. Kapoor P, Singh H, Gautam A, *et al*. TumorHoPe: a database of tumor homing peptides. *PloS One* 2012;**7**(4):e35187.
17. Magnan CN, Baldi P. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics* 2014;**30**(18):2592–7.
18. Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. 2017. arXiv preprint arXiv:1706.03762.
19. Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. 2013. arXiv preprint arXiv:1301.3781.
20. Altschul SF, Madden TL, Schäffer AA, *et al*. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 1997;**25**(17):3389–402.
21. SWISS-PROT. Berlin Heidelberg: Springer, 2008.
22. Asai K, Hayamizu S, Handa K. Prediction of protein secondary structure by the hidden Markov model. *Bioinformatics* 1993;**9**(2):141–6.
23. Potter SC, Luciani A, Eddy SR, *et al*. HMMER web server: 2018 update. *Nucleic Acids Res* 2018;**46**(W1):W200–w4.
24. Cho K, Van Merriënboer B, Gulcehre C, *et al*. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014. arXiv preprint arXiv:1406. 1078.
25. Wang Y, Mao H, Yi Z. *Protein secondary structure prediction by using deep learning method*. *Knowledge-Based Systems* 2017;**118**:115–23.
26. Liu T, Wang Z. SOV_refine: a further refined definition of segment overlap score and its significance for protein structure similarity. *Source Code Biol Med* 2018;**13**:1.
27. Yong H, Huang J, Meng D, *et al*. Momentum batch normalization for deep learning with small batch size. *European Conference on Computer Vision* 2020;**12357**:224–40.
28. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 2012;**25**: 1097–105.