Mathematical Biosciences and Engineering

MBE, 20(5): 7721–7737.
DOI: 10.3934/mbe.2023333
Received: 06 December 2022
Revised: 18 January 2023
Accepted: 11 February 2023
Published: 20 February 2023

http://www.aimspress.com/journal/MBE

Research article

WG-ICRN: Protein 8-state secondary structure prediction based on Wasserstein generative adversarial networks and residual networks with Inception modules

Shun Li, Lu Yuan, Yuming Ma* and Yihui Liu*

School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China

* **Correspondence:** Email: mym@qlu.edu.cn, yxl@qlu.edu.cn.

Abstract: Protein secondary structure is the basis of studying the tertiary structure of proteins, drug design and development, and the 8-state protein secondary structure can provide more adequate protein information than the 3-state structure. Therefore, this paper proposes a novel method WG-ICRN for predicting protein 8-state secondary structures. First, we use the Wasserstein generative adversarial network (WGAN) to extract protein features in the position-specific scoring matrix (PSSM). The extracted features are combined with PSSM into a new feature set of WG-data, which contains richer feature information. Then, we use the residual network (ICRN) with Inception to further extract the features in WG-data and complete the prediction. Compared with the residual network, ICRN can reduce parameter calculations and increase the width of feature extraction to obtain more feature information. We evaluated the prediction performance of the model using six datasets. The experimental results show that the WGAN has excellent feature extraction capabilities, and ICRN can further improve network performance and improve prediction accuracy. Compared with four popular models, WG-ICRN achieves better prediction performance.

Keywords: residual network; Wasserstein generative adversarial network; inception; prediction of protein 8-state secondary structures

1. Introduction

Proteins play an extremely important role in our daily activities, with functions such as immunity and cell signaling. Their different functions are due to their different structures. Therefore, to fully understand the functions of proteins and related research, it is necessary to predict their structure. Although the advent of AlphaFold2 [1] has changed the protein prediction landscape, it has achieved very reliable results for the prediction of protein tertiary structure [2], as the prediction of secondary structure is still of great significance, because the secondary structure will improve the alignment of the tertiary structure, thereby affecting the spatial morphology of the protein, so this paper proposes a method based on deep learning to predict the secondary structure of proteins.

Protein secondary structure is the local spatial conformation of amino acid residues in protein polypeptide chains, mainly in the form of 3-states (helix (H), chain (E), coil (C)), which can be divided into 8-states, namely α-helix (H), helix (G), π-helix (I), β-bridge (B), β-sheet (E), bend (S), turn (T) and coil (C) [3–5]. This study was devoted to the 8-state prediction of proteins, which can be more informative and more challenging.

In the 1990s, Burkhard Rost and Chris Sander first used neural networks to predict the secondary structure of proteins [6]. In addition to achieving excellent results, this method was pioneered in the field of protein structure prediction. Early protein secondary structure prediction used statistical methods and heuristic rules [7], such as Support Vector Machine [8], Bayesian classification algorithm, Markov model [9], and Feedforward neural network [10,11] that have been applied in the prediction of protein secondary structure. With the advent of the post-genomic era, the amount of protein data has increased. Owing to the high cost and difficulty of experiments, traditional experimental determination methods have been unable to meet the growing demand for protein and structural data analyses. Therefore, methods for protein structure prediction have become a hot issue in bioinformatics. In the last few years, as deep learning has made tremendous progress in natural language processing, machine vision and speech recognition, bioinformatics has also begun to extensively use deep learning methods.

In recent years, many scholars and researchers have achieved excellent results in the field of 8-state research on protein secondary structure. Busia et al. proposed a protein sequence prediction technique, which combined the successful experience of using convolutional neural networks in the past and language modeling, and achieved good results [12]. Using the combined synergy of a convolutional neural network, residual network and bidirectional recurrent neural network prediction, Zhang et al. [13] designed a local block composed of convolutional filters and raw input to capture local Sequence Features. Krieger et al. determined estimated class membership probabilities of residues in proteins using the nearest neighbor search, which is then fed into another dynamic programming algorithm, showing good results on the CASP dataset [14]. Uddin et al. proposed to combine the self-attention mechanism with the Deep Inception-Inside-Inception (Deep3I) network to track residues between amino acids at different distances through interaction [15]. Kotowski et al. proposed a single-sequence-based method called ProteinUnet, which effectively shortens the inference time, and improves the training speed [16]. Sonsare and Gunavathi proposed a model consisting of a 1D-Convnet and an improved recurrent neural network with an improved sequential coin toss optimizer, achieving good prediction accuracy on CB513 and CullPDB [17].

This paper proposes an 8-state protein secondary structure prediction method named WG-ICRN, as it based on WGAN and ResNet with Inception. WG-ICRN extracts the feature information of the

protein use WGAN, and then combines this information with PSSM [18] to enhance the features, and the combined feature matrix is named WG-data. The increased length and width of WG-data makes its feature maps larger in area and richer in feature information, since WG-data was input into the ICRN module as input data. ICRN was a transformation of the residual network. Inception was introduced into the residual network to replace the convolution layer, and the width of the input data feature map was increased through multi-scale convolution to further enrich the features.

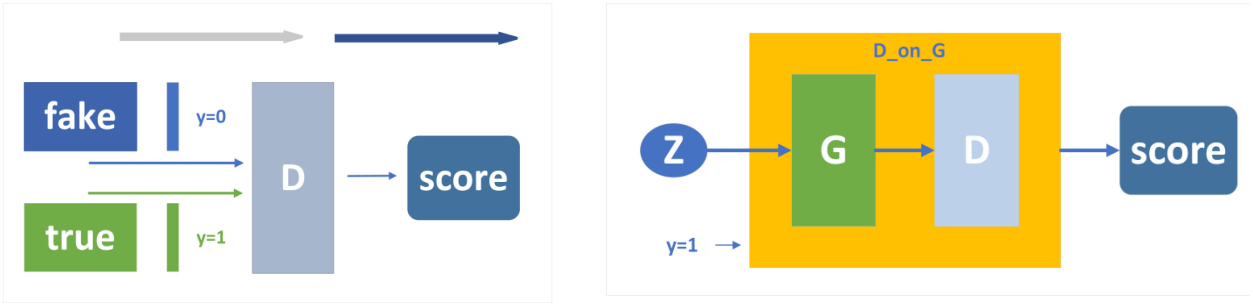
The main contributions of this study are: (1) We use WGAN to extract protein information in sliding window processed PSSM, and combine PSSM to build a new feature set with rich protein features. (2) The ICRN model combines Inception and the residual network, increasing the width of the network through Inception, while the residual network guarantees the depth of the network, improving the performance of the network from two aspects. (3) ICRN reduces the number of training parameters by using multiple smaller filters to reduce the dimension of the data, so the training time is shorter than the residual network, saving system resources. (4) Experimental results show that WG-ICRN is superior to other popular models in prediction accuracy.

2. Model structure and related theories

2.1. Wasserstein generative adversarial network

Generative adversarial network (GAN) [19] was proposed by Ian Goodfellow in 2014, and consists of two parts: generator (G) and discriminator (D). G can generate similar fake data by learning the distribution characteristics of real data, while D judges and scores the authenticity of the data. GAN has been applied to image denoising and feature extraction [20–22], and has been proved to have good properties. GAN also has the problem that the model is difficult to optimize, as the tedious problem of G and D parameter optimization is difficult to solve. In recent years, a lot of optimization algorithms [23–26], such as Aquila optimizer [27] and the Gazelle optimization algorithm [28], provide a direction to solve this difficult problem. But the more critical issue for GAN is this: Owing to the approximate optimal D of GAN, the G loss faces the problem of gradient disappearance. The WGAN uses the Wasserstein distance, which can alleviate this critical problem, and has the advantage of reflecting the distance between two distributions even if they do not have any overlap [29].

The specific training process of WGAN is the constant game and confrontation between G and D. When training D, the data generated by the previous round of G and real data are directly spliced together as x, the fake data corresponded to 0, and the real data corresponds to 1. Then, a score (a number from 0 to 1) can be generated through D, x input, and through the loss function composed of the score and y, gradient backpropagation can be performed. The training process of D is shown in Figure 1(a). When training G, G and D need to be treated as a whole, which is named "D_on_G". The output of this whole system (referred to as the DG system) is still the score. Entering a set of random vectors z, we can generate a set of random data in G, and score the generated set of data through D to obtain the score, which is the forward process of the DG system. The training process is presented in Figure 1(b).



(a). The training process of the discriminator (b). The training process of the generator

Figure 1. The training process WGAN.

The GAN objective function is as formula (1), where, x and z represent the input real and random data, G(z) represents the data generated after G processes the random data z, and D(x) represents the probability that the data is the real data. In the most ideal case, G can generate data G(z) that is very similar to the real protein data, and it is difficult for D to judge the authenticity of these data, that is, D(G(z)) = 0.5.

$$\min_G \max_D (\mathrm{D}, \mathrm{G}) = \mathrm{E}_{x\sim P_{\mathrm{data}}(x)}[\log \mathrm{D}(x)] + \mathrm{E}_{z\sim P_z(x)}\left[\log\left(1-\mathrm{D}(\mathrm{G}(z))\right)\right] \tag{1}$$

Objective function (1) to be optimized by the GAN can be divided into 2 parts: Part 1, fix the G and optimize the D, then (1) can be rewritten as formula (2), convert it to minimized form as formula (3). Part 2, fix the D, optimize the G, which is equivalent to minimizing, as formula (4), so that the argument of D does not exceed a fixed constant, just maximize the formula (5).

$$\max_D \mathrm{E}_{x\sim P_r}[\log \mathrm{D}(x)] + \mathrm{E}_{x\sim P_g}\left[\log\left(1-\mathrm{D}(x)\right)\right] \tag{2}$$

$$\min_D -\mathrm{E}_{x\sim P_r}[\log \mathrm{D}(x)] - \mathrm{E}_{x\sim P_g}\left[\log\left(1-\mathrm{D}(x)\right)\right] \tag{3}$$

$$\min_G \mathrm{E}_{x\sim P_g}\left[\log\left(1-\mathrm{D}(x)\right)\right] \tag{4}$$

$$L = \mathrm{E}_{x\sim P_r}[\mathrm{D}(x)] - \mathrm{E}_{x\sim P_g}\left[\log\left(\mathrm{D}(x)\right)\right] \tag{5}$$

In this experiment, we introduced CNN in WGAN to assist in feature extraction. Local receptive fields and weight sharing operations in CNN can realize displacement, scaling and distortion invariance. We use ReLU as the activation function of the CNN, which is calculated as Eq. (6).

$$F_k^i = f\left(\sum_h P_h^{i-1} * W_k^i + b\right) \tag{6}$$

Here, f is ReLU, which P_h^{i-1} represents the feature map obtained from the input protein data and the convolution kernel of the previous layer, W_k^i is the convolution kernel of the No. i layer, k is the number of convolution kernel, and b is the bias parameter. At the same time, we use gradient punishment to improve the stability of the network during WGAN training. The network structure of WGAN used in the experiment is shown in Figure 2.

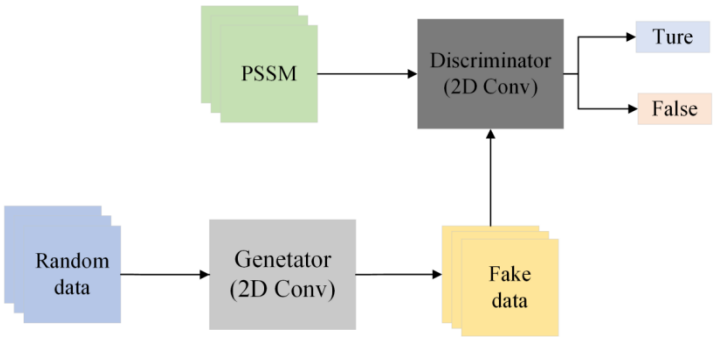


Figure 2. WGAN model diagram.

2.2. Residual networks (ResNet)

The network depth is very important for the performance of the model, but, in fact, the deep network will face degradation problems, and the accuracy will also decrease. Studies have shown that this deep network has the problem of gradient explosion or disappearance, and Residual Networks (ResNet) [30] introduces the residual learning to alleviate this problem. Nowadays, ResNet is used in computer vision and medical analysis [31,32].

The specific process is that for a block structure, where the learned characteristics from when the input is X are recorded as $H(X)$, and we hope that the residual $F(X) = H(X) - X$ can be learned, when the original characteristics are $F(X) + X$. Because residual learning is easier than the original feature direct learning, when the residual is 0, the block only makes the constant mapping, which makes the network performance not decline, but in fact the residual will not be 0, which will also make the block learn the new feature on the basis of the input feature, so that it has better performance. Residual learning is similar to short-circuit connections, and is structured as shown in Figure 3.

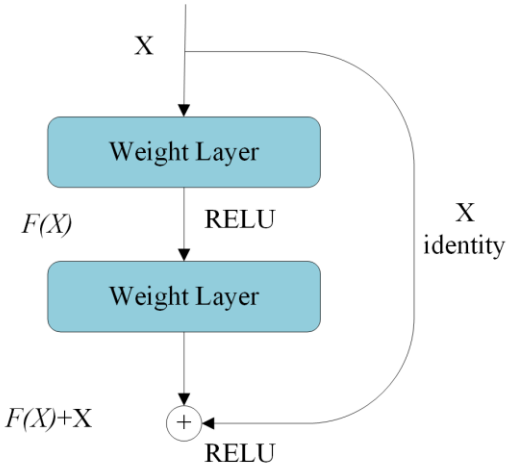
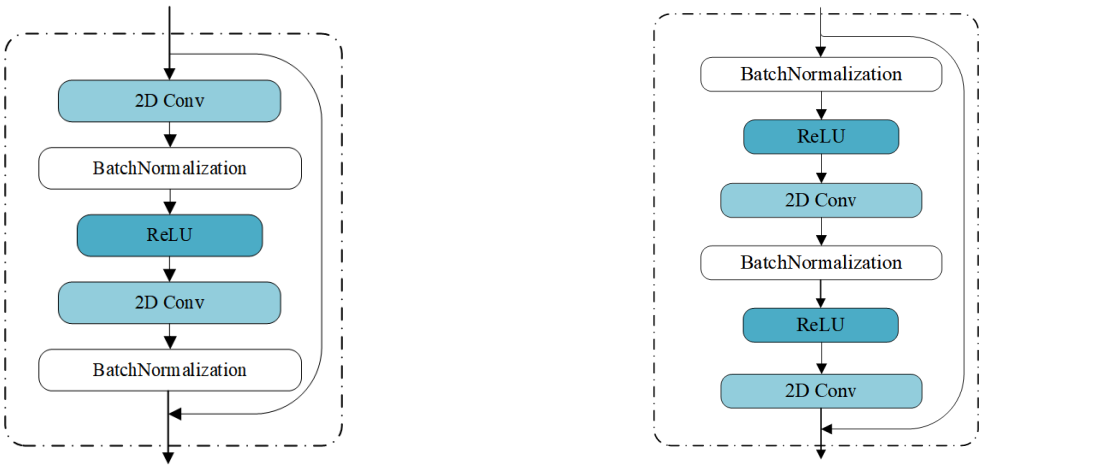


Figure 3. Structure of residual learning.

The origin of the residual block structure consists of convolution and pooling before residual learning. The origin residual connection method is shown in Figure 4(a). The article [33] has conducted a more detailed analysis experiment on the residual structure and obtained the optimal residual learning structure, that is, batch normalization and ReLU were performed before convolutional layers, and the structure is shown in Figure 4(b).



(a). The origin residual block (b). The residual block used in this experiment

Figure 4. The residual block structure.

2.3. Inception

In 2014, Szegedy et al. proposed the Inception structure for the first time [34]. Inception performs convolution operations on the feature map at a certain moment by using convolution kernels of different sizes, so as to obtain a new feature map, and then samples the size of the input feature according to the feature map of different sizes. It is worth noting that Inception does not change the size of the original features, but only enriches the characteristic information of the protein through different convolution kernels, making the characteristics diversified. The network structure of InceptionV2 is shown in Figure 5.

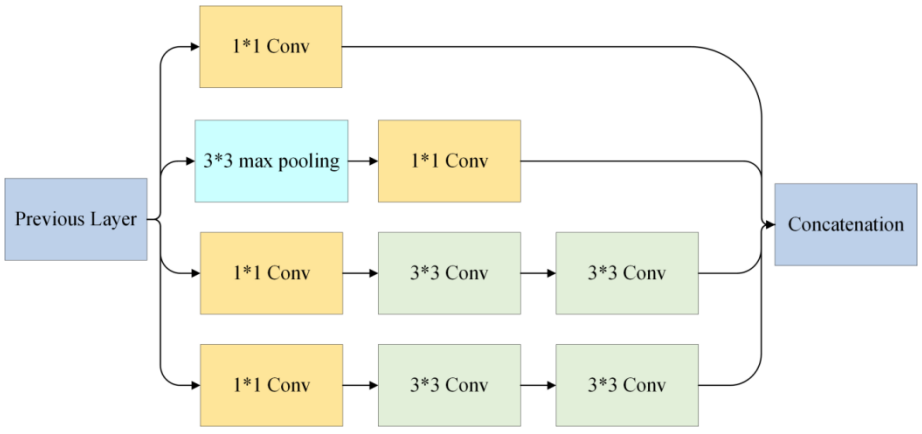


Figure 5. The network structure of InceptionV2.

2.4. The proposed ICRN model

In this experiment, we use the improved Inception module instead of the first convolutional layer and maxpooling layer in the ResNet model, and the improved Inception module informs the WG-data to extract learning at different scales through convolutional kernels of different sizes, which enriches the feature information and improves the prediction accuracy of protein secondary structure prediction. Improved Inception module structure is shown in Figure 6.

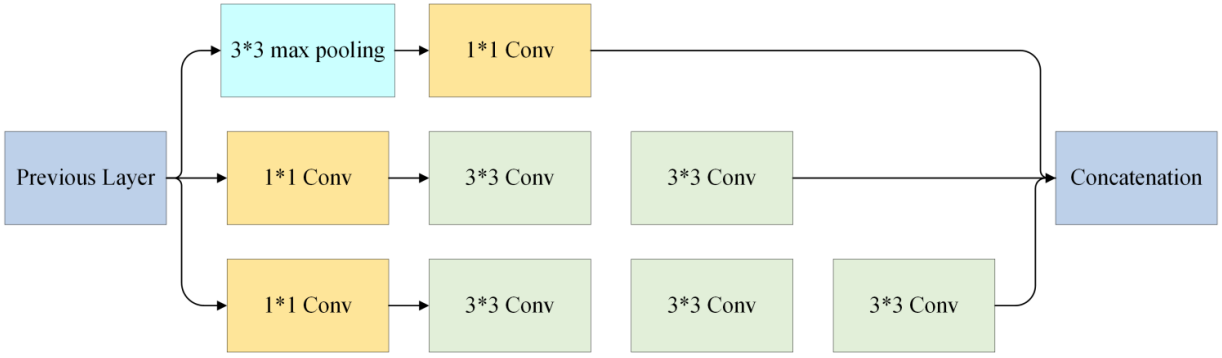


Figure 6. The improved Inception module structure.

In this paper, we use ICRN-N to represent the improved ResNet of different depths, and N refers to the number of network layers with privileged values, that is, only convolutional layers, as fully connected layers and pooled layers are included. We set the number of layers with weights of 10, 18 and 34 as the experimental model, and the structures of ICRN-10, ICRN-18, and ICRN-34 are shown in Table 1, respectively.

Table 1. ICRN structure at different depths.

Layer name	ICRN-10	ICRN-18	ICRN-34
Inception Block		$\begin{bmatrix}1\times1\\3\times3\\3\times3\\3\times3\end{bmatrix}$ $\begin{bmatrix}1\times1\\3\times3\\3\times3\end{bmatrix}$ $\begin{bmatrix}\text{Max pool}\\1\times1\end{bmatrix}$	
Residuals-Block-1	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times3$
Residuals-Block-2	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,128\\3\times3,128\end{bmatrix}\times4$
Residuals-Block-3	/	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,256\\3\times3,256\end{bmatrix}\times6$
Residuals-Block-4	/	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3,512\\3\times3,512\end{bmatrix}\times3$
Average pool, fully connected, softmax			

2.5. WG-ICRN networks structure

The structure of WG-ICRN is shown in Figure 7, and it can be seen that our network model is mainly divided into the WGAN and ICRN modules. Firstly, a protein was processed into PSSMS with size of 20 × L, where 20 is the feature dimension, and L represents the protein length. Since the lengths of different proteins were different, sliding Windows (The length is W) were used to cut PSSMS. The processed data would be used as the learning model of WGAN, and key features would be extracted through the confrontation of G and D. We use several convolutional layers to assist G and D networks; G networks use Leaky ReLU as the activation function, due to the large number of iterations, and to prevent overfitting, we use Dropout in G networks. We Concatenated the final data (Si-data) generated by D and the PSSM processed by sliding window into a matrix of 40 × W, named WG-data.

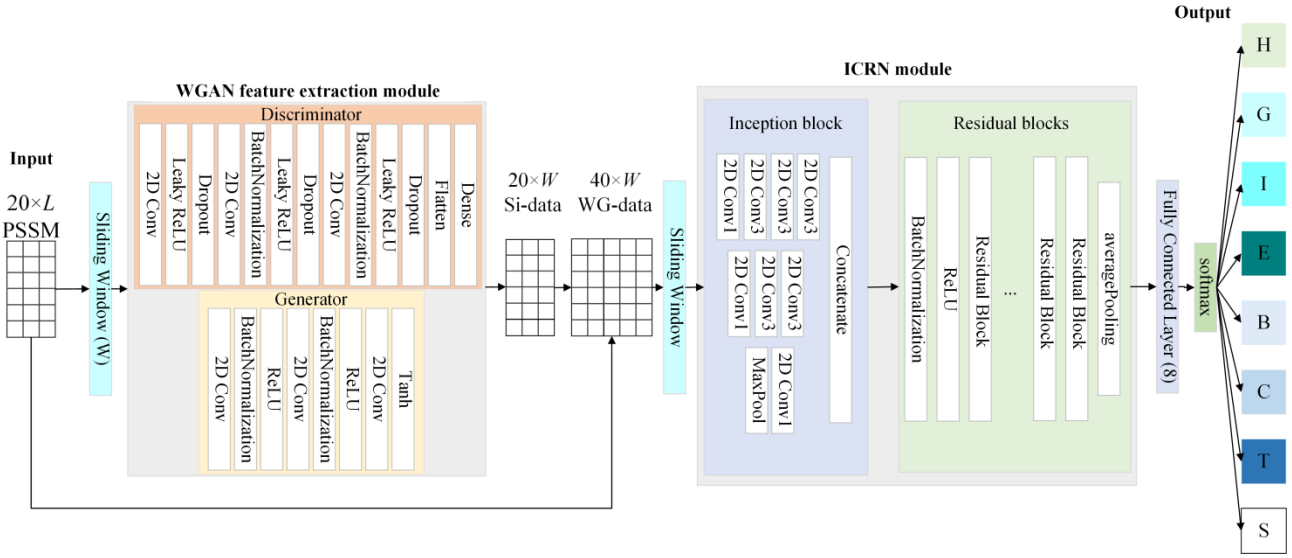


Figure 7. WG-ICRN networks structure.

The ICRN module consists of two parts, namely Inception block and residual block. The improved Inception will replace the first convolution layer with a size of 7 × 7, and the max pooling layer with a size of 3 × 3 in this model. The improved Inception can achieve the same convolution effect by using three layers of 3 × 3 convolution layers with fewer training parameters, which will save training time. At the same time, the multi-scale convolution model in Inception can extract feature maps of different sizes, which, when combined together, will increase the richness of features to some extent.

After two feature enhancements, the residual block in ICRN will conduct the final training on the data. We respectively use the residual network of different depths to test the data. At the end of the network, we adopt an average pooling layer to replace the flattening of the matrix features, which reduces the number of parameters.

Finally, in the output layer of the model, we use the fully connected layer and softmax layer to output the final prediction results and calculate the prediction accuracy through the evaluation criteria.

3. Experiment results

3.1. Datasets and features

The main public datasets used in this study are the CullPDB [35] dataset and the datasets [36–41] CASP10, CASP11, CASP12, CASP13, CASP14 and CB513. The CullPDB dataset contains 12,288 proteins. These datasets show that the similarity of the data was less than 25%. In this study, the repeated protein dataset CullPDB was removed as the training set, with a total of 11650 proteins. For the CASP10-14, and CB513 datasets, there were 99, 81, 19, 22, 24 and 513 protein chains, respectively. The number of protein sequences in datasets is listed in Table 2.

Table 2. Statistical data in datasets.

Datasets	Number of proteins
CullPDB	11650
CASP10	99
CASP11	81
CASP12	19
CASP13	22
CASP14	24
CB513	513

Position-Specific Scoring Matrix (PSSM) is rich in biological evolution information, which greatly improves the accuracy of protein secondary structure prediction. It is a widely used feature for information. The PSSM of this experiment was generated by multiple sequence alignment of proteins in the NR database, setting the PSI-BLAST [42] parameter threshold to 0.001 and 3 iterations.

3.2. Evaluation criteria

Q8 and SOV are evaluation criteria for evaluating protein prediction performance from different perspectives. Q8 is the ratio of the number of correctly predicted amino acids to all amino acids. It can be expressed by formula (7), and S is the total number of amino acids.

$$Q8 = \frac{S_H+S_E+S_G+S_B+S_I+S_S+S_T+S_C}{S} \times 100 \tag{7}$$

Among them, S_H, S_E, S_G, S_B, S_I, S_S, S_T and S_C are the numbers of correctly predicted α-helices, beta-sheets, β bridges, 310 helices, π helices, turns, β-turns and random coils, respectively, and S is the total number of amino acids. The secondary structure accuracy of a state is calculated as formula (8).

$$Q_j = \frac{S_j}{N_j}, j \in \{H, E, G, B, I, S, T, C\} \tag{8}$$

SOV [43] is a measure based on the ratio of overlapping segments. Assuming that all observed structural fragments are labeled S_{ob}, all predicted fragments are labeled S_{pr}, and S_o is a fragment with the same state as S_{ob} and S_{pr}, and for any pair of fragments in S_o, the actual length is *minov* (S_{ob}, S_{pr}),

where at least one residue has a total length of *maxov* (S_{ob}, S_{pr}). The SOV calculation formula as formula (9).

$$SOV = \frac{100}{N_{SOV}} \sum_{S_o} \left[\frac{minov(S_{ob}, S_{pr}) + \sigma(S_{ob}, S_{pr})}{maxov(S_{ob}, S_{pr})} length(S_{ob}) \right] \tag{9}$$

Among them, $\sigma(S_{ob}, S_{pr})$ allows the change of the observed fragment boundary in the protein structure, which is defined by the formula (10).

$$\sigma(S_{ob}, S_{pr}) = min \begin{Bmatrix} (maxov(S_{ob}, S_{pr}) - minov(S_{ob}, S_{pr})) \\ minov(S_{ob}, S_{pr}) \\ int[len(S_{ob})]/2 \\ int[len(S_{pr})]/2 \end{Bmatrix} \tag{10}$$

3.3. Experimental results and parameter influence

The experiment in this paper was done with the processor Intel(R) Xeon(R) Glod 5118, and the graphics card RTX2080Ti and the system Linux. Firstly, we tested the influence of the number of CNN convolutional layers on the WGAN feature extraction ability. The size of the convolution kernel was set to 3 × 3 × 64, and different convolutional layers were set to 1, 2, 3, 4 and 5, and tested on CASP11-14. As can be seen from Table 3, when the number of convolutional layers is 3, the data generated by G are closer to the real data.

Table 3. Effect of the number of convolutional layers on Q8.

Layers	CASP11	CASP12	CASP13	CASP14
1	68.26	68.85	67.21	68.36
2	71.27	70.63	67.76	69.41
3	72.55	71.81	69.88	70.29
4	71.71	71.23	68.73	68.22
5	70.33	69.46	67.24	67.61

Because the number of iterations of the generator and discriminator will also affect the feature extraction ability of the WGAN, this study tested the influence of different iterations on the experiment, in which 3 convolutional layers are set, and the parameters of the convolution kernel are set to 3 × 3 × 64, 3 × 3 × 128 and 3 × 3 × 256, and the experimental results under different iterations are shown in Figure 8.

As shown in Figure 7, the best effect is achieved when the number of iterations is 20,000, that is, the features extracted by G are the most realistic and effective. After more than 20,000 iterations, D's ability to judge the authenticity of the generated data decreases to the point where there is a large error between the simulated and real features.

To test the influence of the length of sliding window on the experimental results, we selected 13, 15, 17, 19 and 21 for Q8 prediction. The experimental results are shown in Table 4, which shows that when the sliding window is 19, the experimental results are the best.

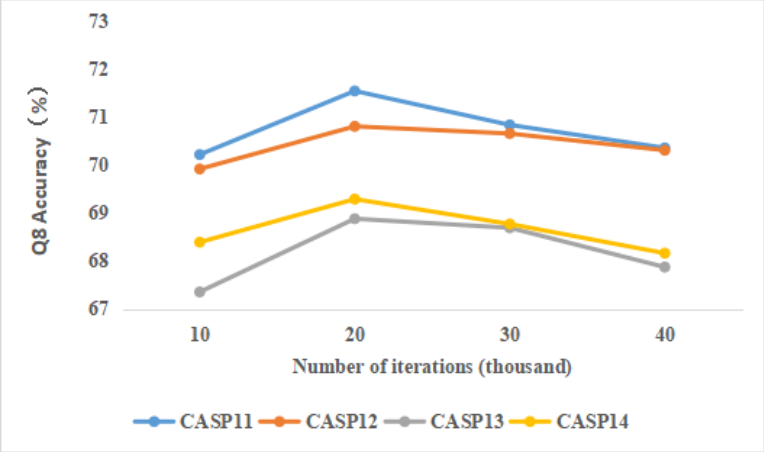


Figure 8. Q8 accuracy with different iterations.

Table 4. Q8 accuracy under different length of sliding windows.

Sliding window	CASP11	CASP12	CASP13	CASP14
13	67.47	68.16	65.88	65.20
15	68.09	69.27	66.67	67.41
17	70.66	70.24	68.18	69.13
19	**71.55**	**70.81**	**68.88**	**69.29**
21	70.29	70.41	68.42	68.94

Using different depths of ResNet, we tested CASP11-14, and obtained the experimental results shown in Figure 9. It can be seen that WG-ICRN-18 has the highest accuracy, because the dimension of WG-data is not high, and when the number of layers is too deep, part of the data will be lost, which causes a decrease in accuracy. In addition, we calculate the *SOV* and *Qj* ($j \in \{H, E, G, B, I, S, T, C\}$) of each test set under the WG-ICRN method, and the results are shown in Table 5.

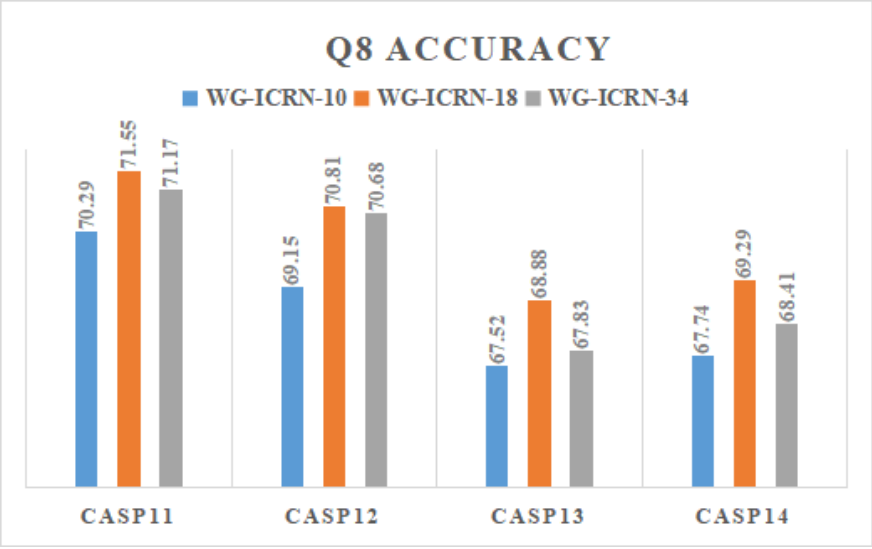


Figure 9. Q8 accuracy under WG-ICRN at different depths.

Table 5. Q8 and SOV accuracy in the datasets.

Dataset	CASP10	CASP11	CASP12	CASP13	CASP14	CB513
SOV	70.98	69.37	68.83	67.41	66.39	73.91
Q8	73.32	71.55	70.81	68.88	69.29	75.56
Q_G	52.72	55.32	47.90	36.56	33.51	37.71
Q_H	92.66	85,74	87.43	93.2	89.75	92.25
Q_I	0	0	0	0	0	0
Q_T	62.21	49.47	56.78	57.31	45.29	53.67
Q_B	9.81	19.30	7.25	8.30	3.88	7.68
Q_E	88.80	82.15	77.76	84.37	76.79	80.44
Q_S	53.88	43.68	48.90	34.74	13.33	25.39
Q_C	68.12	62.91	67.38	70.71	68.54	71.37

This paper divides CullPDB into five parts for five-fold cross-validation, four as training sets and one as test, and the results of cross-validation are shown in Table 6.

Table 6. Q8 under five-fold cross-validation.

	1	**2**	**3**	**4**	**5**	**Average**
Q8	72.72	73.18	73.43	72.61	71.36	72.66

We did ablation experiments to demonstrate the importance of each structure. We used five network models to test CASP11-14, and the experimental results are presented in Table 7. Thus, WG-ICRN is the model proposed in this paper, WG-Res is combining WGAN and ResNet and WG-CNN is a network model combining WGAN and CNN, where the three methods input data adopts WG-data, and the network model structure of CNN uses 3 convolutional layers: 3 × 3 × 64, 3 × 3 × 128 and 3 × 3 × 256. ResNet is a residual network model based on the best ResNet-18, CNN is using a 3-layer convolutional neural network, a structure is 3 × 3 × 64, 3 × 3 × 128, and 3 × 3 × 256. The input data for ResNet and CNN were PSSM. In addition, we calculated the average training time of each of the 11650 proteins in the CullPDB dataset for the 5 methods. These results are shown in Table 7.

Table 7. Comparison of results from ablation experiments.

	CASP11	CASP12	CASP13	CASP14	Training time (s)
WG-ICRN	71.55	70.81	68.88	69.29	21.9
WG-Res	71.43	70.67	68.83	69.17	22.4
WG-CNN	70.47	68.79	67.33	68.24	21.7
ResNet	68.76	67.84	65.57	66.19	9.8
CNN	66.62	65.29	63.69	64.71	9.6

By comparing the experimental results of the five methods in the table, it can be seen that, when the input data is the same PSSM, the prediction accuracy of ResNet is higher than that of CNN, because the deeper number of network layers makes training more adequate and increases training time, but the efficiency of ResNet is still better than that of CNN. WGAN extracted features significantly improves the prediction accuracy of Q8, and greatly increases the training time because of the increased size of the training data. Our proposed ICRN model reduces the time complexity by introducing Inception and extracts horizontal multi-scale feature fusion, which reduces the training time and improves the prediction accuracy compared with ResNet.

3.4. Comparison with other methods

Furthermore, we compared other models with our proposed method. Common with WG-ICRN is that these methods are improvements or hybrid models of CNN, and among them is, DeepACLSTM [44], which combines asymmetric convolution (ACNN) and bidirectional long short-term memory neural network (BiLSTM), 1D-Inception [45] Taking inspiration from InceptionV3 to extract features from 1D sequences using several parallel convolutions, DCRNN [46] uses an end-to-end model with multi-scale CNN and stacked bidirectional GRU. CNN_BIGRU [47] used CNN and bidirectional gated recurrent units to prediction. We re-run the code of the above method on the same computer, and the training set uses the same as WG-Res, which has been screened by data, and contains a total of 11,650 proteins. The experimental results are shown in Table 8. By comparison, it can be seen that WG-ICRN has excellent performance in predicting the secondary structure of protein 8 states, because of the deep layers advantages of ResNet, and, in addition, the matrix will contain richer feature information than the one-dimensional sequence, so the experimental results as input data will be better.

Table 8. Q8 accuracy comparison of five methods.

Method	CASP10	CASP11	CASP12	CASP13	CASP14	CB513
DeepACLSTM	73.09	71.49	70.35	**68.91**	68.81	75.51
1D-Inception	71.86	70.07	69.78	67.51	68.3	74.68
DCRNN	72.11	70.50	69.41	68.05	68.87	74.85
CNN_BIGRU	71.87	70.94	69.67	67.83	68.69	75.54
WG-ICRN	**73.32**	**71.55**	**70.81**	68.88	**69.29**	**75.56**

4. Conclusions and future works

The prediction of protein secondary structure is important work to comprehensively understand and explore the diverse functions and spatial structure of proteins. This paper combines WGAN and ICRN, for the first time, to propose a novel protein 8-state secondary structure prediction method, WG-ICRN. In WG-ICRN, WGAN can extract protein features in amino acid sequences, and then we combine PSSM with the extracted features into a new feature matrix WG-data that contains more protein feature information. We also use ICRN to further extract the residue interactions in WG-data and complete the prediction. We introduced the improved Inception module into ResNet and proposed the ICRN model, which cannot only reduce parameter calculation and improve efficiency, but also increase network width to improve network performance. We evaluate the proposed model on six datasets CASP10-14 and CB513. Experimental results show that the prediction performance of WG-

ICRN is better than the four other popular methods. In addition, this paper also proves that WGAN has a powerful feature extraction ability, and the ICRN model can handle protein data more comprehensively, and the combination of the two has achieved remarkable results. However, it is difficult for WGAN to achieve the balance between generator and discriminator, which also makes training more tedious and time-consuming. In addition, secondary structure prediction is also slightly affected by residues in the global range, but WG-ICRN mainly focuses on local features and ignores long-range features. In future work, we will continue to optimize the feature extraction technique and fully utilize different feature information of protein sequences to improve prediction performance.

Availability

The codes and datasets for this work are at https://github.com/ShunLi999/WG-ICRN.git

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant number 61375013) and the Natural Science Foundation of Shandong Province (grant number ZR2013FM020).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, et al., Improved protein structure prediction using potentials from deep learning, *Nature*, **577** (2020), 706–710. https://doi.org/10.1038/s41586-019-1923-7
2. J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, et al., Highly accurate protein structure prediction with AlphaFold, *Nature*, **596** (2021), 583–589. https://doi.org/10.1038/s41586-021-03819-2
3. J. Zhou, O. Troyanskaya, Deep supervised and convolutional generative stochastic network for protein secondary structure prediction, in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, **32** (2014), 745–753.
4. A. Yaseen, Y. H. Li, Template-based c8-scorpion: A protein 8-state secondary structure prediction method using structural information and context-based features, *BMC Bioinformatics*, **15** (2014). https://doi.org/10.1186/1471-2105-15-S8-S3
5. W. Kabsch, C. Sander, Dictionary of protein secondary structure, *Biopolymers*, **22** (1983), 2577–2637.
6. B. Rost, C. Sander, Combining evolutionary information and neural networks to predict protein secondary structure, *Proteins.*, **19** (1994), 55–72. https://doi.org/10.1002/prot.340190108
7. Y. Yang, J. Gao, J. Wang, R. Heffernan, J. Hanson, K. Paliwal, et al., Sixty-five years of the long march in protein secondary structure prediction: The final stretch?, *Brief. Bioinform.*, **19** (2018), 482–494. https://doi.org/10.1093/bib/bbw129
8. Y. Ma, Y. Liu, J. Cheng, Protein secondary structure prediction based on data partition and semi-random subspace method, *Sci. Rep.*, **8** (2018), 1–10. https://doi.org/10.1038/s41598-018-28084-8

9. M. Lasfar, H. Bouden, A method of data mining using hidden markov models (HMMs) for protein secondary structure prediction, *Procedia Comput. Sci.*, **127** (2018), 42–51. https://doi.org/10.1016/j.procs.2018.01.096
10. A. Drozdetskiy, C. Cole, J. Procter, et al. JPred4: A protein secondary structure prediction server, *Nucleic Acids Res.*, **43** (2015), 389–394. https://doi.org/10.1093/nar/gkv332
11. D. T. Jones, Protein secondary structure prediction based on position-specific scoring matrices, *J. Mol. Biol.*, **292** (1999), 195–202. https://doi.org/10.1006/jmbi.1999.3091
12. A. Busia, N. Jaitly, Next-step conditioned deep convolutional neural networks improve protein secondary structure prediction, preprint, arXiv: 2017:1702.0386.
13. B. Z. Zhang, J. Y. Li, Q. Lü, Prediction of 8-state protein secondary structures by a novel deep learning architecture, *BMC Bioinformatics*, **19** (2018), 1–13. https://doi.org/10.1186/s12859-018-2280-5
14. S. Krieger, J. Kececioglu, Boosting the accuracy of protein secondary structure prediction through nearest neighbor search and method hybridization, Bioinformatics, **36** (2020). https://doi.org/10.1093/bioinformatics/btaa336
15. M. R. Uddin, S. Mahbub, Saifur Rahman, M., Bayzid, M.S. SAINT: Self-attention augmented inception-inside-inception network improves protein secondary structure prediction, *Bioinformatics*, **36** (2020), 4599–4608. https://doi.org/10.1093/bioinformatics/btaa531
16. K. Kotowski, T. Smolarczyk, I. Roterman-Konieczna, K. Stapor, ProteinUnet-An efficient alternative to SPIDER3-single for sequence-based prediction of protein secondary structures, *J. Comput. Chem.*, **42** (2021), 50–59. https://doi.org/10.1002/jcc.26432
17. P. M. Sonsare, C. Gunavathi, Cascading 1D-convnet bidirectional long short term memory network with modified COCOB optimizer: A novel approach for protein secondary structure prediction, *Chaos Soliton. Fract.*, **153** (2021), 111446. https://doi.org/10.1016/j.chaos.2021.111446
18. M. J. Zvelebil, J. O. Baum, *Understanding Bioinformatics*, Garland Science, New York, 2007.
19. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial networks, *Commun. ACM*, **63** (2020), 139–144. https://doi.org/10.1145/3422622
20. R. Wang, X. Xiao, B. Guo, Q. Qin, R.Chen, An effective image denoising method for UAV images via improved generative adversarial networks, *Sensors*, **18** (2018), 1985. https://doi.org/10.3390/s18071985
21. S. Yu, H. Chen, E. B. Garcia Reyes, N. Poh, Gaitgan: Invariant gait feature extraction using generative adversarial networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, **2017** (2017), 30–37.
22. Y. Zhao, H. Zhang, Y. Liu, Protein secondary structure prediction based on generative confrontation and convolutional neural network, *IEEE Access*, **8** (2020), 199171–199178. https://doi.org/10.1109/ACCESS.2020.3035208
23. L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Method. Appl. M.*, **376** (2021), 113609. https://doi.org/10.1016/j.cma.2020.113609
24. L. Abualigah, A. Diabat, P. Sumari, A. H. Gandomi, Applications, deployments, and integration of internet of drones (iod): A review, *IEEE Sens. J.*, **21** (2021), 25532–25546. https://doi.org/10.1109/JSEN.2021.3114266

25. L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, A. H. Gandomi, Reptile search algorithm (rsa): A nature-inspired meta-heuristic optimizer, *Expert Syst. Appl.*, **191** (2022), 116158. https://doi.org/10.1016/j.eswa.2021.116158
26. A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, A. H. Gandomi, Prairie dog optimization algorithm, *Neural Comput. Appl.*, **34** (2022), 20017–20065. https://doi.org/10.1007/s00521-022-07530-9
27. J. O. Agushaka, A. E. Ezugwu, L. Abualigah, Gazelle optimization algorithm: A novel nature-inspired metaheuristic optimizer, *Neural Comput. Appl.*, **35** (2023), 4099–4131. https://doi.org/10.1007/s00521-022-07854-6
28. L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-Qaness, A. H. Gandomi, Aquila optimizer: A novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.*, **157** (2021), 107250. https://doi.org/10.1016/j.cie.2021.107250
29. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in *International Conference on Machine Learning*, **70** (2017), 214–223.
30. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 770–778.
31. M. Farooq, A. Hafeez, Covid-resnet: A deep learning framework for screening of covid19 from radiographs, preprint, arXiv:2003.14395.
32. Z. Wu, C. Shen, A. Van Den Hengel, Wider or deeper: Revisiting the resnet model for visual recognition, *Pattern Recogn.*, **90** (2019), 119–133. https://doi.org/10.1016/j.patcog.2019.01.006
33. K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in *European Conference on Computer Vision*, Springer, Cham, (2016), 630–645.
34. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2015), 1–9.
35. G. Wang, R. L. Dunbrack, Pisces: Recent improvements to a PDB sequence culling server, *Nucleic Acids Res.*, **33** (2005), W94–W98. https://doi.org/10.1093/nar/gki402
36. J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, A. Tramontano, Critical assessment of methods of protein structure prediction (casp)-round x, *Proteins.*, **82** (2014), 1–6. https://doi.org/10.1002/prot.24452
37. J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, A. Tramontano, Critical assessment of methods of protein structure prediction: Progress and new directions in round xi, *Proteins.*, **84** (2016), 4–14. https://doi.org/10.1002/prot.25064
38. J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, A. Tramontano, Critical assessment of methods of protein structure prediction (casp)-round xii, *Proteins.*, **86** (2018), 7–15. https://doi.org/10.1002/prot.25415
39. A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, J. Moult, Critical assessment of methods of protein structure prediction (casp)-round xiii, *Proteins.*, **87** (2019), 1011–1020. https://doi.org/10.1002/prot.25823
40. A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, J. Moult, Critical assessment of methods of protein structure prediction (casp)-round xiv, *Proteins.*, **89** (2021), 1607–1617. https://doi.org/10.1002/prot.26237
41. J. A. Cuff, G. J. Barton, Evaluation and improvement of multiple sequence methods for protein secondary structure prediction, *Proteins.*, **34** (1999), 508–519.

42. S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller et al., Gapped blast and psi-blast: a new generation of protein database search programs, *Nucleic Acids Res.*, **25** (1997), 3389–3402. https://doi.org/10.1093/nar/25.17.3389
43. B. Rost, C. Sander, R. Schneider, Redefining the goals of protein secondary structure prediction, *J. Mol. Biol.*, **235** (1994), 13–26. https://doi.org/10.1016/S0022-2836(05)80007-5
44. Y. Guo, W. Li, B. Wang, H. Liu, D. Zhou, Deepaclstm: Deep asymmetric convolutional long short-term memory neural models for protein secondary structure prediction, *BMC bioinformatics*, **20** (2019), 1–12. https://doi.org/10.1186/s12859-019-2940-0
45. A. R. Ratul, M. Turcotte, M. H. Mozaffari, W. S. Lee, Prediction of 8-state protein secondary structures by 1D-Inception and BD-LSTM, BioRxiv, **2019** (2019), 871921. https://doi.org/10.1101/871921
46. Z. Li, Y. Yu, Protein secondary structure prediction using cascaded convolutional and recurrent neural networks, preprint, arXiv:1604.07176.
47. I. Drori, I. Dwivedi, P. Shrestha, J. Wan, Y. Wang, Y. He, et al., High quality prediction of protein q8 secondary structure by diverse neural network architectures, preprint, arXiv:1811.07143.

AIMS Press