Prediction of Protein Secondary Structure Based on WS-BiLSTM Model

Yang Gao, Yawu Zhao, Yuming Ma * and Yihui Liu *

School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250316, China; 1043119211@stu.qlu.edu.cn (Y.G.); B21070009@ups.edu.cn (Y.Z.)
* Correspondence: mym@qlu.edu.cn (Y.M.); yxl@qlu.edu.cn (Y.L.)

Abstract: Protein secondary structure prediction is an important topic in bioinformatics. This paper proposed a novel model named WS-BiLSTM, which combined the wavelet scattering convolutional network and the long-short-term memory network for the first time to predict protein secondary structure. This model captures nonlocal interactions between amino acid sequences and remembers long-range interactions between amino acids. In our WS-BiLSTM model, the wavelet scattering convolutional network is used to extract protein features from the PSSM sliding window; the extracted features are combined with the original PSSM data as the input features of the long-short-term memory network to predict protein secondary structure. It is worth noting that the wavelet scattering convolutional network is asymmetric as a member of the continuous wavelet family. The Q3 accuracy on the test set CASP9, CASP10, CASP11, CASP12, CB513, and PDB25 reached 85.26%, 85.84%, 84.91%, 85.13%, 86.10%, and 85.52%, which were higher 2.15%, 2.16%, 3.5%, 3.19%, 4.22%, and 2.75%, respectively, than using the long-short-term memory network alone. Comparing our results with the state-of-art methods shows that our proposed model achieved better results on the CB513 and CASP12 data sets. The experimental results show that the features extracted from the wavelet scattering convolutional network can effectively improve the accuracy of protein secondary structure prediction.

Keywords: protein; protein secondary structure; PSSM; wavelet scattering convolution network; long-short-term memory network

1. Introduction

Protein is an essential component of organisms, complete immunity, cellular signal transmission, and other functions. The in-depth study of proteins is beneficial to research more targeted diseases and drug research and development. With the completion of human genome sequencing, protein sequence databases have proliferated, and protein research has become an important task. Protein structure can be divided into primary, secondary, tertiary, and quaternary structures. Among them, the tertiary structure is biologically active.

Inspired by the great success in the fields of computer vision [1], speech recognition [2], and emotion classification [3], the method based on deep learning has been widely used in many biological research fields [4,5]. Examples include protein contact map [6], drug-target binding affinity [7,8], chromatin accessibility [9], protein function [10,11], and using Support Vector Machine (SVM) to solve the problem of protein structure prediction [12]. The main advantage of the deep learning method is that it can automatically represent the original sequence and learn hidden patterns through nonlinear transformation [13].

With the continuous development of deep learning, various neural network structures have been applied to predict protein secondary structures.

Sønderby and Winther [14] used Long-Short-Term Memory networks (LSTM) to predict protein secondary structure on the CB513 data set, and Conover et al. [15] also used the LSTM model for protein model evaluation. McGuffin et al. established a PSIPRED...
server to predict the protein secondary structure [16]. Ma et al. proposed a new protein secondary structure prediction method based on data Partition and Semi-Random Subspace Method (PSRSM), with a prediction accuracy of 84.53% on the test set CBS13 [17]. The Deep Inception-Inside-Inception (Deep3I) network is used in MOUFOLD [18]. It consists of two initial nested modules for convolution, a convolution layer, and a fully connected dense layer. Wang et al. proposed RaptorX to predict the protein secondary structure based on Deep Convolutional Neural Fields (DeepCNF). DeepCNF models the complex sequence-structure relationship through the deep hierarchical structure and the interdependence between adjacent attribute tags [19]. Spider [20] uses LSTM and the Bidirectional Recurrent Neural Network (BRNN) to predict protein secondary structure without using windows. SPOT-1D [21] uses the combination of Residual Convolution Network (ResNets) and LSTM-BRNN to identify and spread the short-term and long-short-term dependencies of the whole protein sequence.

In 2011, Mallat proposed a scattering operator based on wavelet transform [22]. This operator restores the lost high frequency of the image and extracts the required features while maintaining translation invariance and deformation stability. Based on this background, this paper proposes a protein secondary structure prediction method based on the wavelet scattering convolutional network and long-short-term memory network. Based on the characteristics of the scattering operator, the protein features were extracted and combined with the original protein features, which were sent into the long-short-term memory network according to sequence pattern to predict the secondary structure of the protein.

2. Protein Secondary Structure Prediction Model

The protein sequence is expressed based on the Position-Specific Scoring Matrix (PSSM) [23] of PSI-BLAST. The protein secondary structure prediction is completed by predicting the structure type corresponding to the amino acid residue of the protein sequence. Kabsch and Sander developed the Define Secondary Structure of Proteins (DSSP) algorithm that classified secondary structure into eight types: H (α-helix), G (3-10-helix), I (π-helix), E (β-strand), B (isolated β-bridge), T (turn), S (bend), and C (others) [24]. Protein secondary structure could be calculated using DSSP for proteins with already known structures, and it is usually applied to train data sets to establish class labels. In this experiment, G, H, I will be replaced by H, B, E will be replaced by E, and the others will be divided by C. It is worth noting that the test set used in this article is not included in the training set.

This paper proposes the WS-BiLSTM model to predict the secondary structure of proteins. First, the data are preprocessed, and the PSSM matrix is divided according to the sliding window size of 13 and 19, respectively. The PSSM matrix is sent to the WS-BiLSTM model, and the wavelet scattering convolution network is used to extract the protein features. The features obtained are combined with the PSSM sequence without sliding window processing, and sent to the long-short-term memory network in sequence mode for prediction. The prediction model is shown in Figure 1.

2.1. Data Set

In this paper, the CullPDB [25] data set is used as the training set of the model. The CullPDB data set contained 15,125 proteins with less than 25% sequence identity. We removed the same protein as the test set, leaving 14,199 proteins. The test set uses CASP class data sets, including CASP9, CASP10, CASP11, and CASP12 [26–28]. In addition, the CB513 [29] and PDB25 [30] data sets are also used as test sets of the model, in which the protein sequences of the test set and training set are not repeated. The number of protein sequences in the test set is shown in Table 1.
2.1. Data Set

In this paper, the CullPDB [25] data set is used as the training set of the model. The CullPDB data set contained 15,125 proteins with less than 25% sequence identity. We removed the same protein as the test set, leaving 14,199 proteins. The test set uses CASP class data sets, including CASP9, CASP10, CASP11, and CASP12 [26–28]. In addition, the CB513 [29] and PDB25 [30] data sets are also used as test sets of the model, in which the protein sequences of the test set and training set are not repeated. The number of protein sequences in the test set is shown in Table 1.

| Data Set  | Protein Sequence Number |
|-----------|-------------------------|
| CASP9     | 122                     |
| CASP10    | 99                      |
| CASP11    | 81                      |
| CASP12    | 19                      |
| CB513     | 513                     |
| PDB25     | 1672                    |

2.2. Wavelet Scattering Convolution Network

In 2012, Mallat proposed a scattering operator based on wavelet transform [22,31], which uses a wavelet scattering convolution network for texture segmentation and image feature extraction, and achieves good results. The wavelet scattering convolutional network is a scattering decomposition network composed of two complex-valued two-dimensional Morlet [32] filter banks and isotropic scale invariance. The Morlet wavelet is a continuous wavelet with time attributes and two essential wavelet scales: scale and displacement. Therefore, projecting protein data into the wavelet scattering convolution network can extract protein features with different scales and displacements that include time attributes.

The scattering wavelet can obtain the two-dimensional directional wavelet function by scaling and rotating the one-dimensional band-pass filtering function $\mu$. The multiresolution wavelet function can be obtained by scaling and rotating the function $\mu$ in binary:

$$\mu_\omega(x) = 2^{2i} \mu(2^ir^{-1}x)$$  \hspace{1cm} (1)

In the formula $\alpha = 2^ir \in \Lambda = D \times Z, i \in Z, r \in D$ ($D$ is a discrete, finite rotation group), $i$ determines the scale of $\mu(x)$ and $r$ determines the direction of $\mu(x)$. Then, the form of wavelet transform for high-frequency information $\theta(x)$ follows:

$$P_\alpha \theta(x) = \theta \ast \mu_\omega(x)$$  \hspace{1cm} (2)

Figure 1. Model structure.

Table 1. Number of protein sequences in the test set.
Therefore, the wavelet transform modulus is:

\[ W_\alpha \theta(x) = \| \theta * \mu_\omega(x) \| \]  

(3)

The wavelet extracts the high-frequency information \( \theta(x) \) in the protein and convolves with the scaling function \( \varphi_q(x) \) to obtain the low-frequency information of the protein.

\[ B_q \theta(x) = \theta(x) * \varphi_q(x) \], Of which \( \varphi_q(x) = 2^{-2q} \varphi(2^{-q}x) \)  

(4)

Thus the modulus operator of wavelet transform is:

\[ W_q \theta = \{ B_q \theta = \theta * \varphi_q, W_q \theta = \| \theta * \mu_\alpha \| \} \in \Lambda \]  

(5)

The use of wavelet transform modulus operator to act on protein data will lose some high-frequency information, thus reducing the amount of high-frequency protein information converted into low-frequency information. The extracted protein features are incomplete, affecting the results of protein prediction.

When we iterate over the \( W_\alpha \), the high-frequency information of the protein will be dispersed into different paths of \( e = \{ a_n \}_{1 \leq n \leq |e|} \) and the scatter propagator will be obtained:

\[ W(e)\theta = \cdots \theta * \mu_{a_1} * \mu_{a_2} \cdots * \mu_{a_{|e|}} \]  

(6)

The scattering operator is defined by \( W_q \theta = \theta * \varphi_q \) and \( W_\alpha \theta = \| \theta * \mu_\alpha \| \).

Defining (scattering operator) sequence \( e = \{ a_n \}_{1 \leq n \leq |e|} \) as the wavelet transform path, and the scattering operator is defined on all path sets \( e_q \) of the path \( e = \{ a_n = (i_n, \gamma_n) \}_{1 \leq n \leq |e|} \) (where \( i_n \prec\prec q \)):

\[ S_q(e)\theta = B_q S(e)\theta, \text{ Of which } S(e)\theta = \prod_{n=1}^{|e|} W_{a_n}\theta \]  

(7)

when \( e = 0, S(0)\theta = \theta \), thus \( S_q(0)\theta = B_q \theta \).

The scattering operator can also be marked as:

\[ S_q(e)\theta = \cdots \theta * \mu_{a_1} * \mu_{a_2} \cdots * \mu_{a_{|e|}} = H_q \]  

(8)

The wavelet scattering convolution network is a multilevel convolution network; the network flow is shown in Figure 2. The coefficients of ordinary convolution networks must be obtained by continuous learning, and the results are only distributed in the last layer. The difference of the wavelet scattering convolution network is that it has a fixed wavelet filter bank, the parameters do not need to be obtained through learning, and its output coefficients are distributed in each layer.

To extract protein features using a wavelet scattering convolution network, we need to divide PSSM into sliding windows with a size \( 20 * N \) and adjust the scattering wavelet coefficients of different scales and directions to obtain PSSM features. Because the energy features extracted from the wavelet scattering convolution network are mainly distributed in the first three layers of the scattering convolution network, we use the characteristics of the scattering operator to transform the high-frequency information of the first three layers into low-frequency information, collect and integrate the features of all directions and scales, and fuse with the original PSSM features, which will provide the input data for the long-short-term memory network.
2.3. Long-Short-Term Memory Network

Bidirectional Long-Short-Term Memory networks (Bi-LSTM) gradually developed from Recurrent Neural Network (RNN) and LSTM [33]. LSTM can remember the state of protein long-sequence data and solve possible problems such as gradient disappearance and gradient explosion, which RNN cannot solve. The long-short-term memory network structure is shown in Figure 3.

Figure 3. Network structure of long-short-term memory.

Bi-LSTM can be seen as a forward LSTM combined with a reverse LSTM. They take the beginning and end of the amino acid sequence as input, respectively. After passing through the Bi-LSTM network, the amino acid features obtained are integrated, spliced, and then sent to the classifier. Bi-LSTM trains the amino acid features extracted by the wavelet scattering convolution network according to sequence patterns. Bi-LSTM can consider fully the information of amino acids and their interaction and improve prediction accuracy. The schematic diagram of Bi-LSTM training is shown in Figure 4.

LSTM adds three types of gate structures based on RNN: forgetting gate, input gate, and output gate. This design makes the amino acid information in the cell state be selectively forgotten. The new amino acid information will be selectively recorded in the cell state, and the previous amino acid information will be saved in the hidden layer. The schematic diagram of the LSTM unit is shown in Figure 5.
The output $d_t$ of the amnesia gate is obtained by activating the hidden state $C_{t-1}$ of the last amino acid sequence and the amino acid data $X_t$ of this sequence. The input gate consists of two parts. The first part uses the sigmoid activation function; the output is $e_t$. The second part uses the tanh activation function, and the output is the multiplication of the output of the two results of $i_t$, to update the cell state. Both the forgetting gate and the input gate will act on the cell state $A_t$ to update the LSTM amino acid cell state. The hidden state $C_t$ of this amino acid sequence consists of the output amino acid sequence state $b_t$ and the cell state $A_t$. The mathematical expression of the LSTM unit follows:

\begin{align}
    d_t &= \rho(Z_d * X_t + Y_d * C_{t-1} + m_d) \\
    e_t &= \rho(Z_e * X_t + Y_e * C_{t-1} + m_e) \\
    i_t &= \rho(Z_i * X_t + Y_i * C_{t-1} + m_i) \\
    b_t &= \rho(Z_b * X_t + Y_b * C_{t-1} + m_b)
\end{align}

(9) 
(10) 
(11) 
(12)

where $Z$ is the input weight, $Y$ is the recursive weight, $m$ is the bias, $d$ represents the forgetting gate, $e$ represents the candidate gate, $i$ represents the input gate, $b$ represents the output gate, and $\rho$ represents the activation function. First, Formula (9) represents the output of the forgetting gate, Formula (10) represents the input gate to decide which values to update. The results of the two are multiplied, and then the addition operation is updated to the cell state $A_t$, as shown in Formula (13). Finally, the output gate determines the final
Finally, the neurons in the fully connected layer of Bi-LSTM are connected to the neurons in the upper layer, which are classified by the Softmax classifier. The probability of the three structures of C, E, and H, where the amino acid is located, is calculated to complete the classification.

2.4. Result Evaluation

Protein secondary structure prediction is usually evaluated by the accuracy of Q3 [17], which is simply the percentage of the protein sequence in which the secondary structure location is correctly predicted. The Segment of Overlap Score (SOV) [34] is widely used to evaluate protein secondary structure prediction. According to DSSP, G, H, and I are transformed into H, E, and B into E, and other structures are transformed into C. The calculation formula follows:

\[ Q_3 = \frac{S_C + S_E + S_H}{S} \times 100\% \]  

where \( S_C \) represents the number of class C protein structures accurately predicted, \( S_E \) represents the number of class E protein structures accurately predicted, and \( S_H \) represents the number of class H protein structures accurately predicted. \( S \) represents the total number of amino acids. The calculation results of the accuracy of each secondary structure follow:

\[ Q_i = \frac{S_i}{A_i}, \quad i \in \{C, H, E\} \]  

\( Q_i \) represents the accurate proportion of amino acids predicted in state I and \( A_i \) represents the total number of amino acids in state \( i \).

The calculation of SOV is a measure based on the ratio of overlapping fragments, which treats the predicted and observed results. Assuming that the observed sequence is recorded as the sequence predicted by \( S_m \), as \( S_n \), \( S_{mn} \) as the fragment with the same state as \( S_m \) and \( S_n \), then \( S_{mn} \) must contain a pair of overlaps and a helix. The length of \( S_m \) is marked as \( \text{length}(S_m) \), and the union set of \( S_m \) and \( S_n \) sequences in each pair is marked as \( \max(S_m, S_n) \), and the intersection set of \( S_m \) and \( S_n \) sequences is marked as \( \min(S_m, S_n) \). Based on this, the SOV calculation formula follows:

\[ \text{SOV} = \frac{100}{N_{SOV}} \sum_{S_{mn}} \left[ \frac{\min(S_m, S_n) + \rho(S_m, S_n)\text{length}(S_m)}{\max(S_m, S_n)} \right] \]  

Among the variables, the setting for \( \rho \) allows the change of the fragment at the edge of the protein structure, and the \( \rho(S_m, S_n) \) value conforms to the following definition:

\[ \rho(S_m, S_n) = \min \left\{ \begin{array}{c} (\max(S_m, S_n) - \min(S_m, S_n)) \\ \min(S_m, S_n) \\ \text{int}[\text{length}(S_m)/2] \\ \text{int}[\text{length}(S_n)/2] \end{array} \right\} \]  

3. Experimental Results

3.1. Hyperparameters of Wavelet Scattering Convolutional Network

In order to evaluate the accuracy of the proposed model and verify the effectiveness of the wavelet scattering convolutional network for feature extraction, two separate experiments were set up to predict the protein secondary structure. The first experiment used the WS-BiLSTM model to predict the secondary structure of the protein, and the
second experiment used the Bi-LSTM model to predict the secondary structure of the protein. The sliding window size in this article was 13 and 19, respectively. Aiming at the hyperparameters of the wavelet scattering convolutional network, we set up a comparative test on its image invariance scale, hoping to obtain the appropriate scale to obtain the best experimental results. The experimental results are shown in Tables 2 and 3.

Table 2. Impact of different image invariance scales on the accuracy of window 13.

| Scale | CASP9 | CASP10 | CASP11 | CASP12 | CB513 | PDB25 |
|-------|-------|--------|--------|--------|-------|-------|
| 5     | 83.10 | 84.16  | 82.43  | 83.09  | 84.75 | 83.53 |
| 9     | 83.28 | 84.09  | 83.22  | 83.96  | 85.56 | 83.36 |
| 13    | 84.80 | 84.96  | 84.01  | 84.17  | 85.87 | 84.24 |

Table 3. Impact of different image invariance scales on the accuracy of window 19.

| Scale | CASP9 | CASP10 | CASP11 | CASP12 | CB513 | PDB25 |
|-------|-------|--------|--------|--------|-------|-------|
| 5     | 84.7  | 85.27  | 84.04  | 84.63  | 85.68 | 85    |
| 9     | 84.95 | 85.53  | 84.75  | 85.01  | 85.92 | 85.49 |
| 13    | 85.03 | 85.72  | 84.59  | 84.96  | 86.03 | 85.5  |
| 19    | 85.26 | 85.84  | 84.91  | 85.13  | 86.1  | 85.52 |

It can be seen from Tables 2 and 3 that when the sliding window size is 19, the accuracy rate is higher because the more extensive the sliding window length, the more protein feature information can be obtained. When the sliding window is 19, as the scale of image invariance increases, the accuracy rate shows an upward trend. When the scale increases to the maximum of 19, the accuracy reaches the highest rate. In the process of extracting features by wavelet scattering convolutional network, the protein feature data obtains 19*19 spatial support through the scaling filter specified by the scale of 19, thereby obtaining the theoretically most extensive protein feature and obtaining the highest accuracy.

For the hyperparameters of Bi-LSTM, we adjusted the number of Bi-LSTM layers, hidden units, model learning rate, regularization coefficient, and dropout value. The experiment tuning process uses CASP10 as the test set.

3.2. Network Layers and Hidden Units

In Bi-LSTM, the number of Bi-LSTM layers and the number of hidden units affect the experimental results. The irrational number of hidden units lead to overfitting the model, and too many Bi-LSTM layers lead to network bloating and excessive time consumption. In the network structure design, we hope that the complexity of the design is reasonable and the model has good generalization under the appropriate training time. Therefore, we adjust the number of network layers and hidden units. The experimental results are shown in Tables 4 and 5.

Table 4. Effect of setting a layer of Bi-LSTM on the accuracy of Q3.

| Layers | Filters | Q3    |
|--------|---------|-------|
| 1      | 200     | 78.72 |
|        | 400     | 79.57 |
|        | 600     | 80.97 |
|        | 800     | 81.16 |
|        | 1000    | 81.43 |
|        | 1200    | 81.66 |
Comparing Tables 4 and 5, we find that Table 5 shows a higher accuracy rate than Table 4. It can be seen that the higher the number of layers, the higher the accuracy. However, it can be seen from Table 4 that the setting of the number of hidden units still has a significant influence on the accuracy, so the network takes the output of each time step of the first layer as the input of the corresponding time step of the next layer, which is used for deep-level feature learning. The experimental results show that the optimal combination can be reached under the combination of 1000 hidden units in the first layer and 1000 hidden units in the second layer. Considering the feasibility of the experiment and the complexity of the model, the final network structure is determined as two layers of Bi-LSTM and two layers of 1000 hidden units, respectively.

### 3.3. Learning Rate

Adam’s optimization algorithm [17] is used in the network model. Compared with the traditional stochastic gradient descent method, Adam can update variables according to the concuss of historical gradient and the actual historical gradient after filtering concussion, which is suitable for solving optimization problems with large-scale data and parameters. The initial value of the learning rate will significantly impact the results, so in this group of experiments, to determine the appropriate initial learning rate, as shown in Table 6 of the experimental results, the rough adjustment is first carried out with a 10-fold reduction. A fine adjustment is then carried out at an appropriate order of magnitude. The final learning rate is determined to be 0.0004, and the result can reach 84.04%.

### 3.4. Dropout

Due to the two-layer Bi-LSTM design adopted by the model in this paper, many hidden units were added. In order to avoid overfitting problems caused by excessive training time, we introduced a dropout layer after each layer of the Bi-LSTM model to partially inactivate neurons. We set the two dropout layers to the same value for synchronous adjustment. The experimental results are shown in Table 7. This works well when the two-layer dropout value is set synchronously to 0.2.
Table 7. Effect of setting different dropout values on the accuracy of Q3.

| Dropout | Q3     |
|---------|--------|
| 0.2     | 84.09  |
| 0.4     | 84     |
| 0.6     | 83.82  |
| 0.8     | 83.84  |

3.5. Regularization

In the long-short-term memory network, L2 regularization can further prevent the model from overfitting. L2 regularization is a penalty term added to the loss function, and the so-called penalty term refers to the sum of squares of each element in the weight vector and then finds the square root. First, the rough adjustment is made by ten times shrinkage, and then the fine adjustment is carried out after the range is determined. The experimental results are shown in Table 8. L2 regularization dramatically improves the experimental results. When L2 regularization is 0.000006, the result can reach 85.84% on CASP10.

Table 8. Effect of setting different L2 regularization coefficients on the accuracy of Q3.

| L2       | Q3     |
|----------|--------|
| 0.0000001| 84.09  |
| 0.000001 | 83.81  |
| 0.000002 | 85.2   |
| 0.000004 | 85.03  |
| 0.000006 | 85.84  |
| 0.000008 | 84.96  |
| 0.0001   | 84.16  |
| 0.001    | 83.58  |
| 0.001    | 83.51  |

3.6. Discussion

From Tables 2–8, the parameter settings in WS-BiLSTM have a more significant impact on the experimental results. It can be seen from Tables 4 and 5 that as the number of hidden units increases, the accuracy of Q3 also increases, increasing by about 3%. At the same time, the two-layer network is better than the single-layer network, and the accuracy is improved by 0.8%. In order to balance the complexity and performance of the model, we no longer superimpose the number of hidden units and the number of network layers. Compared with Tables 4 and 5, the improvement of Q3 accuracy in Table 6 is also significant. Based on the number of hidden units and model layers after optimization, the learning rate optimization increases by 1.56%. Different dropout values are set in Table 7, but there is almost no improvement based on the former. The optimization of the regularization in Table 8 increased the results again by 1.42%.

Last but not least, the adjustment of the parameters in the wavelet scattering convolutional network model is also essential. It can be seen from Tables 2 and 3 that as the sliding window increases, the experimental results generally increase by about 1%. This is because the field of view of the wavelet scattering convolutional network for extracting the local interaction ability of amino acids is increased. At the same time, when the image invariance scale is increased to 19, the Q3 accuracy has once again completed a leap, and the experimental results increased by about 1% again. A larger image invariance scale provides more local correlation between amino acid residues.

Table 9 shows the experimental results of 70% discount cross-validation using the CULLPDB data set in this experiment. The CULLPDB data set is divided into three parts, two as training sets and one as the test set. The average value of 70% discount cross-validation can reach 81.84%. It is worth noting that the optimized parameters derived in Sections 3.1–3.5 are used to perform cross-validation.
Table 9. CULLPDB trifold cross-validation predicted Q3 accuracy.

| Times | 1     | 2     | 3     | Average |
|-------|-------|-------|-------|---------|
| Q3    | 81.72 | 81.76 | 82.04 | 81.84   |

We only use a long-short-term memory neural network to predict protein structure, and the results are shown in Table 10.

Table 10. Bi-LSTM prediction accuracy of Q3.

| Methods | CASP9   | CASP10  | CASP11  | CASP12  | CB513   | PDB25   |
|---------|---------|---------|---------|---------|---------|---------|
| Bi-LSTM | 83.11   | 83.68   | 81.41   | 81.94   | 81.88   | 82.77   |

Table 11 lists the Q3 accuracy of the model on CASP9, CASP10, CASP11, CASP12, CB513, and PDB25. Comparing Tables 10 and 11, we find that the WS-BiLSTM model that integrates the wavelet scattering convolutional network and the long-short-term memory network is much better than using the long-short-term memory network alone. The Q3 accuracy on the test set CASP9, CASP10, CASP11, CASP12, CB513, and PDB25 reached 85.26%, 85.84%, 84.91%, 85.13%, 86.10%, and 85.52%, which are 2.15%, 2.16%, 3.5%, 3.19%, 4.22%, 2.75% higher, respectively, than using the long-short-term memory network alone. The reason may be that when we only use the Bi-LSTM model to predict protein secondary structure, Bi-LSTM can capture the most comprehensive distance-dependent information. It can be seen from Tables 3 and 4 that when the Bi-LSTM network is double-layered, and the number of hidden units is greater than or less than 1000, Bi-LSTM cannot capture more information about the residues in the protein sequence. Moreover, our WS-BiLSTM model, with the robust feature extraction ability of wavelet scattering convolutional network, can extract more local correlations between small amino acid residues and combine with Bi-LSTM to extract more abundant amino acid characteristics.

Table 11. Q3 accuracy under different data sets.

| Data Set | Q3     | QC     | QH     | SOV     |
|----------|--------|--------|--------|---------|
| CASP9    | 85.26  | 84.24  | 82.32  | 88.23   | 80.86   |
| CASP10   | 85.84  | 86.79  | 82.47  | 87.11   | 79.05   |
| CASP11   | 84.91  | 82.11  | 85.56  | 86.4    | 79.94   |
| CASP12   | 85.13  | 79.76  | 85.01  | 91.53   | 78.09   |
| CB513    | 86.1   | 86.26  | 81.7   | 88.78   | 78.9    |
| PDB25    | 85.52  | 84.22  | 82.98  | 88.69   | 80.45   |

This paper selects five prediction methods: RaptorX-SS8 [35], PSIPred [16], Jpred [36], DeepCNF [19], and MUFOOLD-SS [18], which are compared with this model on CASP10, CASP11, CASP12, and CB513. In contrast, RaptorX-SS8 uses conditional neural fields, PSIPred uses two layers of feedforward neural networks, Jpred uses two layers of artificial neural networks from the SNNS neural network package, DeepCNF is a combination of deep neural networks and conditional neural fields, and MUFOOLD-SS uses a deep neural network. The comparison results are shown in Table 12 and Figure 6.

Table 12. Q3 accuracy of this model and other five models.

| Methods      | CASP10 | CASP11 | CB513 | CASP12 |
|--------------|--------|--------|-------|--------|
| RaptorX-SS8  | 78.9   | 79.1   | 78.3  | -      |
| PSIPred      | 81.2   | 80.7   | 79.2  | 78.1   |
| Jpred        | 81.6   | 80.4   | 82.3  | -      |
| DeepCNF      | 84.4   | 84.7   | -     | -      |
| MUFOOLD-SS   | 85.98  | 85.39  | -     | 80.5   |
| Our Model    | 85.84  | 84.91  | 86.1  | 85.13  |
DeepCNF can model complex sequence–structure relationships through a deep hierarchical architecture and interdependencies between adjacent labels. In contrast, WS-BiLSTM integrates a wavelet scattering convolutional network and a long-short-term memory network. The same wavelet scattering convolutional network extracts features while modeling the interdependence between adjacent labels. The long-short-term memory network constructs a complex sequence–structure relationship similar to DeepCNF. Nevertheless, WS-BiLSTM better handles the relationship between the two, as seen from the comparison results. MUFOLD-SS uses a smaller convolution window than popular neural networks. However, through the stacking of deep convolution blocks, the network can also represent local and global contexts while maintaining efficient calculations. Although WS-BiLSTM is slightly behind in CASP10 and CASP11, because WS-BiLSTM has a more vital ability to extract advanced features, the nonlocal interaction of residues gives WS-BiLSTM a significant lead on CASP12.

4. Conclusions

Compared with only using the long-short-term memory network, our model has a more significant improvement in Q3 accuracy on the CASP9, CASP10, CASP11, CASP12, CB513, and PDB25 data sets. Compared with RaptorX-SS8, PSIpred, Jpred, DeepCNF, it improved the CASP10, CASP11, CASP12, and CB513 data sets. Compared with MUFOLD-SS, it also obtained good results on the CASP12 data set. Mallat [14] and others characterized three properties that deep learning architectures possess for extracting valuable features from data: multiscale contractions, linearization of hierarchical symmetries, and sparse representations. The wavelet scattering convolutional network exhibits all of these properties. The wavelet scattering convolutional network linearizes the small deformations between amino acid features, such as dilation, by separating the changes of different scales. Combining the characteristics of wavelet and deep learning to extract features makes the features extracted by wavelet scattering convolutional network more competitive in subsequent classification. Because of the design of the WS-BiLSTM model in this paper, the circulation form of the data features is from the partial sequence of amino acids to the...
complete sequence. This allows us to obtain the partial characteristic information of the protein and the general information of the protein sequence. It is more likely to improve the accuracy of our protein prediction. At the same time, the experimental results show that the model proposed in this paper is effective in predicting protein secondary structure.

In future work, we will apply WS-BiLSTM to predict other protein structure-related properties, such as backbone torsion angles, solvent accessibility, and protein order/disorder region. These predicted features are helpful for protein tertiary structure prediction and protein model quality assessment.

Author Contributions: Y.G., Y.Z. and Y.L. designed the network model structure; Y.G. carried out the experiment; Y.G. and Y.M. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This project was supported in part by the National Natural Science Foundation of China under grant 61375013 and the Natural Science Foundation of Shandong Province under grant ZR2013FM020.

Acknowledgments: The authors want to thank all the collaborators who have helped this project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Panda, B.; Majhi, B.; Thakur, A. An integrated-OFFT model for the prediction of protein secondary structure class. Curr. Comput.-Aided Drug Des. 2019, 15, 45–54. [CrossRef] [PubMed]
2. Cao, Y.; Park, S.J.; Im, W. A systematic analysis of protein–carbohydrate interactions in the Protein Data Bank. Glycobiology 2021, 31, 126–136. [CrossRef] [PubMed]
3. Dos Santos, C.; Gatti, M. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of the COLING 2014: The 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 69–78.
4. Lasfar, M.; Bouden, H. A method of data mining using Hidden Markov Models (HMMs) for protein secondary structure prediction. Procedia Comput. Sci. 2018, 127, 42–51. [CrossRef]
5. Zhang, B.; Li, J.; Lü, Q. Prediction of 8-state protein secondary structures by a novel deep learning architecture. BMC Bioinform. 2018, 19, 2931. [CrossRef]
6. Wang, S.; Sun, S.; Li, Z.; Zhang, R.; Xu, J. Accurate de novo prediction of protein contact map by ultra-deep learning model. PLoS Comput. Biol. 2017, 13, e1005324. [CrossRef]
7. Öztürk, H.; Özgür, A.; Ozkirimli, E. DeepDTA: Deep drug–target binding affinity prediction. Bioinformatics. Bioinformatics 2018, 34, i821–i829. [CrossRef]
8. Lee, I.; Nam, H. Identification of drug-target interaction by a random walk with restart method on an interactome network. BMC Bioinform. 2018, 19, 9–18. [CrossRef]
9. Min, X.; Zeng, W.; Chen, N.; Chen, T.; Jiang, R. Chromatin accessibility prediction via convolutional long short-term memory networks with k-mer embedding. Bioinformatics 2017, 33, i92–i101. [CrossRef]
10. Gligorijević, V.; Barot, M.; Bonneau, R. deepNF: Deep network fusion for protein function prediction. Bioinformatics 2018, 34, 3873–3881. [CrossRef]
11. Sun, T.; Zhou, B.; Lai, L.; Pei, J. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. BMC Bioinform. 2017, 18, 277. [CrossRef]
12. Kurniawan, I.; Haryanto, T.; Hasibuan, L.S.; Agmalaro, M.A. Combining PSSM and physicochemical feature for protein structure prediction with support vector machine. J. Phys. Conf. Ser. 2017, 835, 012006. [CrossRef]
13. Butt, A.H.; Rasool, N.; Khan, Y.D. A treatise to computational approaches towards prediction of membrane protein and its subtypes. J. Membr. Biol. 2017, 250, 55–76. [CrossRef] [PubMed]
14. Sønderby, S.K.; Winther, O. Protein secondary structure prediction with long short term memory networks. arXiv 2014, arXiv:1412.7828. [CrossRef]
15. Conover, M.; Staples, M.; Si, D.; Sun, M.; Cao, R. AngularQA: Protein model quality assessment with LSTM networks. Comput. Math. Biophys. 2019, 7, 1–9. [CrossRef]
16. McGuffin, L.J.; Bryson, K.; Jones, D.T. The PSIPRED protein structure prediction server. Bioinformatics 2000, 16, 404–405. [CrossRef]
17. Ma, Y.; Liu, Y.; Cheng, J. Protein secondary structure prediction based on data partition and semi-random subspace method. Sci. Rep. 2018, 8, 9856. [CrossRef]
18. Fang, C.; Shang, Y.; Xu, D. MUFOOLD-SS: New deep inception-inside-inception networks for protein secondary structure prediction. Proteins Struct. Funct. Bioinform. 2018, 86, 592–598. [CrossRef]
19. Wang, S.; Peng, J.; Ma, J.; Xu, J. Protein secondary structure prediction using deep convolutional neural fields. Sci. Rep. 2016, 6, 18962. [CrossRef]
20. Heffernan, R.; Yang, Y.; Paliwal, K.; Zhou, Y. 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. *Bioinformatics* 2017, 33, 2842–2849. [CrossRef]

21. Hanson, J.; Paliwal, K.; Liffin, T.; Yang, Y.; Zhou, Y. Improving Prediction of Protein Secondary Structure, Backbone Angles, Solvent Accessibility, and Contact Numbers by Using Predicted Contact Maps and an Ensemble of Recurrent and Residual Convolutional Neural Networks. *Bioinformatics* 2019, 35, 2403–2410. [CrossRef]

22. Mallat, S. Group invariant scattering. *Commun. Pure Appl. Math.* 2012, 65, 1331–1398. [CrossRef]

23. Jones, D.T. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* 1999, 292, 195–202. [CrossRef]

24. Mallat, S. Group invariant scattering. *Commun. Pure Appl. Math.* 2012, 65, 1331–1398. [CrossRef]

25. Cuff, J.A.; Barton, G.J. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins Struct. Funct. Bioinform.* 1999, 34, 508–519. [CrossRef]

26. Lindgren, R.; Sjöström, M.; Stenqvist, P.; Bäck, M.; Kjellander, M.; Käll, L. Enhancing protein secondary structure prediction with deep convolutional networks. *Bioinformatics* 2017, 33, 3737–3744. [CrossRef]

27. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* 1997, 9, 1735–1780. [CrossRef] [PubMed]

28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 2017, 60, 84–90. [CrossRef]

29. Yang, Y.; Gao, J.; Wang, J.; Heffernan, R.; Hanson, J.; Paliwal, K.; Zhou, Y. Sixty-five years of the long march in protein secondary structure prediction: The final stretch? *Brief. Bioinform.* 2018, 19, 482–494. [CrossRef] [PubMed]