BBA - Proteins and Proteomics 1868 (2020) 140504

Contents lists available at ScienceDirect

BBA - Proteins and Proteomics

journal homepage: www.elsevier.com/locate/bbapap

Tetramer protein complex interface residue pairs prediction with LSTM combined with graph representations

Daiwen Sun[a], Xinqi Gong[a,b,*]

[a] Mathematics Intelligence Application LAB, Institute for Mathematical Sciences, Renmin University of China, Beijing 100872, PR China
[b] Beijing Advanced Innovation Center for Structural Biology, Tsinghua Univeristy, Beijing 100091, PR China

ARTICLE INFO

Keywords:
Protein tetramers
Interface residue pair prediction
Graph
Long-Short Term Memory networks(LSTMs)

ABSTRACT

Motivation: Protein-protein interactions are important for many biological processes. Theoretical understanding of the structurally determining factors of interaction sites will help to understand the underlying mechanism of protein-protein interactions. Taking advantage of advanced mathematical methods to correctly predict interaction sites will be useful. Although some previous studies have been devoted to the interaction interface of protein monomer and the interface residues between chains of protein dimers, very few studies about the interface residues prediction of protein multimers, including trimers, tetramer and even more monomers in a large protein complex. As we all know, a large number of proteins function with the form of multibody protein complexes. And the complexity of the protein multimers structure causes the difficulty of interface residues prediction on them. So, we hope to build a method for the prediction of protein tetramer interface residue pairs.
Results: Here, we developed a new deep network based on LSTM network combining with graph to predict protein tetramers interaction interface residue pairs. On account of the protein structure data is not the same as the image or video data which is well-arranged matrices, namely the Euclidean Structure mentioned in many researches. Because the Non-Euclidean Structure data can't keep the translation invariance, and we hope to extract some spatial features from this kind of data applying on deep learning, an algorithm combining with graph was developed to predict the interface residue pairs of protein interactions based on a topological graph building a relationship between vertexes and edges in graph theory combining multilayer Long Short-Term Memory network. First, selecting the training and test samples from the Protein Data Bank, and then extracting the physicochemical property features and the geometric features of surface residue associated with interfacial properties. Subsequently, we transform the protein multimers data to topological graphs and predict protein interaction interface residue pairs using the model. In addition, different types of evaluation indicators verified its validity.

1. Introduction

Proteins are the principal catalytic agents, structural elements, signal transmitters, transporters and molecular machines in cells [1]. But individual proteins do not function alone; they must interact with other molecules to carry out their cellular roles. At present, there are some issues in the field of protein-protein interaction, and the study of protein-protein interaction structure is one of them. That is, we know the structure of two proteins and they will interact with each other, and we need to determine how they interact with each other in the atom level. The difficulty of this problem is about the same with protein folding. It's crucial to know protein-protein interaction interface binding sites (interface residue pairs) for comprehensively understanding molecular mechanism and confirming potential drug targets [2]. Besides, the prediction results of protein-protein interaction interface residue pairs can assist in predicting protein 3D structure.

There are many experimental methods to confirm protein-protein interaction interface residue pairs including X-ray crystallography and nuclear magnetic resonance (NMR). These experiments are extremely valuable and have contributed greatly to our knowledge of protein recognition mechanisms. However, technical challenges, such as difficulties in expressing and purifying aggregation-prone protein samples, obtaining high quality crystals, as well as the protein size constraints (for NMR), make such experiments both labor-intensive and time-consuming. Because high throughput experimental characterization of protein interfaces is not yet possible, reliable computational approaches to identify interfacial residues are especially valuable. Therefore, using deep learning methods to predict protein interaction interfaces has

[*] Corresponding author at: Mathematics Intelligence Application LAB, Institute for Mathematical Sciences, Renmin University of China, Beijing 100872, PR China.
E-mail address: xinqigong@ruc.edu.cn (X. Gong).

https://doi.org/10.1016/j.bbapap.2020.140504
Received 28 March 2020; Received in revised form 30 June 2020; Accepted 16 July 2020
Available online 24 July 2020

become an inevitable trend.

At present, the contact map of protein monomer and the technology of protein dimer interface prediction are quite mature. There are currently two major methods for predicting protein contact maps: Evolutionary Coupling Analysis (ECA) and machine learning methods. ECA method uses Multiple Sequence Alignments (MSAs) [3] to determine the correlation of co-evolutionary residue pairs based on the idea that adjacent residues are mutated and evolved in synchronization with function and structure. These methods benefit from the acquisition of protein sequence information over the past decade. Popular ECA methods include: CCMPred [4], FreeContact [5], GREMLIN [6], PlmDCA [7] and PSICOV [8]. In addition, there are many more accurate prediction methods based on machine learning. These methods predict the interface by learning the relationship between sequence-based features and data labels. Early machine learning methods include: Support Vector Machines (SVM) [9], SVMCon [10], SVNSEq. [11] and the recently-released R2C [12]. In addition, with the development of deep learning technology in recent years, some methods based on deep neural networks have emerged, such as RNN and Deep Belief Networks [13]. Such predictors include Betacon [14], CMAPPro [15], DeepConPred [16], NNCon [17] and ResNet [18].

However, one of the great challenges now facing us is the problem of protein polymer interface prediction. This is because a considerable part of protein is composed of multiple monomers. Due to the complex structure of protein polymers, the prediction of the interface residue pairs of the protein polymers by computational methods has important guiding significance for the experimental biologists to analyze the structure. In this paper, we did some work on predicting protein tetramer interface residue pairs, and set up a web server for others to use.

We defined the interface residue pair as follows. If the contact areas between two amino acids from two different monomers are not zero, we called these two residues in contact. And these two residues are interface residue pair.

2. Materials and methods

2.1. Dataset

In this paper, we find 107 protein tetramers in the following table satisfying our requirements in the Protein Data Bank. The requirements include the following several points: the number of chains is 4, the chain length is between 50 and 500, and the experimental method is X-ray. We divided the 107 tetramers in two parts randomly, three fifths of the tetramers (65 tetramers) into training set and the rest of the tetramers (42 tetramers) into test set. Besides, the training set is divided into five equal parts randomly in order to do five-fold cross-validation. (See Table 1.)

Table 2
Nine features used in this paper.

Features	Abbreviation	Software or Researchers
Absolute Exterior Solvent Accessible area	AESA	NACCES [19]
Relative Exterior Solvent Accessible area	RESA	NACCES
Exterior Contact area with other residues	EC	Qcontacts [20]
Interior Contact area	IC	Qcontacts
Exterior Void area	EV	NACCES, Qcontacts
Hydropathy index, version 1	H1	Jack Kyte et al. [21]
Hydropathy index, version 2	H2	David Eisenberg [22]
pKa1: computation	pKa1	PROPKA3.1 [23]
pKa2: standard	pKa2	PROPKA3.1

The first column is the names of features. The second column is the abbreviation of these features. The third column is the software to calculate features or researchers proposing features.

2.2. Features

Protein-protein interface residues have different values in some features, which can help us to distinguish interface residue pairs and non-interface residue pairs. We used nine features to describe each residue, some of which were proposed by our lab.(See Table 2.) These features include five geometric characteristics: Interior Contact area (IC), Exterior Contact area with other residues (EC), Exterior Void area (EV), Absolute Exterior Solvent Accessible area (AESA) and Relative Exterior Solvent Accessible area (RESA). The Interior Contact area is the contact area among atoms in one residue. The Exterior Contact area with other residues is the contact area among the atoms of aimed residue and other residues. The Exterior Void area is the area of the part of aimed residue, which doesn't contact with other residues. Absolute Exterior Solvent Accessible area is the surface area of a residue that is accessible to a solvent. Relative Exterior Solvent Accessible area of a protein residue is a measure of residue solvent exposure. It can be calculated by formula:

$$RESA = \frac{AESA}{\max AESA}$$

where max$AESA$ is the maximum possible exterior accessible surface area for the residue. Besides, we also employed other four features to describe a residue, such as Hydropathy Index (HI, two versions) and pKa (two versions). Fig. 1 shows three of the geometric features.

2.3. Methods

2.3.1. Regarding a protein residue as a graph

Every residue of a protein monomer can be affected by residues around it. In addition, we all know that a protein monomer has

Table 1
Dataset.

DataSet		PDB ID												
Training Set	Cross-Validation Set 1	1L3A	1U4F	1WYT	2A2U	2EPI	3IB6	3ITY	1REW	2Z8U	3SDL	1AGQ	2PBY	1B79
	Cross-Validation Set 2	1FS2	1INL	1Q15	2NLZ	3CKY	3IWV	3QBP	1A4Y	2ZIH	3KKK	1F5Z	2PK2	1I7X
	Cross-Validation Set 3	1E65	1LBI	1SWF	1SWH	2EP5	2QW6	3B8F	1GPQ	1QUQ	2ZME	1PV1	1U9Y	3G33
	Cross-Validation Set 4	1BML	1QYN	1UDD	2H3N	2OKA	2XHZ	3AUP	1BDF	2WBL	3CUQ	3KYH	2OGK	1Y14
	Cross-Validation Set 5	1J2W	1QSO	2ESN	2GAC	2WKC	3E5B	3TUO	1J1J	1UDR	2OZK	3Q2S	3AQQ	1M1L
Test Set		1C4P	1FTR	1FX3	1HXH	1OFT	1Q5V	1QVC	1TJV	1UFQ	1VL2	1X75	2E3D	2E6E
		2H8N	2JBR	2NQO	2Y32	3DFQ	3ESI	3G7K	3HM0	3OHP	3RD4	3V15	1IZ1	1JL2
		1KAM	1NSW	2GJD	2R90	2ZYZ	3CDK	3CO2	3DMP	3F6Z	1BV4	1YIF	2ACI	3BF0
		1P27	1WWH	2NNW										

We divided the training set into five parts. When training the model, we use data of four parts to train the model, and use the other one part to test the performance of the model. Each cross-validation set is used to test the model. Then we calculated the average performance of the model. After that, we used all the training data to train the model, and used the test set data to test the model.

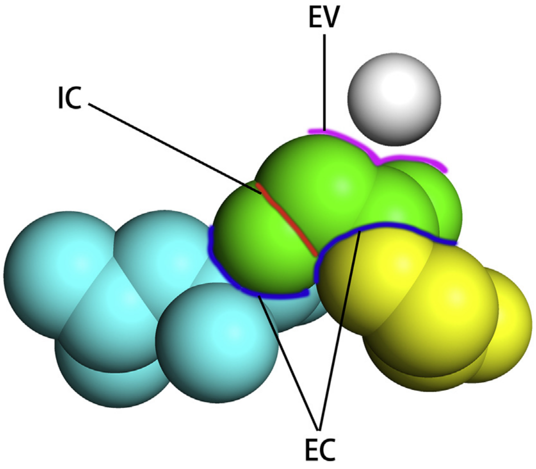


Fig. 1. Schematic diagram of EC, IC and EV. We used the features mentioned previously to discriminate between interface residue pairs and non-interface residue pairs.

sequence structure and three-dimensional folding structure. That means a residue can be affected by residues close in space, not only by residues near in the sequence order. For each residue, we can find several residues of this monomer, which have shortest Euclidean distances between them and the aimed residue. We took these residues as the neighbors of the aimed residue. At this time, we can make a simple graph for each residue. The graph has a center node, which is the aimed residue. Other nodes of the graph are neighbor nodes. There are edges between center node and neighbor nodes. Every node of the graph is represented by feature vector, which contains nine features mentioned previously. Every edge of the graph is represented by the distance between nodes, which is the distance of residues. Fig. 2 shows the process of regarding a protein residue as a graph.

2.3.2. Long short-term memory networks

Long short-term memory (LSTM) is an artificial recurrent neural network architecture [24] used in the field of deep learning. Unlike standard feedforward neural networks, LSTMs has feedback connections that make it a “general purpose computer” (that is, it can compute anything that a Turing machine can) [25]. It can not only process single data points, but also entire sequences of data.

In response to the problem of gradient vanishing and gradient exploding problem in RNN (Recurrent Neural Network), Hochreiter and Schmidhuber improved its hidden layer in 1997 and invented the LSTM neural network. LSTM introduces memory cells to replace hidden nodes in traditional RNN. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out the cell. The structure diagram of LSTM neural network is shown in Fig. 3.

In the LSTM neural network, it is divided into long-term memory and short-term memory. Long-term memory runs through the network. In each memory unit, the forget gate determines the part to be forgotten from the long-term memory, the input gate determines to obtain the information from the input and short-term memory and update it to the long-term memory, and the output gate determines the output information from the input and the short-term memory. Finally, the output information from long-term memory and output gate are combined to obtain a new short-term memory, which is also the output of the memory unit. LSTM memory unit structure diagram is shown in Fig. 4.

The description of LSTM memory unit:

In Fig. 4., C_t, h_t and X_t is the long-term memory, short-term memory and input at time t, respectively. σ is the sigmoid activation function and tanh is hyperbolic tangent activation function.

Step 1: Select the forgotten information. Through the forget gate, the short-term memory and input of the previous moment are stitched together, and we get f_t by sigmoid. Each element of the f_t vector is in the interval [0,1], where 0 represents all forgotten, and 1 represents all reserved.

$$f_t = \sigma(W_f \cdot [h_{t-1}, \mathrm{X}_t] + b_f)$$

Step 2: Select the information to be retained. In the input gate, the sigmoid function is used to decide which information to update (i_t), and the tanh function is used to generate the updated long-term memory information $\widetilde{C_t}$.

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i)$$

$$\widetilde{C_t} = \tanh(W_c \cdot [h_{t-1}, X_t] + b_c)$$

Step 3: Update the information in the memory unit. Multiply the f_t output from the forget gate by the long-term memory C_{t-1} in the previos layer to determine the part of long-term memory to be forgotten. Then multiply the update information generated by the second step $\widetilde{C_t}$ and *i_t* to determine which of the input information is retained. Then we can add the information from these two steps to get a new long-term memory.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t}$$

Step 4: Decide on output. First we use the sigmoid function to determine which state information needs to be output (o_t). Then the long-term memory is compressed between −1 and + 1 by tanh function, and multiplied by o_t obtained before to calculate the information to be output.

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

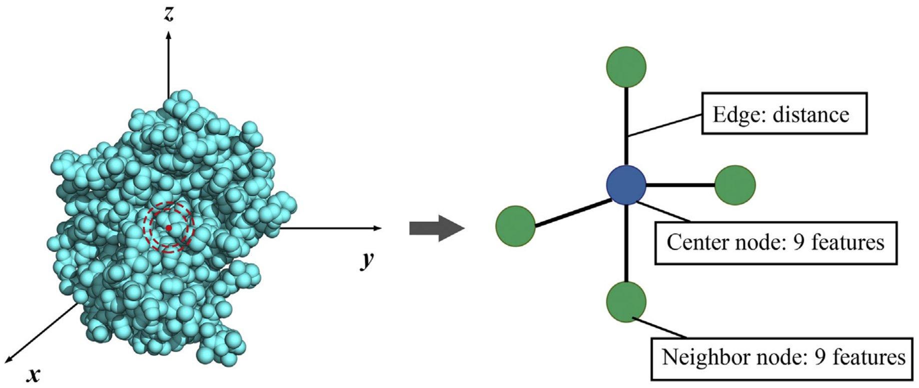


Fig. 2. Schematic diagram of graph representation.

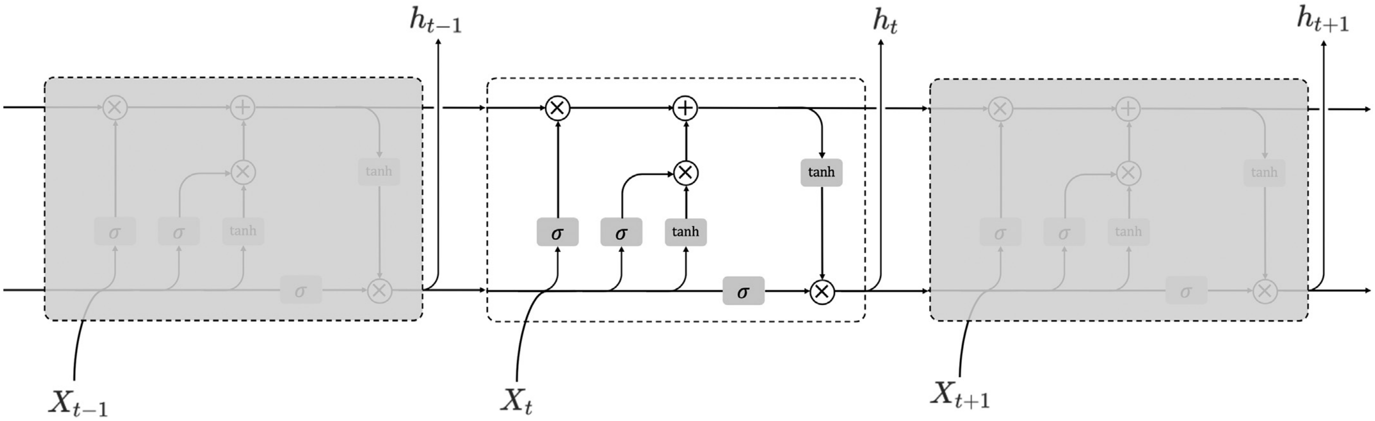


Fig. 3. The structure diagram of LSTM neural network.

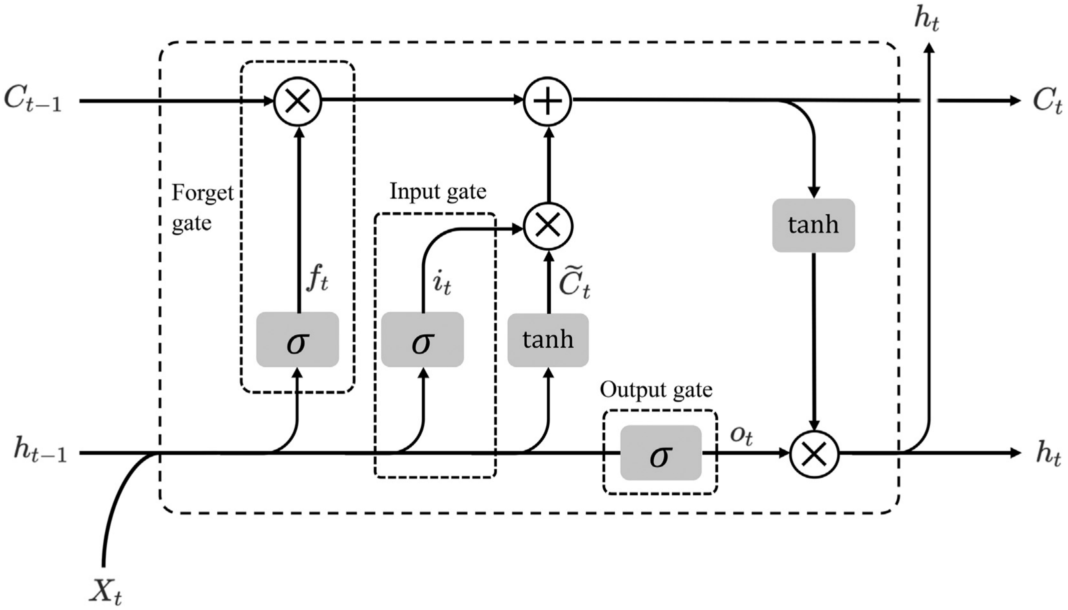


Fig. 4. The structure diagram of LSTM memory unit.

where, W_f, W_i, W_c, W_o and b_f, b_i, b_c, b_o are the weights and biases corresponding to each step respectively. The weights and biases of each step are not shared.

Through the previous description of LSTM Network, we can know that LSTM is not limited to the classification of time series data. It can process any sequence data. For example, a dataset X, of which sample space size is N. And the sample $\{x_1, x_2, \ldots, x_N\}$ is a series data, that is, there is a relationship between each sample and the samples before and after it. Such data is within the range that the LSTM model can handle.

2.3.3. Our model

In this paper, we proposed a method for predicting protein-protein interface residue pairs base on deep learning, by using protein geometric and physicochemical characteristic, and LSTM model combined with graph representation. The main ideas are as follows.

First, we downloaded protein tetramers from Protein Data Bank (PDB) and calculated geometric and physicochemical features of protein residues. Then, we found the k nearest residues of each residue, where k is a variable and we want to find the optimal value of k. We constructed each residue into a graph using the method mentioned above. Then we can represent a residue pair in two graphs and perform a convolution operation on each graph.

In detail, each node in the graph has the form: $x = (x_1, x_2, \ldots, x_n)$, where n is the number of features of nodes. We did a convolution operation on the graph, which can be expressed as:

$$z = \sigma\left(W^C x^c + \frac{1}{k}\sum_{i=1}^{k} W^N x_i^N + \frac{1}{k}\sum_{i=1}^{k} W^E E_i + b\right)$$

where x^C is the “center node”, x_i^N is the i$^{\text{th}}$ “neighbor node”, E_i is the i$^{\text{th}}$ edge, W^C, W^N and W^E is the weight matrix of center node, neighbor nodes and edges respectively, k is the number of neighbor nodes around a center node, b is a vector of biases, σ is a activation function. The dimensionality of the weight matrices is determined by the dimensionality of the inputs and the number of filters. And the weights and biases can be upgraded during the training process.

Next, we stitched the convolutional results together as a vector. Each concatenated vector represents a residue pair, and the residue pairs formed between every two interacting protein single chains constitute the residue pair sequence. We used the sequence formed by the residue pairs of all tetramers in the training set as the input of the LSTM model. At the end of the LSTM model, there are two fully connected layers. There are two neurons in the last fully connected layer, which can be used as the basis for binary classification. The number of these two neurons represent the probability of 0 and 1, respectively, that is, the probability of whether it is an interface residue pair. The schematic of the model is shown in Fig. 5.

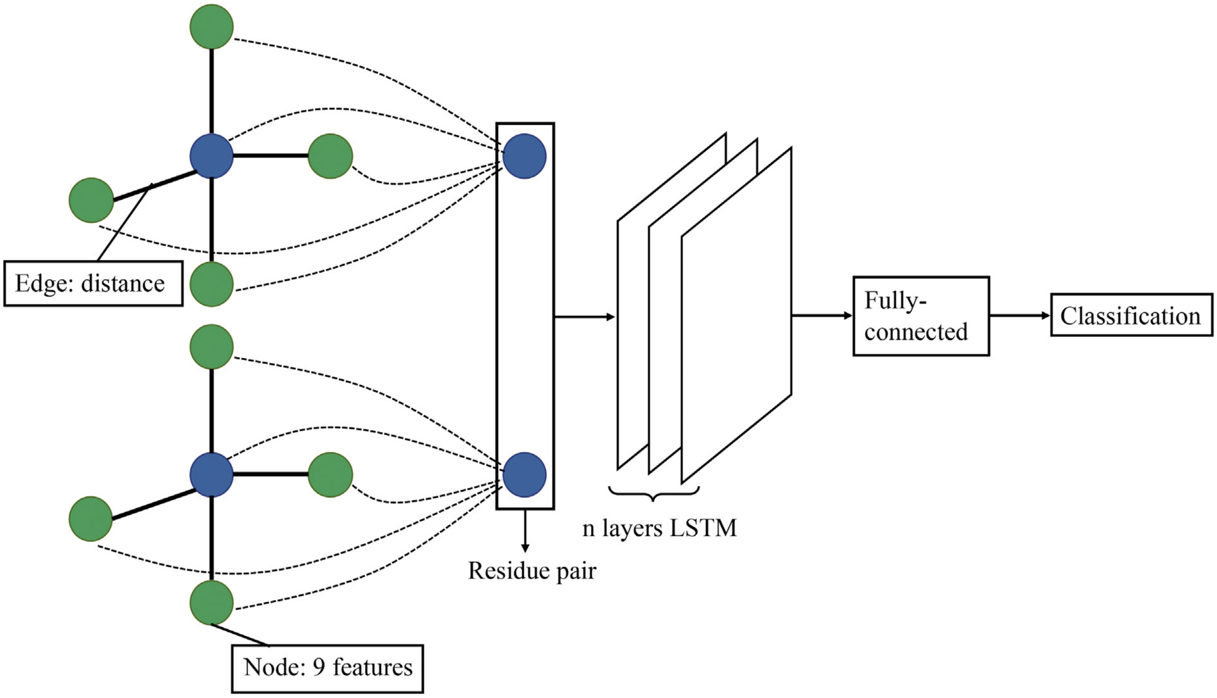


Fig. 5. Schematic of the model. Two residues from different chains were represented by two graphs. Then we concatenated the convolution results together as the input of the LSTM layers.

In this paper, there are two tunable parameters, the number of neighbors of aimed residues and the number of LSTM layers. Through training models, we selected models with suitable parameters which have good performances. Then we evaluated the performances of these selected models in test set by measures previously proposed. For example, we use Neighbor_3_LSTM_4 to indicate the parameters of the model: the number of neighbors of aimed residues is 3, the number of LSTM layers is 4.

2.3.4. Experiments

First, we extracted nine features of each residue in these tetramers. For each chain of a tetramer, we calculated the distance between each amino acid with other amino acids in the same chain. From that, we selected the nearest k amino acids as the k neighbors of this amino acid. In other words, we can take each amino acid and its k nearest neighbor amino acids as a graph. The nine features vector of each amino acid can be taken as the center node in a graph. At the same time, the features vectors of its k nearest neighbor amino acids can be taken as k neighbor nodes of the center node in this graph. And the distances between each amino acid with its k nearest amino acids are features of the edges in a graph. Then, we use the graph to represent the center amino acid. To summarize, if a protein chain has n amino acids, we can get n graphs to represent this protein chain.

A tetramer has four protein chains. If we choose two chains as a pair, we can get six pairs of chains. In each pair of chains, two residues from different chains consist of a residue pair. We use two graphs of the residues to represent the residue pair. Then, enter these two graphs into the model mentioned above to get classification results.

We trained the model with the training dataset by five-fold cross-validation and get the average performance of the model, in order to select the optimal number of neighbors and LSTM layers. After obtaining the optimal parameters, we trained the model with all the training dataset and tested the performance of the model with the testing set data. The experiment flow chart is shown in Fig. 6.

2.3.5. Evaluation criteria

In this research, we are more concerned about how many of the possible interface residue pairs given by each tetramer are correct. For several protein monomers that interact, if we give more correct interface residue pairs, it is helpful for subsequent protein docking and biological experiments. Therefore, we proposed the following evaluation criteria, and the performances of methods were evaluated basically by these measures.

The Number of Positive interface Residue Pairs in top 10 predictions (NPRP), which NPRP is a 6-dimensional vector as: $NPRP = (n_1, n_2, n_3, n_4, n_5, n_6)$. n_i represents the number of the positive interface residue pairs of the i th possible interface.

The Rank of the First Positive Prediction was defined as follow: $RFPP(m, n) = k$, if m of tetramers have n interfaces satisfying at least one true positive interface residue pair among top k predictions. Thus, an ideal model would have $RFPP(100\%, 6) = 1$, i.e., in every possible interface of tetramers, the top prediction is interface residue pair.

the Number of Correctly Predicted Tetramers: $NCPT(k, n) = m$ means that m tetramers satisfy there are n interfaces have at least one true residue pair among the top k predictions of each possible interface.

$$Accuracy\ order = \frac{RFPP}{TNRP}$$

TNRP is the total number of residue pairs in one interface.

$$Accuracy\ rate\,(k, n) = \frac{NCPT\,(k, n)}{TNT} \times 100\%$$

TNT represents the total number of protein tetramers in the dataset.

In order to understand these evaluation criteria better, we give three examples to explain them. (See Fig. 7).

From the diagram, we can get:

For the first case, $NPRP = (3, 4, 0, 3, 0, 2)$, $\| NPRP\|_1 = 12$;
For the second case, $NPRP = (3, 2, 0, 1, 2, 1)$, $\| NPRP\|_1 = 9$;
For the third case, $NPRP = (1, 1, 1, 1, 1, 0)$, $\| NPRP\|_1 = 5$.

$$NCPT\,(1, 1) = 1,\ NCPT\,(3, 1) = 2,\ NCPT\,(5, 1) = 3;$$

$$NCPT\,(2, 2) = 1,\ NCPT\,(5, 2) = 2,\ NCPT\,(6, 2) = 3;$$

$$NCPT\,(3, 3) = 1,\ NCPT\,(6, 3) = 2,\ NCPT\,(8, 3) = 3.$$

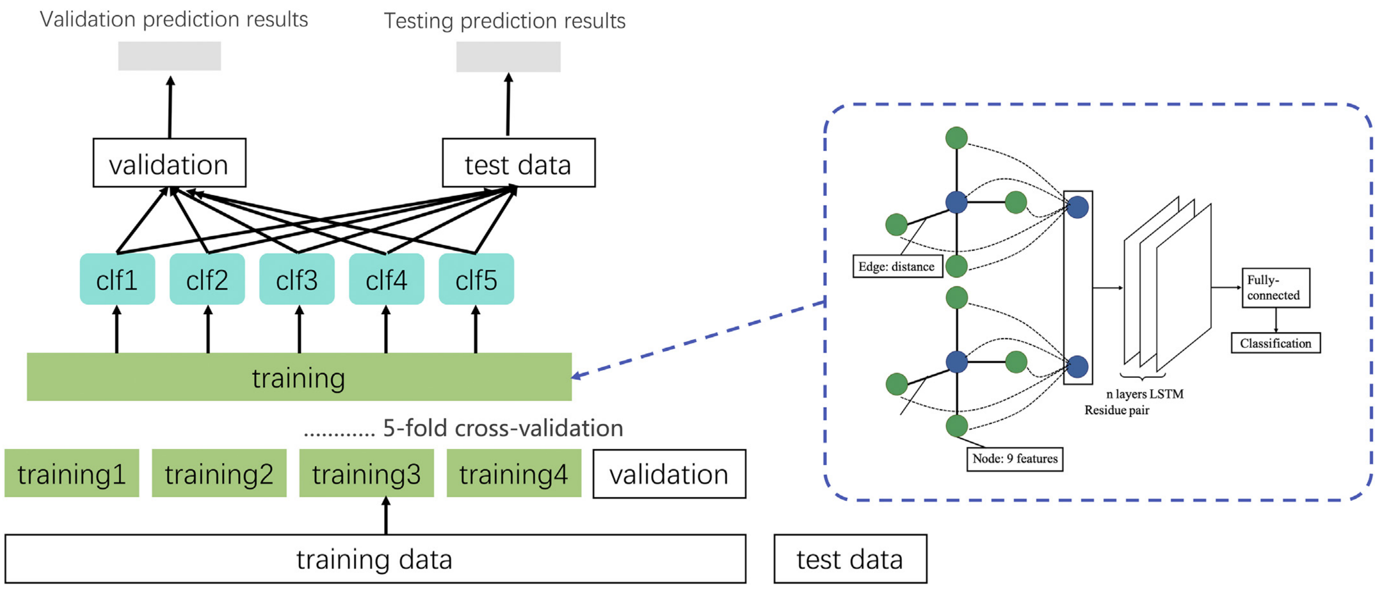


Fig. 6. The experiment flow chart.

1	2	3	4	5	6
1	0	0	0	0	0
0	1	0	0	0	0
1	1	0	0	0	1
0	0	0	0	0	0
0	0	0	1	0	0
0	0	0	0	0	0
0	1	0	1	0	0
1	0	0	0	0	1
0	0	0	0	0	0
0	1	0	1	0	0
0	0	0	1	0	1
1	0	0	1	0	0
0	0	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
0	0	1	0	0	1

a.

1	2	3	4	5	6
0	0	0	0	0	0
0	0	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
0	1	0	0	0	1
0	0	0	0	0	0
0	0	0	0	1	0
0	1	0	1	0	0
1	0	0	0	0	1
1	0	0	0	0	0
0	0	0	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
0	0	1	0	0	1
0	0	0	0	0	0

b.

1	2	3	4	5	6
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
1	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	0	0
1	0	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
0	0	1	0	0	1
0	0	0	0	0	0
0	0	0	0	0	0

c.

Fig. 7. Schematic diagram of prediction examples of three tetramers. Fig. 7 a. b. c. are three prediction results of three tetramers respectively. Each result has six columns, which represent the top 20 residue pairs from large to small in probability on each possible interface respectively. 0 or 1 indicates whether this residue pair are actually interface residue pair or non-interface residue pair.

$$RFPP(33.33\%,1) = 1,\ RFPP(33.33\%,2) = 2,\ RFPP(33.33\%,3) = 3;$$

$$RFPP(66.67\%,1) = 3,\ RFPP(66.67\%,2) = 5,\ RFPP(66.67\%,3) = 6;$$

$$RFPP(100\%,1) = 5,\ RFPP(100\%,2) = 6,\ RFPP(100\%,3) = 8.$$

$$\begin{aligned} \mathit{Accuracy\ rate}(1,1) &= 33.33\%,\ \mathit{Accuracy\ rate}(3,1) \\ &= 66.67\%,\ \mathit{Accuracy\ rate}(5,1) = 100\%; \end{aligned}$$

$$\begin{aligned} \mathit{Accuracy\ rate}(2,2) &= 33.33\%,\ \mathit{Accuracy\ rate}(5,2) \\ &= 66.67\%,\ \mathit{Accuracy\ rate}(6,2) = 100\%; \end{aligned}$$

$$\begin{aligned} \mathit{Accuracy\ rate}(3,3) &= 33.33\%,\ \mathit{Accuracy\ rate}(6,3) \\ &= 66.67\%,\ \mathit{Accuracy\ rate}(8,3) = 100\%. \end{aligned}$$

3. Results and discussion

3.1. The prediction results of the models

In order to fix the parameters and test the accuracy of the deep architecture, we applied this method to 65 proteins in our training set. We input two graphs of each protein residue pair and through the network, then we obtained the possibility of a residue pair to be an interface residue pair. We used index accuracy defined before to evaluate the network structure. We have provided the prediction results with the models which have different number of neighbors and LSTM layers.

From the Table 3, we can see that model Neighbor_3_LSTM_4 has the most correct number whether we take the top 10, top 20 or the top 100 results. This shows that this model can have more true interface residue pairs in giving prediction results. From another evaluation indicator in Table 4, the Neighbor_3_LSTM_4 model has smaller accuracy order. The smaller accuracy order, it means that in the given prediction results, the true interface residue pair appear earlier. Based on these two points, the Neighbor_3_LSTM_4 model is the best performing of the nine models.

We calculated the average value of each cross-validated accuracy order of each model, and calculated the average value of each group of

Table 3
Mean of $\|NPRP\|_1$ for each model.

	top10	top20	top100
Neighbor_3_LSTM_3	4.02	7.09	28.83
Neighbor_3_LSTM_4	**4.08**	**7.62**	**30.06**
Neighbor_3_LSTM_5	1.82	3.08	12.25
Neighbor_4_LSTM_3	3.34	6.92	28.29
Neighbor_4_LSTM_4	3.46	6.14	23.55
Neighbor_4_LSTM_5	2.37	4.60	17.46
Neighbor_5_LSTM_3	3.92	7.26	29.03
Neighbor_5_LSTM_4	1.69	3.00	11.45
Neighbor_5_LSTM_5	2.20	4.26	17.88

The indicator shows that each model takes the average number of correct predictions from the top 10, top 20, top 100 results.

Table 4
The accuracy order of these models.

Accuracy order(%)	Neighbor_3_LSTM_3	Neighbor_3_LSTM_4	Neighbor_3_LSTM_5	Neighbor_4_LSTM_3	Neighbor_4_LSTM_4	Neighbor_4_LSTM_5	Neighbor_5_LSTM_3	Neighbor_5_LSTM_4	Neighbor_5_LSTM_5
cv_1	3.3004	3.1483	2.3475	3.2406	3.1398	3.1327	3.0734	21.6662	3.5245
cv_2	2.4710	1.2816	1.4624	2.5062	1.3345	18.6690	2.5086	18.6690	0.9025
cv_3	0.3837	0.3319	31.2968	0.3245	0.3687	31.2390	0.5458	0.2848	31.2968
cv_4	0.7907	0.7168	21.9799	0.9347	1.4306	2.2711	0.7866	1.4189	1.4864
cv_5	0.3354	0.4258	27.7013	0.2382	27.7013	0.3009	0.3396	27.7013	27.7013
Average	1.4562	**1.1809**	16.9576	1.4488	6.7950	11.1225	1.4508	13.9480	12.9823

cross-validation to characterize the performance of each model. From the table, we can see the average accuracy order of Neighbor_3_LSTM_4 is the smallest, which indicates that this model performs best.

3.2. Analysis of the best model

From the above analysis, Neighbor_3_LSTM_4 has the best predictive ability. If we regard top 100 residue pairs of each chain pair as interface ones for this interface of the chain pair, we can correctly predict 69.23%, 92.31%, 84.62%, 84,62% and 100.00% tetramers in the five cross validations, which the model successfully predicted 3 interfaces, respectively. (See Table 5).

From Table 5, we can see that if we give top 10 prediction results, at least 80% of the proteins have at least one interface with the correct interface residue pairs in the result. And, 60% of the proteins meet the condition: in the top 10 predictions, at least two interfaces have the correct interfaces residue pairs. If the top 100 predictions are given, each protein will have at least one interface with the correct interface residue pairs. Among the top 100 predictions, the proportion of proteins with at least three interfaces having true interface residue pairs is as high as 86.15%.

3.3. Prediction results of the testing set with the best model

Next, we used the Neighbor_3_LSTM_4 model to make predictions on the testing set tetramers. In the testing set, there are 42 tetramers, including 24 tetramers with 6 interfaces, 11 tetramers with 5 interfaces, 4 tetramers with 4 interfaces and 3 tetramers with 3 interfaces. The accuracy rate of Neighbor_3_LSTM_4 model on testing set is shown in Table 6.

We did a statistic of tetramers which all the interfaces were predicted correctly by giving top 10, 20 and 100 residue pairs of each interface under Neighbor_3_LSTM_4 model in our testing set. The results are shown in the following Table 7.

Table 8 gives some examples on which the model performed well. "—" represents that the protein doesn't have the interface. From the Table 8, top 100 of the predicted results of protein 1FX3, 2H8N and 3ESI all include true interface residue pairs. And Fig. 8 shows two examples of prediction results.

3.4. Compared with random results

Actually, the prediction of protein tetramer interface residue pairs is a binary classification problem. A residue pair is an interface residue pair or not. The output result of our predictor is valued between 0 and 1, which shows the probability of interface residue pair. We sorted the possibilities from big to small, and the residue pairs with higgest possibility were regarded as interface residue pairs. Table 9. shows the specific predictions of the top 50,000 pairs of residue pairs as interface predictions in each cross-validation. We found that the recall, precision and F1 value is very low and the specificity and accuracy is very high. Analysis of the reason, we found that this is due to the face that the proportion of interface residues in the dataset is too low for all the residue pairs. There are 22,794,073 residue pairs in the training set, of which only 25,370 true interface residue pairs, which is 0.11% of all residue pairs. That means $TP + FN \ll TN + FP$. This makes the prediction quite difficult. Therefore, the value of $Recall = \frac{TP}{TP + FN}$ is very low.

We take the top 50,000 residue pairs predicted as interface residue pairs. According to the ratio of the interface residue pairs in the entire dataset to all the residue pairs, there are only 55 interface residue pairs in the 50,000 residue pairs, and the precision should be 0.0011, and our precision is greater than this value. Just because the interface residue pairs are too sparse, the precision is not high. Therefore, the F1 value determined by the recall and the precision is also low. Fig. 9 is the ROC

Table 5
The accuracy rate of Neighbor_3_LSTM_4 model on five cross validation sets.

	k = 10			k = 20			k = 100		
	n = 1	n = 2	n = 3	n = 1	n = 2	n = 3	n = 1	n = 2	n = 3
cv_1	84.62%	69.23%	30.77%	100.00%	76.92%	46.15%	100.00%	92.31%	69.23%
cv_2	84.62%	61.54%	15.38%	92.31%	76.92%	53.85%	100.00%	100.00%	92.31%
cv_3	76.92%	46.15%	30.77%	76.92%	76.92%	46.15%	100.00%	100.00%	84.62%
cv_4	69.23%	53.85%	30.77%	76.92%	61.54%	38.46%	100.00%	92.31%	84.62%
cv_5	84.62%	69.23%	38.46%	100.00%	92.31%	61.54%	100.00%	100.00%	100.00%
Average	80.00%	60.00%	29.23%	89.23%	76.92%	49.23%	100.00%	96.92%	86.15%

Table 6
The accuracy rate of Neighbor_3_LSTM_4 model on testing set.

	k = 10	k = 20	k = 100
n = 1	83.33%	92.86%	95.24%
n = 2	71.43%	85.71%	95.24%

Table 7
Tetramers in the testing set which all interfaces were predicted correctly.

top_100	1FX3	1OFT	2H8N	2Y32	3ESI	3HM0	1IZ1	1NSW
	2GJD	3CDK	3CO2	3DMP	1YIF	1P27	1WWH	
top_20	2Y32	1NSW	1P27	1WWH				
top_10	2Y32							

Table 8
The number of correctly predicted residue pairs in top 100 results of each interface.

		In1	In2	In3	In4	In5	In6
Proteins with 6interfaces	1FX3	17	7	3	2	7	17
	2H8N	8	10	4	6	9	8
	1QVC	18	11	0	1	10	14
	3ESI	17	6	6	8	6	17
Proteins with 5 interfaces	1NSW	26	–	2	4	16	24
	3CO2	3	2	12	17	–	3
	3DMP	17	1	–	2	8	18
Proteins with 4 interfaces	1BV4	23	–	–	0	1	21
Proteins with 3 interfaces	1WWH	22	–	–	–	11	14

curve of model Neighbor_3_LSTM_4 on the training set, and the AUC value is 0.890.

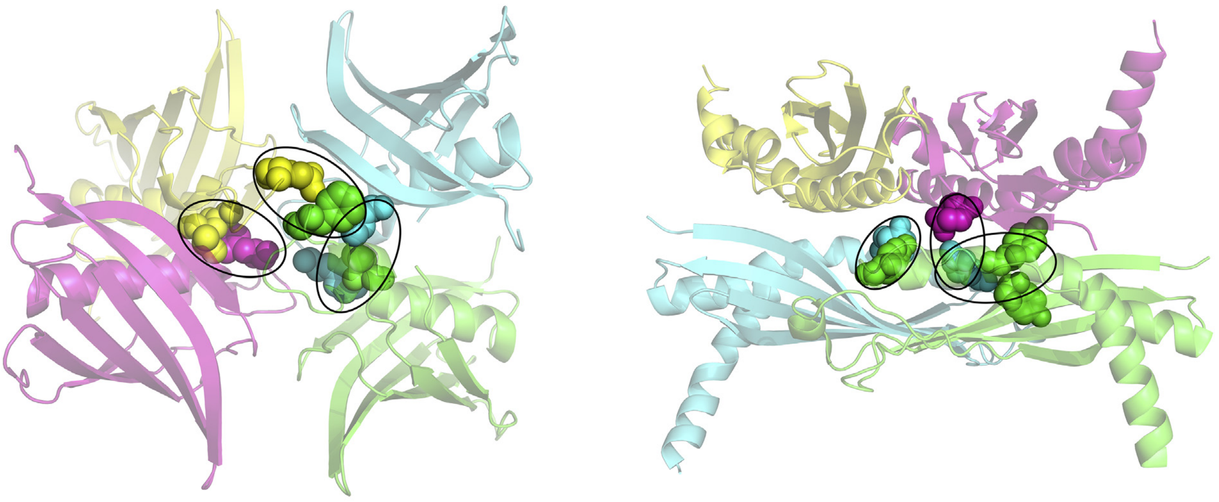
We assume that the stochastic prediction obeys the hypergeometric distribution [26]: $X\sim HG(N,M,K)$, where X is the number of real interface residue pairs in the top K predictions, N is the number of all the residue pairs in a interface of a tetramer, M is the number of real interface residue pairs in this interface. Then the probability of having x interface residue pairs in the K results of an interface given by the stochastic model is:

$$P(X=x)=\frac{\binom{M}{x}\binom{N-M}{K-x}}{\binom{N}{K}}$$

For six interfaces in a tetramer, we can calculate the probability of each interface choosing the top 100 residue pairs as $P_1, P_2, \ldots, P_6$. Then the probability that each tetramer can be predicted correctly three interfaces is

$$P = 1-\prod_{i=1}^{6}(1-P_i)-\sum_{i=1}^{6}P_i\prod_{\substack{j\in 1,\ldots,6\\ j\neq i}}(1-P_j)-\sum_{\substack{i,j\in 1,\ldots,6\\ i\neq j}}P_iP_j\prod_{\substack{k\in 1,\ldots,6\\ k\neq i\neq j}}(1-P_k)$$

We calculated the average prediction probability of 65 tetramers in the training set using the Monte Carlo simulation method to be 0.0928, which is a small number. This shows that random prediction is almost impossible to predict better than our method.



(a) 3ESI_ABCD

(b) 1FX3_ABCD

Fig. 8. A correctly predicted three-dimensional representation of the protein tetramer interface residue pairs.

Table 9
Results of the Neighbor_3_LSTM_4 in 5-fold cross-validation on the training set.

	TP	TN	FP	FN	Recall	Precision	Specificity	F1	Accuracy
Cross-validation1	1230	3,996,938	48,770	4644	0.209	0.025	0.988	0.044	0.987
Cross-validation2	1296	8,076,105	48,704	4332	0.230	0.026	0.994	0.047	0.993
Cross-validation3	1598	3,308,899	48,402	2606	0.380	0.032	0.986	0.059	0.985
Cross-validation4	1194	3,664,234	48,806	3392	0.260	0.024	0.987	0.044	0.986
Cross-validation5	1591	3,479,436	48,409	3487	0.313	0.032	0.986	0.058	0.985

3.5. Discussion

After the above analysis, we have reason to believe that our method can predict the interface residue pairs of protein tetramers on a computation level and achieve good results. At the same time, some experimental results can also support our prediction results. Here are some examples. (See Fig. 10)

For tetramer 2Y32 (PDB ID), it's mentioned in the experimental article that Tyr90 residue on each subunit is located in the center of the tetramer, which plays an important role in the formation and stability of the structure [27]. In our prediction results, we successfully predicted the Tyr90-Tyr90 residue pairs on all interfaces. For tetramer 3F6Z (PDB ID), on the dimer interface of the tetramer, L36 makes a close contact with the other surface formed by L34, L62, L69 and V81 [28]. In our prediction results, in the dimer formed by the B and D chains, the interface residue pairs L34-L34 and L34-L36 were successfully predicted. For tetramer 3 V15 (PDB ID), experiments have shown that the center of the dimers interface in the tetramer is methionine (M143) and a leucine (L221) [29]. We successfully predicted the residue pair M143-P139 on the first dimer interface and residue pair P139-M143 on the second dimer interface.

4. Conclusion

In this paper, we have developed a model for predicting protein tetramer interaction interface residue pairs. This method takes advantage of the physicochemical and geometric properties of amino acids as features, considering the effects of an amino acid and the surrounding amino acids, using LSTM neural networks combined with graph representation. In our test, the accuracy rate of successfully predicting one interface and two interfaces by giving 10 residue pairs were 83.33% and 71.43% respectively, which has a great guidance significant to biological experiment.

Availability of data and materials

Our code and parameters of model can be found in https://github.com/Sundw-818/Tetramer and full testing data is available in ftp://202.112.126.135/pub/Tetramer/.

Credit author

Xinqi Gong: Conceptualization, Data curation, Supervision, Fund

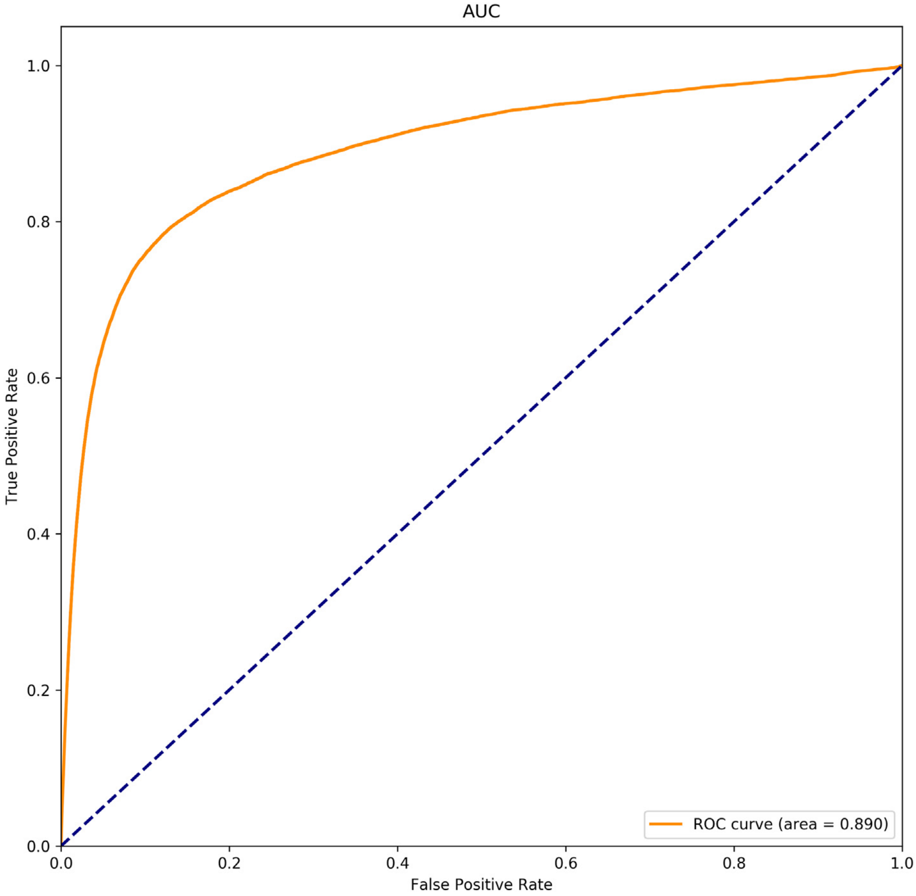


Fig. 9. The ROC curve of model Neighbor_3_LSTM_4 on the training set.

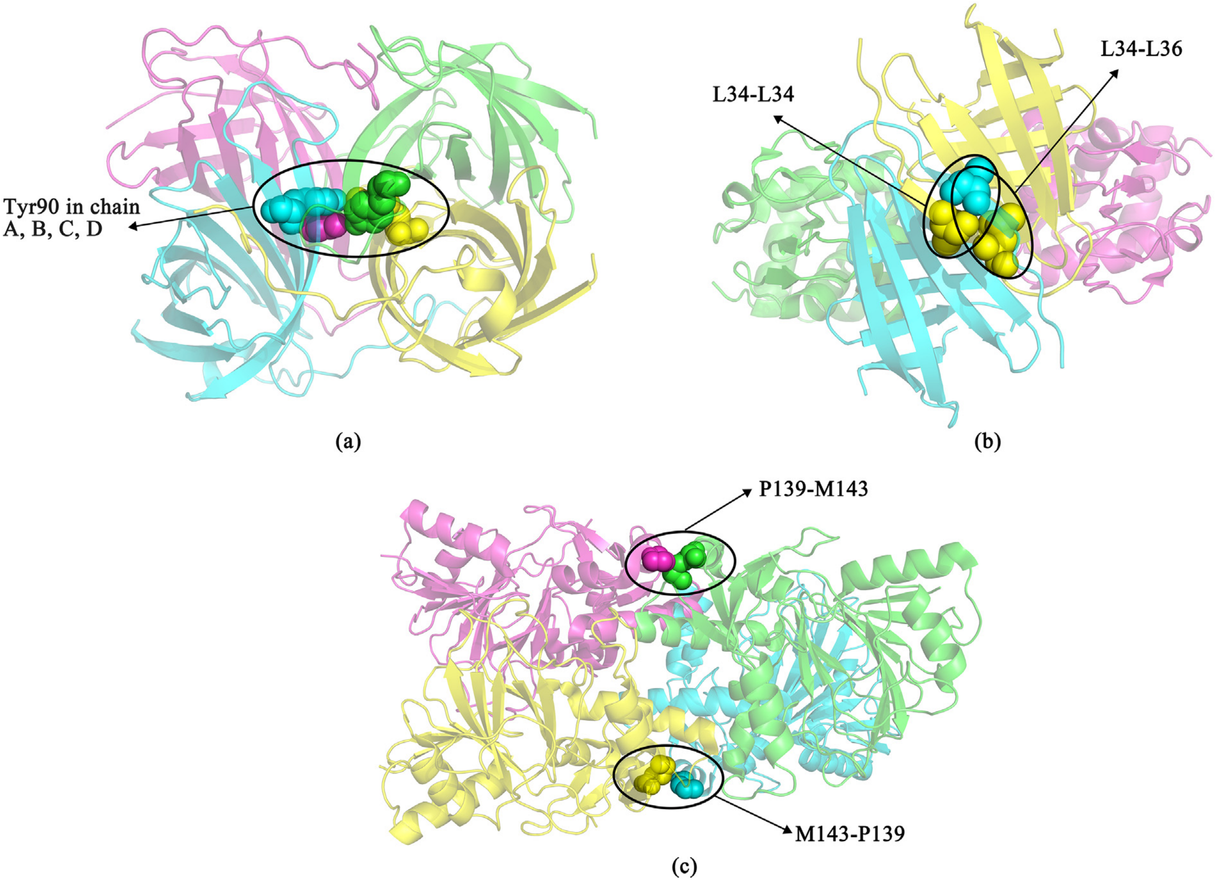


Fig. 10. The schematic diagram of the experimental results. Fig. 10 (a), (b) and (c) respectively show the experimental results of tetramer 2Y32, 3F6Z and 3 V15.

acquisition, Writing- Reviewing and Editing.

Daiwen Sun: Methodology, Visualization, Investigation, Software, Validation, Writing- Original draft preparation,

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We'd like to thank for National Natural Science Foundation of China (No. 31670725) for providing financial supports for this study.

References

[1] D.L. Nelson, A.L. Lehninger, M.M. Cox, W.H. Freeman (Ed.), Lehninger principles of biochemistry[M]. Lehninger principles of biochemistry, 2008, pp. 947–948.

[2] G. Sudha, R. Nussinov, N. Srinivasan, An overview of recent advances in structural bioinformatics of protein-protein interactions and a guide to their principles[J], Prog. Biophys. Mol. Biol. 116 (2–3) (2014) 141.

[3] U. Göbel, C. Sander, R. Schneider, et al., Correlated mutations and residue contacts in proteins[J], Proteins: Structure, Function, and Bioinformatics 18 (4) (1994) 309–317.

[4] S. Seemayer, M. Gruber, J. Söding, CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations[J], Bioinformatics 30 (21) (2014) 3128–3130.

[5] L. Kaján, T.A. Hopf, M. Kalaš, et al., FreeContact: fast and free software for protein contact prediction from residue co-evolution[J], BMC bioinformatics 15 (1) (2014) 85.

[6] H. Kamisetty, S. Ovchinnikov, D. Baker, Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era[J], Proc. Natl. Acad. Sci. 110 (39) (2013) 15674–15679.

[7] M. Ekeberg, C. Lövkvist, Y. Lan, et al., Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models[J], Phys. Rev. E 87 (1) (2013) 012707.

[8] D.T. Jones, D.W.A. Buchan, D. Cozzetto, et al., PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments[J], Bioinformatics 28 (2) (2011) 184–190.

[9] V. Vapnik, The Nature of Statistical Learning Theory[M], Springer science & business media (2013).

[10] J. Cheng, P. Baldi, Improved residue contact prediction using support vector machines and a large feature set[J], BMC bioinformatics 8 (1) (2007) 113.

[11] S. Wu, Y. Zhang, A comprehensive assessment of sequence-based and template-based methods for protein contact prediction[J], Bioinformatics 24 (7) (2008) 924–931.

[12] J. Yang, Q.Y. Jin, B. Zhang, et al., R2C: improving ab initio residue contact map prediction using dynamic fusion strategy and Gaussian noise filter[J], Bioinformatics 32 (16) (2016) 2435–2443.

[13] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets [J], Neural Comput. 18 (7) (2006) 1527–1554.

[14] J. Cheng, P. Baldi, Three-stage prediction of protein β-sheets by neural networks, alignments and graph algorithms[J], Bioinformatics 21 (suppl_1) (2005) i75–i84.

[15] P. Di Lena, K. Nagata, P. Baldi, Deep architectures for protein contact map prediction[J], Bioinformatics 28 (19) (2012) 2449–2457.

[16] D. Xiong, J. Zeng, H. Gong, A deep learning framework for improving long-range residue–residue contact prediction using a hierarchical strategy[J], Bioinformatics 33 (17) (2017) 2675–2683.

[17] A.N. Tegge, Z. Wang, J. Eickholt, et al., NNcon: improved protein contact map prediction using 2D-recursive neural networks[J], Nucleic Acids Res. 37 (suppl_2) (2009) W515–W518.

[18] S. Wang, S. Sun, Z. Li, et al., Accurate de novo prediction of protein contact map by ultra-deep learning model[J], PLoS Comput. Biol. 13 (1) (2017) e1005324.

[19] S.J. Hubbard, J.M. Thornton, Naccess[J]. Computer Program, Department of Biochemistry and Molecular Biology, University College London, 1993, p. 2(1).

[20] T.B. Fischer, J.B. Holmes, I.R. Miller, et al., Assessing methods for identifying pair-wise atomic contacts across binding interfaces[J], J. Struct. Biol. 153 (2) (2006) 103–112.

[21] J. Kyte, R.F. Doolittle, A simple method for displaying the hydropathic character of a protein[J], J. Mol. Biol. 157 (1) (1982) 105–132.

[22] R. Lüthy, D. Eisenberg, Protein[M]//Sequence Analysis Primer, Palgrave Macmillan, London, 1991, pp. 61–87.

[23] C.R. Søndergaard, M.H.M. Olsson, M. Rostkowski, et al., Improved treatment of ligands and coupling effects in empirical calculation and rationalization of p K a values[J], J. Chem. Theory Comput. 7 (7) (2011) 2284–2295.

[24] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[25] Hava T. Siegelmann, Eduardo D. Sontag, On the Computational Power of Neural Nets, ACM. COLT 92 (1992) 440–449.

[26] L.C. Xue, D. Dobbs, A.M. Bonvin, et al., Computational prediction of protein interfaces: a review of data driven methods[J], FEBS Lett. 589 (23) (2015) 3516–3526.

[27] J. Leppiniemi, T. Grönroos, J.A.E. Määttä, et al., Structure of Bradavidin–C-Terminal Residues Act as Intrinsic Ligands[J], PLoS One (2012) 7(5).

[28] S. Yum, M.J. Kim, Y. Xu, et al., Structural basis for the recognition of lysozyme by MliC, a periplasmic lysozyme inhibitor in gram-negative bacteria[J], Biochem. Biophys. Res. Commun. 378 (2) (2009) 244–248.

[29] S.H. Knauer, O. Hartl-Spiegelhauer, S. Schwarzinger, et al., The Fe (II)/α-ketoglutarate-dependent taurine dioxygenases from Pseudomonas putida and Escherichia coli are tetramers[J], FEBS J. 279 (5) (2012) 816–831.