PLOS ONE

RESEARCH ARTICLE

A novel matrix of sequence descriptors for predicting protein-protein interactions from amino acid sequences

Xue Wang[1,2,3], **Yuejin Wu**[1,2], **Rujing Wang**[2,3], **Yuanyuan Wei**[1], **Yuanmiao Gui**[1,2]*

1 Institute of Technical Biology & Agriculture Engineering, Hefei Institutes of Physical Science, Chinese Academy of Sciences, HeFei City, AnHui Province, China, **2** University of Science and Technology of China, HeFei City, AnHui Province, China, **3** Institute of Intelligent Machine, Hefei Institutes of Physical Science, Chinese Academy of Sciences, HeFei City, AnHui Province, China

* smalltalkman@foxmail.com

Citation: Wang X, Wu Y, Wang R, Wei Y, Gui Y (2019) A novel matrix of sequence descriptors for predicting protein-protein interactions from amino acid sequences. PLoS ONE 14(6): e0217312. https://doi.org/10.1371/journal.pone.0217312

Editor: Jie Zhang, Newcastle University, UNITED KINGDOM

Received: January 28, 2019

Accepted: May 8, 2019

Published: June 7, 2019

Data Availability Statement: Supplementary data related to this article can be found in S1, S2, S3 and S4 Files. Supplementary experimental code and results related to this article can be found at https://github.com/smalltalkman/hppi-tensorflow.

Funding: This work is supported by grants from the National Natural Science Foundation of China (61773360 and 31671586).

Competing interests: The authors have declared that no competing interests exist.

Abstract

Protein-protein interactions (PPIs) play an important role in the life activities of organisms. With the availability of large amounts of protein sequence data, PPIs prediction methods have attracted increasing attention. A variety of protein sequence coding methods have emerged, but the training of these methods is particularly time consuming. To solve this issue, we have proposed a novel matrix sequence coding method. Based on deep neural network (DNN) and a novel matrix protein sequence descriptor, we constructed a protein interaction prediction model for predicting PPIs. When performed on human PPIs data, the method achieved an accuracy of 94.34%, a recall of 98.28%, an area under the curve (AUC) of 97.79% and a loss of 23.25%. A non-redundant dataset was used to evaluate this prediction model, and the prediction accuracy is 88.29%. These results indicate that the matrix of sequence (MOS) descriptor can enhance the predictive power of PPIs and reduce training time, which can be a useful complement for future proteomics research. The experimental code and experimental results can be found at https://github.com/smalltalkman/hppi-tensorflow.

Introduction

Protein-protein interactions (PPIs) are useful for elucidating the changing mechanisms of organisms in physiological or pathological conditions and are important for disease prevention and drug development. In the last decade, numerous methods for studying protein-protein interactions, such as yeast two-hybrid screens [1], hybrid approaches [2] and protein chips [3], have emerged. However, all of these experimental methods have the disadvantage of being time-consuming and costly. Therefore, using computational approaches to predict unknown PPIs has become an important research topic in bioinformatics. In recent years, many computer prediction methods have been proposed to predict PPIs based on a phylogenetic profile method [4], amino acid index distribution [5] and gene fusion events [6, 7].

However, these methods are not universal because the reliability of these methods depends on a priori information about the protein pairs.

In recent years, a large amount of protein sequence information has been accumulated, and numerous computer calculation methods and sequence-based methods have become more universal and acceptable [8–18], such as support vector machines (SVM) [8–10], Naïve Bayes [11, 12], decision trees [13–14], random forests [15–16], and deep learning [17–18]. From the above methods, the accuracy of PPIs prediction is not only related to machine learning methods but also to protein coding methods. Protein coding methods and classification algorithms are the core steps of PPI prediction and have become primary tasks of current life science research. Until now, many efficient protein coding methods have been proposed for inferring PPIs based on protein sequence, such as the conjoint triad method (CT) [19], the auto covariance method (AC) [20] and local descriptor (LD) [21]. Among them, the conjoint triad method (CT) [19] considers considered the order relationship of three amino acids. In such a protein coding method, the 20 amino acids are clustered into seven classes according to the dipoles and volumes of the side chains. Auto covariance (AC) [20] considers the order relationship of 30 amino acids. Local descriptor (LD) is an alignment-free approach, and its effectiveness depends largely on the underlying amino acid groups, and only considers the neighbouring effect of two adjacent types of amino acids [21].

Though the various methods described above for protein coding methods are useful, one of the drawbacks is that the order relationship of the entire amino acid sequence is not considered. To overcome this problem, we propose a sequence-based method based on a novel representation of the matrix of sequence (MOS). The MOS descriptor is first classified into 7 classes according to the successful use of classification in Shen et al. [20]. Then, we combine this classification with a novel representation of protein sequence descriptors. Next, we constructed a (deep neural network- matrix of sequence) DNN-MOS model by combining the DNN and MOS. Finally, we evaluated the performance of the DNN-MOS protein prediction model. When performed on human data, our method had an accuracy of 94.34%, a recall of 98.28%, an area under the curve (AUC) of 97.79% and a loss of 23.25%. To prove the effectiveness of MOS, we compared MOS with existing protein coding methods. We found that the MOS can greatly reduce the loss and training time, and the prediction performance is improved. Additionally, we found that MOS achieves better performance in other classifiers such as decision tree, k-neighbors and random forest.

Materials and methods

Data set construction

(1) Benchmark dataset: The benchmark PPIs dataset was used in our experiment, which was provided by Pan et al. [22]. Among this benchmark dataset, the positive samples were taken from the Human Protein Reference Database (HPRD) 2007 version, and the negative samples were taken from the Swiss Swiss-Prot database 57.3 version. These positive samples are usually verified by reliable methods [23–24]. The negative samples (non-interacting pairs of proteins) were generated by pairing proteins found in different subcellular locations, according to the following requirements [19, 25]: (1) the non-interactive pairs cannot appear in interacting data sets; (2) sequences annotated with ambiguous or uncertain subcellular location terms were excluded to construct the negative samples; (3) sequences annotated by two or more locations were excluded due to lack of the uniqueness. After removing the self-interactions and duplicate interactions of the positive dataset, we finally obtained 36,630 positive pairs and 36,480 negative pairs. Protein pairs with unusual amino acids and <50 amino acids were excluded, such as B, J, O, U, X and Z to yield 36,591 positive samples and 36,324 negative

samples to form the benchmark dataset. We mixed the positive and negative samples in the benchmark dataset and randomly selected 60,000 pairs (30,000 positive samples, 30,000 negative samples as training datasets for models, with the remainder constituting the training set as a hold-out test set to validate the model).

(2) Non-redundant dataset: This dataset was provided by Pan et al. [22]. The protein pairs of this dataset exclude proteins with ≥25% sequence identity from the benchmark dataset. This dataset contains 3,899 positive protein pairs and 4,262 negative protein pairs.

Matrix of sequence (MOS)

Classification of amino acids. According to Shen et al. [19], 20 amino acids can be divided into seven different groups based on their dipole and side chain volumes. The seven different amino acid classifications are shown in Table 1. Then, a protein sequence is represented by these seven groups according to Table 1. For example, the protein sequence "AGCRQTSPLGVKSE" would be represented as "11754332211536".

Related definitions. Vector of protein sequence (VOS): Hypothetical non-empty finite set: $\Omega = \{w_1, \ldots, w_7\}$, where w_i is amino acid classification. Given sequence: $S = S_1, S_2, \ldots, S_L$, where L represents the length of sequence S, $S_i \in \Omega$, $1 \leq i \leq L$. The sequence vector of a given sequence S can be expressed as: $VOS = (C_{W_1}, \ldots, C_{W_N})$, where C_{W_1} is the number of occurrences of the w_i in the sequence S. Based on the definition of the sequence vector, the sum of all elements in the sequence matrix is equal to L.

Matrix of sequence (MOS): Hypothetical non-empty finite set: $\Omega = \{w_1, \ldots, W_N\}$, where N is the number of categories of the sequence. Given sequence: $S = S_1, S_2, \ldots, S_L$, where L represents the length of sequence S, $S_i \in \Omega$, $1 \leq i \leq L$. The sequence matrix of a given sequence S can be expressed as: $MOS = [m_{ij}]_{N \times N}$.

$$m_{ij} = \begin{cases} \ldots w_i \ldots or \ldots w_i \ldots w_i \ldots The\ \ number\ \ of\ \ occurrences, i = j \\ \ldots w_i \ldots w_j \ldots The\ \ number\ \ of\ \ occurrences, i \neq j \end{cases} \quad (1)$$

Based on the definition of the sequence matrix, the sum of all elements in the sequence matrix is equal to $\frac{L(L+1)}{2}$, $m_{ij} = \frac{C_{w_i}(C_{w_i}+1)}{2} (1 \leq i \leq N)$, $m_{ij} + m_{ji} = C_{w_i} C_{w_j} (i \neq j)$. Thus, for any two sequences, when the sequence lengths are different or the sequence lengths are the same but at least one element contains different numbers of elements, the corresponding sequence squares are different.

Algorithm of sequence matrix. Hypothetical non-empty finite set: $\Omega = \{w_1, \ldots, W_N\}$, where N is the number of categories of the sequence. Given sequence: $S = S_1, S_2, \ldots, S_L$, where L represents the length of sequence S, $S_i \in \Omega$, $1 \leq i \leq L$. The sequence matrix of a given sequence S can be expressed as:

Table 1. Amino acid classification based on their dipole and side chain volumes.

Number	Amino Acids
1	A, G, V
2	I, L, F, P
3	Y, M, T, S
4	H, N, Q, W
5	R, K
6	D, E
7	C

https://doi.org/10.1371/journal.pone.0217312.t001

Input sequence: S = S_1, S_2, ..., S_L;
Output sequence matrix: MOS = $[m_{ij}]_{N\times N}$.
The sequence matrix algorithm is calculated as follows:
Step 1. Initial value is set up: i ← L, VOS ← VOS_0 = 0, MOS ← MOS_0 = 0.
Step 2. VOS[s_i] ← VOS[s_i] + 1.
Step 3. MOS[s_i] ← MOS[s_i] + VOS.
Step 4. i ← i − 1.
Step 5. If i≥1, go to step 2.

Protein feature representation. In this article, we present a novel method of protein feature representation by combining sequence matrix descriptors with the amino acid classification method. To reduce the computational vector, we first classify 20 amino acids into 7 classes according to the amino acid classification method in Table 1. Thus, a protein sequence can be represented by a matrix of 7×7, as shown in Eq 2.

$$[m_{ij}]_{7\times 7} = \begin{Bmatrix} m_{11} & m_{12} \dots m_{17} \\ m_{21} & m_{22} \dots m_{27} \\ & \dots \\ m_{71} & m_{72} \dots m_{77} \end{Bmatrix} \tag{2}$$

The next step is to standardize m_{ij} of each matrix element ranging from 0 to 1. To solve this problem, we defined a new parameter p_{ij}, by normalizing m_{ij} with Eq 3:

$$p_{ij} = \frac{m_{ij}}{\sum m_{ij}} \tag{3}$$

$$\sum m_{ij} = \frac{L(L+1)}{2} \tag{4}$$

where L is the length of the protein sequence. The numerical value of p_{ij} of each protein ranges from 0 to 1. The elements in the diagonal of the matrix and the elements above the diagonal are combined into a 28-dimensional vector. To distinguish the lengths of the sequences, a sequence tag is added, and the sequence tags are represented by the reciprocal of the length of the protein. Finally, a total 29-dimensional vector has been built to represent each protein sequence.

Deep neural network (DNN)

A deep neural network is a popular type of deep learning algorithm with three or more hidden layers. The basic structure of a deep neural network is similar to the basic structure of a shallow neural network and consists of an input layer, middle hidden layers, and an output layer. However, the parameters, calculation units and algorithms of deep neural networks are more abundant than traditional shallow neural networks. As shown in Fig 1, input data (x) are given to the input layer, processed layer by layer through the hidden layer, and then transmitted to the output layer. The weights $w_{(i)}$ between neurons are free parameters that capture the model's representation of the data and are learned from input/output samples. Each neuron computes a weighted sum of its inputs and applies a nonlinear activation function to calculate its outputs. The formulation of input data in forward propagation is calculated according to Eq 1:

$$a_i^{(l+1)} = \delta(Z_i) \tag{5}$$

$$Z_i = w_i^{(l+1)} a^l + b_i^{(l+1)} \tag{6}$$

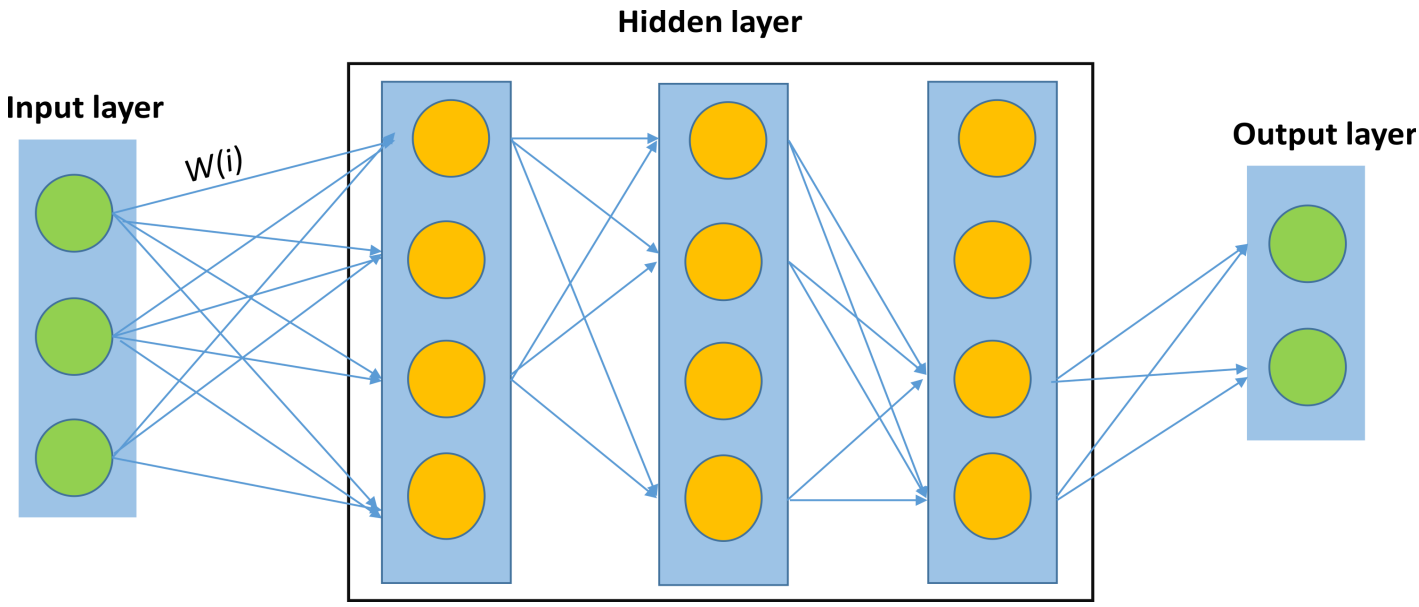

Fig 1. Neural network training procedure.

https://doi.org/10.1371/journal.pone.0217312.g001

where $a^{(l+1)}$ is the input data of the *(l+1)-th* layer, δ denotes the activation of the *(l+1)-th* layer, $w^{(l+1)}$ is the connection weight matrix between the *(l)-th* layer and the *(l+1)-th* layer, a^{l} is the input data of the *(l)-th* layer, and $b^{(l+1)}$ is the bias term in the *(l+1)-th* layer.

Back propagation is the propagation of the output through the hidden layer to the input layer, and the error is distributed to all of the cells of each layer, to obtain the error signal of each layer. In general, ReLU (rectified linear unit) is used as the activation function for neurons in DNN. The ReLU can change all negative values to zero while leaving the positive values unchanged. Compared to other activation functions, ReLU has a few advantages [26, 27]. For linear functions, ReLU is more expressive, especially in deep networks. For non-linear functions, ReLU does not have the disadvantage of gradient disappearance and can thereby maintain the convergence speed of the model at a stable level.

Evaluation measure

The performance of the models was evaluated by a series of evaluation indicators, including the accuracy, recall, AUC and loss in this study. Their criterion functions are defined, respectively, by:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

where *TP*, *TN*, *FP*, and *FN* represent the true positive, true negative, false positive, and false negative, respectively. AUC was calculated using an open source code [28]. Loss was calculated according to the following cross-entropy function:

$$loss(y, y^*) = -\frac{1}{n}\sum_{i=1}^{n}(y_i^* \ln y_i + (1 - y_i^*)\ln(1 - y_i)) \tag{9}$$

where y = (y_1,y_2,y_3. . .. . .y_n) represents the actual output and $y* = (y_1^*, y_2^*, y_3^*, \ldots\ldots, y_n^*)$ represents the desired output.

Results

Selecting optimal parameters

Selecting an optimal parameter is an important step in the model training process and one of the key elements in training a robust model. In this experiment, ReLU was selected as the activation function, Adam as the optimizer, and cross entropy as the cost function. Compared with the Sigmoid and the Tanh activation functions, ReLU has a simple operation, has the sparse expression ability and learning ability of a neural network, and has a faster convergence speed during the gradient descent. Due to the above advantages, ReLU was used as the activation function for this model [27]. Adam combines the advantages of the RMSprop and Adagrad algorithms for the improved handling of noise, which led us to choose it as the optimizer [29, 30]. The cross-entropy cost function can measure the predicted and actual values in a deep neural network, and it can compensate for the defects caused by the easy saturation of the sigmoid function, thus causing the training set to converge faster. In this experiment, we chose to use the cross-entropy cost function.

In our method, the three parameters of learning rate, network width and network depth must be determined. To determine the learning rate, the number of hidden layer nodes is set to 64, the activation function is ReLU, the optimization algorithm is Adam, the batch size is 128, the dropout is 0, and the number of iterations is 300,000. The results of adjusting the learning rate are shown in Table 2. From Table 2, we found that when the learning rate is 0.01, it has the best predictive performance in the context of PPIs prediction. Therefore, here, 0.01 was chosen as the learning rate for our experiment.

To determine the network width of the model, the learning rate is set to 0.01, the other parameters are unchanged, and the network width results of our model are shown in Table 3. According to Table 3, when the network width is 512, the performance of the model is better than that of several other network widths. Therefore, the network width of this model is set to 512.

After the learning rate and network width are determined, the next step is to determine the depth of the model. The depth adjustment results are shown in Table 4. From Table 4, the

Table 2. Adjusting the learning rate of our model.

Learning rate	Accuracy (%)	AUC (%)	Recall (%)	Loss (%)	Train time(s/100 steps)
0.01	0.7926±0.0256	0.8807±0.023	0.8923±0.0239	0.4373±0.0377	0.1296±0.001
0.001	0.7553±0.0377	0.8495±0.0367	0.8603±0.0393	0.4900±0.0489	0.1263±0.0012
0.0001	0.6784±0.0273	0.7584±0.0307	0.7616±0.0337	0.5871±0.0271	0.1307±0.0011
0.00001	0.6363±0.0076	0.6921±0.0096	0.687±0.01	0.6443±0.0193	0.1303±0.001

https://doi.org/10.1371/journal.pone.0217312.t002

Table 3. Adjusting of the network width of our model.

Width	Accuracy (%)	AUC (%)	Recall (%)	Loss (%)	Train time(s/100 steps)
128	0.7191±0.02	0.8139±0.0169	0.8232±0.0182	0.542±0.0236	0.1594±0.0034
256	0.7476±0.0199	0.8455±0.0161	0.8618±0.0182	0.4966±0.0276	0.1875±0.0014
512	0.8277±0.0352	0.9107±0.0296	0.9211±0.0288	0.3826±0.0527	0.2497±0.0026
1024	0.8234±0.0317	0.9081±0.0283	0.9192±0.0278	0.3882±0.0484	0.3483±0.0034

https://doi.org/10.1371/journal.pone.0217312.t003

Table 4. Adjusting the network width of our model.

Depth	Accuracy (%)	AUC (%)	Recall (%)	Loss (%)	Train time(s/100 steps)
512×512	0.8262±0.0376	0.913±0.0316	0.9241±0.0303	0.3904±0.0513	0.8478±0.0081
512×512×512	0.9159±0.0563	0.9621±0.0379	0.9689±0.035	0.2598±0.075	1.4374±0.0159
512×512×512×512	0.7988±0.0407	0.891±0.0335	0.9068±0.0333	0.428±0.0544	2.0333±0.0087
512×512×512×512×512	0.7104±0.0898	0.7976±0.1201	0.8447±0.0483	0.5276±0.0785	2.6208±0.0126

https://doi.org/10.1371/journal.pone.0217312.t004

model performance is best when the depth is 512 × 512 × 512. Therefore, 512 x 512 x 512 was chosen as our model depth.

Performance of MOS on PPIs

Results on benchmark dataset. The proposed DNN-MOS model (protein sequences coded by MOS descriptors) was applied to the human dataset. To investigate the contribution of the novel MOS descriptor, we separately trained DNN based on CT, AC, LD, and MOS. Among them, the parameter settings of DNN-MOS are shown in 3.1. The parameters of DNN-CT (deep neural network- conjoint triad), DNN-AC (deep neural network-auto covariance) and DNN-LD (deep neural network-local descriptor) are set as follows: the activation function was ReLU, the optimization algorithm was Adam, the batch size was 128, the dropout was 0, the number of hidden layer nodes was set to 256, the network depth was [256-256-256], the learning rate was 0.001, the number of times to repeat the hold-out-validation was 30 and the number of times was 10,000 per iteration.

The results of each prediction model are shown in Table 5. From Table 5, we can observe that the predictive performance using MOS is not superior to other descriptors for almost all evaluation metrics. The accuracy and AUC of DNN-MOS are 94.34% and 98.28%, lower than those of DNN-CT and DNN-AC. The AUC of DNN-MOS is slightly higher than that of DNN-LD and significantly lower than those of DNN-CT and DNN-AC. However, the loss of MOS is significantly better than the other three encoding methods.

The training time is related to the parameters, such as the width and depth of the model. To compare the training time of each code, we set the parameters the same. The parameters of DNN-MOS, DNN-CT, DNN-AC and DNN-LD are set as follows: the number of hidden layer nodes was set to 64, the activation function was ReLU, the optimization algorithm was Adam, the batch size was 128, the dropout was 0, the learning rate was 0.001, the number of times to repeat the hold-out-validation was 30 and the number of times was 10,000 per iteration. The results of the training time are shown in Table 6. As shown in Table 6, the DNN-MOS has the lowest training time per 1000 steps, only 0.1261 seconds. The training time of DNN-MOS is nearly 2 times faster than DNN-AC's training time, more than 2 times faster than DNN-CT's and more than 3 times faster than DNN-LD's. From Table 6, we found that the difference in test time was small, but the test time trend was the same as the training time. Therefore, we

Table 5. Results based on DNN with CT, AC, LD, and MOS on the benchmark dataset.

Method	Accuracy	Recall	AUC	Loss
DNN-CT	0.9711±0.0038	0.9891±0.0009	0.9835±0.0018	0.2747±0.0686
DNN-AC	0.9684±0.0013	0.9867±0.0013	0.9802±0.0022	0.6591±0.3178
DNN-LD	0.953±0.0087	0.9828±0.003	0.9757±0.0043	0.3623±0.0924
DNN-MOS	0.9434±0.0078	0.9828±0.0023	0.9779±0.0028	0.2325±0.0154

https://doi.org/10.1371/journal.pone.0217312.t005

Table 6. Results based on DNN with CT, AC, LD, and MOS on the benchmark dataset.

Method	Train time (s)	Test time (s)	The dimensions of vector space	Data set
DNN-CT	0.2852±0.0039	1.39E-05	686	HPRD (36591) + Swiss-Port (36324)
DNN-AC	0.2186±0.0014	1.32E-05	420	HPRD (36591) + Swiss-Port (36324)
DNN-LD	0.4045±0.0141	1.48E-05	1260	HPRD (36591) + Swiss-Port (36324)
DNN-MOS	0.1261±0.0039	1.28E-05	58	HPRD (36591) + Swiss-Port (36324)

https://doi.org/10.1371/journal.pone.0217312.t006

found that MOS can significantly save training time and test time. From Table 6, we can also see that the larger the vector dimension, the more training time was required.

Results on non-redundant dataset. To further assess the practical prediction ability of DNN-MOS, we trained the models of DNN-MOS, DNN-CT and DNN-AC on a non-redundant dataset (removing the samples that has ≥25% sequence identity to any sample in the pre-training set). The prediction results are shown in Table 7. From Table 7, we can observe that the accuracy of DNN-MOS, DNN-CT and DNN-AC on the non-redundant dataset are 88.29%, 89.88% and 93.35%, respectively. Shen et al. [17] studied the PPIs of the dataset using a deep learning algorithm, achieving an accuracy of 85.84%, which is lower than our results.

Comparison with different classifiers

In order to verify the effectiveness of the feature extraction method of MOS on PPIs, we combined the MOS with Decision Tree (DT), K-Neighbors (KN) and Random Forest (RF) on human data to construct three models of DT-MOS (decision tree—matrix of sequence), KN-MOS (K-Neighbors—matrix of sequence) and RF-MOS (random forest—matrix of sequence). The results are shown in Table 8. From Table 8, we can see that these methods present an accuracy of 83.01–97.29%, and the accuracies of DT-MOS, KN-MOS and RF-MOS are 94.36%, 83.01%, and 97.29%, respectively. These results show that the novel MOS of our proposed are also effective in other classifiers such as DT, KN and RF.

Discussion

We have presented a novel protein sequence coding approach for PPIs prediction. Of note, we propose a strategy for projecting protein sequences into a vector space, which is used to represent the matrix space of PPI information. Specifically, we first classify 20 amino acids into 7 amino acids according to their physicochemical properties (Table 1). The dimensions of the

Table 7. Results of DNN with different feature extraction method on a non-redundant dataset.

Methods	Accuracy	Recall	AUC
DNN-MOS	88.29%	93.63%	92.23%
DNN-CT	89.88%	93.79%	91.78%
DNN-AC	93.35%	96.24%	94.99%
Shen's work [17]	85.84%	N/A	N/A

https://doi.org/10.1371/journal.pone.0217312.t007

Table 8. Comparison of the performances of MOS based on different classifiers using the human dataset.

Methods	Accuracy	Recall	AUC
DT-MOS	0.9436	0.9365	0.9436
KN-MOS	0.8301	0.6973	0.8298
RF-MOS	0.9729	0.9611	0.9729

https://doi.org/10.1371/journal.pone.0217312.t008

matrix space can be significantly reduced, from 20×20 to 7×7. Next, we combine the elements on the 7×7 matrix diagonal and the elements above the diagonal into a 28-dimensional vector. To distinguish the length of a sequence, a sequence label is added. Finally, a 29-dimensional vector can represent a protein sequence. We combined MOS with DT, KN and RF and achieved good results. The experimental results show that the proposed MOS feature extraction method is effective. However, the disadvantage of the novel matrix sequence descriptor is that the sequence matrix cannot be in one-to-one correspondence with the protein sequence. For any given two sequences, the corresponding sequence matrices are different when the sequence lengths are different, or the sequence lengths are the same but at least one element contains different numbers of elements. Therefore, pre-processing data is required to remove protein pairs with the same protein sequence length and the same number of elements.

Recently, new feature extraction approaches for PPIs have been developed [30–33]. Among them, Li et al. [30] proposed a new method for predicting self-interacting proteins (SIPs) based on amino acid sequences, achieving high precisions of 86.86 and 91.30% on the *Saccharomyces cerevisiae* and human SIPs datasets, respectively. Wang et al. [31] reported a novel method of PPIs based on pseudo position specific scoring matrix (PSSM) feature descriptors and an ensemble rotation forest (RF) learning system from protein amino acid sequences. Their method achieved accuracies of 98.38%, 89.75%, and 96.25% on the yeast, *H. pylori*, and independent datasets, respectively. Li et al. [32] developed a new hybrid method of physical chemistry and evolution-based feature extraction methods, which can capture discriminant features from evolution-based information and physicochemical features. An et al. [33] explored a new feature representation method based on local binary pattern (LBP), which not only considers the amino acid sequence information but also the evolutionary information of multiple sequence alignments. The above studies show that effective feature extraction methods can mine useful information on protein pairs and improve the performance of PPIs prediction. In this study, although we found that the performance of DNN-MOS is not prominent in Table 5, DNN-MOS can greatly reduce loss and training time (Table 6). In addition, Table 8 show that the novel MOS of our proposed are also effective in other classifiers such as DT, KN and RF. Overall, although the performance of DNN-MOS is not prominent, it can be a useful supplement to PPIs predictions. The reason why the accuracy of DNN-MOS is lower than that of DNN-CT, DNN-AC and DNN-LD may be due to the loss of part of the information when converting the protein sequence into a matrix vector. In future research, we will try our best to solve this problem and improve the predictive performance of DNN-MOS.

Conclusion

With the increasing number of PPI calculation methods, the coding methods of various amino acid feature vectors are also emerging. Although the various protein encoding methods such as AC, CT, and LD are useful, one of the disadvantages is that the order relationship of the entire amino acid sequence is not considered. The CT [19] considers considered the order relationship of three amino acids. AC [20] considers the order relationship of 30 amino acids. LD only considers the neighbouring effect of two adjacent types of amino acids [21]. To overcome this problem, we propose an efficient method for predicting PPIs from amino acid sequences by a novel matrix sequence descriptor feature representation with deep neural network. The novel protein feature extraction method we have proposed considers the order relationship of the entire amino acid sequence. When performed on human PPIs data, DNN-MOS, DT-MOS, KN-MOS and RF-MOS have achieved good results. Additionally, the model was used to evaluate this prediction model on a non-redundant dataset and the prediction accuracy is 88.29%. The experimental results show that the matrix sequence descriptor is

promising for predicting PPIs and can be used as a complementary supplement to other methods.

Supporting information

S1 File. The positive protein-protein interaction. There are 36,630 protein-protein pairs from total 9476 proteins, and the first column is protein ID from HPRD, the second column is the other protein ID and the two proteins constitute the positive Protein-protein interaction.
(DOC)

S2 File. The negative protein-protein interaction. There are 36,480 protein-protein pairs from total 2184 proteins, and the first column is protein ID, the second column is the other protein ID and the two proteins constitute the negative Protein-protein interaction.
(DOC)

S3 File. The identity of positive protein-protein interaction is below 25%. There are 3899 protein-protein pairs from total 2502 proteins, and the first column is protein ID from HPRD, the second column is the other protein ID and the two proteins constitute the positive Protein-protein interaction and protein identity of all the proteins from S3 file is below 25%.
(DOC)

S4 File. The identity of negative protein-protein interaction is below 25%. There are 4262 protein-protein pairs from total 661 proteins, and the first column is protein ID from HPRD, the second column is the other protein ID and the two proteins constitute the positive Protein-protein interaction and protein identity of all the proteins from S4 file is below 25%.
(DOC)

Acknowledgments

We thank YJ Wu and RJ Wang for designing this study, and YM Gui, YY Wei, and other members of our laboratory for providing language help, building the model and writing assistance.

Author Contributions

Data curation: Xue Wang, Yuanyuan Wei.

Formal analysis: Yuanmiao Gui.

Funding acquisition: Rujing Wang.

Investigation: Yuejin Wu, Yuanmiao Gui.

Methodology: Xue Wang, Yuanmiao Gui.

Project administration: Yuejin Wu, Rujing Wang.

Resources: Yuanyuan Wei, Yuanmiao Gui.

Software: Yuanmiao Gui.

Supervision: Yuejin Wu, Yuanyuan Wei.

Visualization: Rujing Wang.

Writing – original draft: Xue Wang.

Writing – review & editing: Xue Wang.

References

1. Fields S, Song O. A novel genetic system to detect protein protein interactions. Nature. 1989; 340 (6230):245–246. https://doi.org/10.1038/340245a0 PMID: 2547163
2. Tong AHY, Becky D, Giuliano N, Bader GD, Brannetti B, Castagnoli L, et al. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. Science. 2002; 295(5553):321–324. https://doi.org/10.1126/science.1064987 PMID: 11743162
3. Zhu H, Bilgin M, Bangham R, Hall D, Casamayor A, Bertone P, et al. Global analysis of protein activities using proteome chips. Science 2001; 293(5537):2101–2105. https://doi.org/10.1126/science.1062191 PMID: 11474067
4. Pellegrini M, Marcotte EM, Thompson MJ, Eisenberg D, Yeates TO. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. Proc. Natl. Acad. Sci.1999; 96(8):4285–4288. https://doi.org/10.1073/pnas.96.8.4285 PMID: 10200254
5. Zhang SW, Hao LY, Zhang TH. Prediction of protein-protein interaction with pairwise kernel support vector machine. International journal of molecular sciences. 2014; 15(2):3220–3233. https://doi.org/10.3390/ijms15023220 PMID: 24566145
6. Marcotte EM. Detecting protein function and protein-protein interactions from genome sequences. Science 1999; 285(5428): 751–753. PMID: 10427000
7. Enright AJ, Iliopoulos I, Kyrpides NC, Ouzounis CA. Protein interaction maps for complete genomes based on gene fusion events. Nature. 1999; 402(6757): 86–90. https://doi.org/10.1038/47056 PMID: 10573422
8. Rashid M, Ramasamy S, Raghava GP, A simple approach for predicting protein-protein interactions, Curr. Protein Pept. Sci., 2010; 11(7): 589–600. PMID: 20887258
9. Dohkan S, Koike A, Takagi T, Improving the performance of an SVM-based method for predicting protein-protein interactions, Silico Biol., 2006; 6(6): 515–529. PMID: 17518762
10. Chatterjee P, Basu S, Kundu M, et al., PPI_SVM: Prediction of protein-protein interactions using machine learning, domain-domain affinities and frequency tables, Cell Mol. Biol. Lett., 2011; 16(2): 264–278. https://doi.org/10.2478/s11658-011-0008-x PMID: 21442443
11. Lin X, Chen XW. Heterogeneous data integration by tree augmented naive bayes for protein-protein interactions prediction proteomics. Proteomics. 2013; 13(2):261–268. https://doi.org/10.1002/pmic.201200326 PMID: 23112070
12. Najafabadi HS, Salavati R, Sequence-based prediction of protein-protein interactions by means of codon usage, Genome Biol., 2008; 9(5): R87. https://doi.org/10.1186/gb-2008-9-5-r87 PMID: 18501006
13. Valente GT, Acencio ML, Martins C, et al., The development of a universal in silico predictor of protein-protein interactions, Plos One,2013; 8(5): e65587. https://doi.org/10.1371/journal.pone.0065587 PMID: 23741499
14. Chen XW, Liu M, Prediction of protein-protein interactions using random decision forest framework, Bioinformatics, 21(24): 4394–4400, 2005. https://doi.org/10.1093/bioinformatics/bti721 PMID: 16234318
15. Saha I, Zubek J, Klingström T, et al., Ensemble learning prediction of protein-protein interactions using proteins functional annotations, Molecular Biosystems, 10(4): 820–830, 2014. https://doi.org/10.1039/c3mb70486f PMID: 24469380
16. Qi Y, Klein-Seetharaman J, Bar-Joseph Z, Random forest similarity for protein-protein interaction prediction from multiple sources, Pac. Symp. Biocomput, 2015; 10: 531–542. PMID: 15759657
17. Sun TL, Zhou B, Lai HH, Pei J. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. BMC Bioinformatics. 2017; 18 (1):277–285. https://doi.org/10.1186/s12859-017-1700-2 PMID: 28545462
18. Wang J, Zhang L, Jia L, Ren Y, Yu G. Protein-protein interactions prediction using a novel local conjoint triad descriptor of amino acid sequences. International Journal of Molecular Sciences. 2017; 18 (2373): 1–17. https://doi.org/10.3390/ijms18112373 PMID: 29117139
19. Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, et al. Predicting protein-protein interactions based only on sequences information. Proc. Natl Acad. Sci. 2007; 104 (11): 4337–4341. https://doi.org/10.1073/pnas.0607879104 PMID: 17360525
20. Guo YZ, Yu LZ, Wen ZN, et al. Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. Nucleic Acids Research 2008; 36 (9): 3025–3030. https://doi.org/10.1093/nar/gkn159 PMID: 18390576
21. Yang L, Xia JF, Gui J. Prediction of protein-protein interactions from protein sequence using local descriptors. Protein Pept. Lett. 2010; 17(9):1085–1090. PMID: 20509850

22. Pan XY, Zhang YN, Shen HB. Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features. Journal of proteome research. 2010; 9(10):4992–5001. https://doi.org/10.1021/pr100618t PMID: 20698572

23. Von Mering C, Krause R, Snel B, Cornell M, Olivier SG, Fields S, et al. Comparative assessment of large-scale data sets of protein-protein interactions. Nature. 2002; 417(6887):399–403. https://doi.org/10.1038/nature750 PMID: 12000970

24. Sprinzak E, Sattath S, Margalit H. How reliable are experimental protein-protein interaction data? Journal of molecular biology. 2003; 327(5):919–923. PMID: 12662919

25. Li ZW, You ZH, Chen X, Gui J, Nie R. Highly accurate prediction of protein-protein interactions via incorporating evolutionary information and physicochemical characteristics. International journal of molecular sciences. 2016; 17(9):1396–1408. https://doi.org/10.3390/ijms17091396 PMID: 27571061

26. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural computation. 2006; 18(7):1527–54. https://doi.org/10.1162/neco.2006.18.7.1527 PMID: 16764513

27. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. International conference on artificial intelligence & statistics, 2011; 15: 315–323.

28. Van LT, Nabuurs SB, Marchiori E. Gaussian interaction profile kernels for predicting drug-target interaction. Bioinformatics. 2011; 27 (21): 3036–3043. https://doi.org/10.1093/bioinformatics/btr500 PMID: 21893517

29. Kang G, Li J, Tao D. Shakeout: A new approach to regularized deep neural network training. IEEE trans pattern anal mach intell. 2018; 40(5):1245–1258. https://doi.org/10.1109/TPAMI.2017.2701831 PMID: 28489533

30. Li JQ, You ZH, Li X, Zhong Z, Chen X. PSPEL: In Silico prediction of self-interacting proteins from amino acids sequences using ensemble learning. IEEE/ACM transactions on computational biology and bioinformatics. 2017; 14 (5):1165–1172. https://doi.org/10.1109/TCBB.2017.2649529 PMID: 28092572

31. Wang L, You ZH, Chen X, Li JQ, Yan X, Zhang W, et al. An ensemble approach for large-scale identification of protein-protein interactions using the alignments of multiple sequences. Oncotarget. 2017; 8 (3): 5149–5159. https://doi.org/10.18632/oncotarget.14103 PMID: 28029645

32. Li ZW, You ZH, Chen X, Gui J, Nie R. Highly accurate prediction of protein-protein interactions via incorporating evolutionary information and physicochemical characteristics. International journal of molecular sciences. 2016; 17(9):1396–1408. https://doi.org/10.3390/ijms17091396 PMID: 27571061

33. An JY, You ZH, Chen X, Huang DS, Yan GY, Wang DF. Robust and accurate prediction of protein self-interactions from amino acids sequence using evolutionary information. Molecular biosystems. 2016; 12 (12):3702–3710. https://doi.org/10.1039/c6mb00599c PMID: 27759121.