Application of K-nearest neighbors in protein-protein interaction prediction

Yuanmiao Gui, Xue Wang*

Institute of Intelligent Machine, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei City, Anhui Province, 230031, China

*Corresponding author e-mail: xwang@iim.ac.cn

Abstract. Protein-protein interactions (PPIs) are an important part of many life processes in organisms. Almost all life processes are related to protein-protein interactions, and the study of protein interactions plays an important role in revealing the mysteries of life activities. In order to improve the prediction performance of protein-protein interaction, we are based on K-Nearest Neighbor (KNN), combined with protein sequence coding methods such as Conjoint Triad (CT), Auto Covariance (AC) and Local Descriptor (LD) to construct KNN-CT, KNN-AC and KNN-LD three prediction models of PPIs. The results show that the prediction models KNN-CT and KNN-AC have obtained accuracy rates of 94.29% and 94.69%, respectively, which are better than existing methods. The results show that K-nearest neighbors can be a useful complement to protein-protein interactions.

Keywords: K-Nearest Neighbor; Conjoint Triad; Auto Covariance; Local Descriptor; Protein-protein interaction.

1. Introduction

Protein-protein interaction (PPIs) is a process in which two or more protein molecules form a protein complex through non-covalent bonds. Almost all life processes are related to protein interactions, such as metabolism, signal transduction, cell cycle regulation, metabolism, cell apoptosis, immune response, and so on. With the development of high-origin experimental technology (such as gene chip, experimental technology, etc.), a large number of biological sequences have appeared [1-3]. There are many unknown biological functions hidden in these data. How to mine the information implicit in these data is an urgent problem for researchers to solve. As a new round of calculation method, machine learning provides very favorable conditions for speeding up data mining. In recent years, machine learning such as support vector machines [4-6], neural networks [7], decision trees [8-9] and random forests [10-11] have played an important role in various fields. It has been widely used in forecasting. For example, Guo et al. [12] proposed a method of protein interaction prediction based on support vector machine (SVM), and obtained a accuracy rate of 86.55% on the Saccharomyces cerevisiae data set. Yang et al. [13] used local descriptors (LD) for the first time to encode protein sequences into space vectors of the same dimension, and achieved an accuracy of 86.15% on the distilled yeast data set. Although machine learning algorithms such as support vector machines and deep learning are widely used in protein interaction prediction, KNN are rarely used in protein interaction prediction.

KNN is a distance metric classification algorithm proposed by Cover and Hart [14]. It is a widely used algorithm in machine learning. KNN can not only be used for classification problems, it can also be used to deal with regression problems. Due to its intuitive, simple, effective, and easy-to-implement characteristics, it is widely used in many classification problem areas such as image detection and target tracking [15-17]. In order to study the performance of protein interaction prediction based on KNN, we combined three protein coding methods of Conjoint Triad (CT), Auto Covariance (AC) and Local Descriptor (LD) to construct KNN-CT and KNN-AC and KNN-LD three PPIs prediction models, after testing, the three models have obtained accuracy rates of 94.29%, 94.67% and 93.45% respectively. The results show that the PPIs prediction based on KNN is feasible and can be used as a useful supplement for future PPIs prediction research.

2. Materials and Methods

2.1 Protein sequence coding method

In this experiment, three protein sequence coding methods including CT, AC, and LD were used to transform protein sequences into feature vectors of specific dimensions.

CT was first proposed by Shen [18]. The method first divides 20 conventional amino acids into 7 groups according to the dipolarity and volume of the side chains, and replaces each amino acid with its category according to the classification; then, calculates the frequency of each category, and converts the protein sequence into a space vector. The specific process of generating vector space is as follows: Use a binary space vector (V, F) to represent a protein sequence, where V represents a sequence feature, and each feature (vi) represents a type of triplet; F is the frequency vector corresponding to V, and F (fi) is the protein frequency of sequence vi. Amino acids are classified into 7 categories, and V is 777; therefore, i = 1,2...,343. If a set of data i1i2i3, then the corresponding value of F(fi) is $i_1 + (i_2 - 1) \times 7 + (i_3 - 1) \times 7 \times 7$. The specific formula is as follows:

$$\begin{Bmatrix} 111 = \text{f1} \;\; 121 = \text{f8} \ldots 177 = \text{f337} \\ 211 = \text{f2} \;\; 221 = \text{f9} \ldots 277 = \text{f338} \\ \ldots \\ 711 = \text{f7} \;\; 721 = \text{f14} \ldots 777 = \text{f343} \end{Bmatrix} \tag{1}$$

After normalization, a protein sequence forms a 343-dimensional vector space, and the interaction between two proteins is represented by a 686-dimensional vector space.

AC is a commonly used protein coding method, which considers the adjacent effects between a certain number of amino acids in the protein sequence, and has been widely used in protein coding [12,19]. The interaction between amino acids is reflected by the seven physicochemical properties of the sequence-based amino acids. After the original values are normalized, the covariance variables are calculated using formula (2).

$$AC_{lag,j} = \frac{1}{n - lag} \sum_{i=1}^{n-lag} (X_{i,j} - \frac{1}{n} \sum_{i=1}^{n} X_{i,j})(X_{(i+lag),j} - \frac{1}{n} \sum_{i=1}^{n} X_{i,j}) \tag{2}$$

Where AClag,j is the AC variable, lag is the distance between residues, j represents one descriptor, i is the position in the sequence X, n is the length of the sequence X. According to the study of Guo et al. [12], the lag value is determined to be 30, and j is the seven physical and chemical properties of amino acids. Therefore, the vector space of AC is 210 (30×7) dimensions. After transforming each protein sequence into the AC variable vector, the two protein vectors are spliced, that is, the 420-dimensional vector space represents a protein pair.

LD [20] is a method that does not require sequence alignment, and its effectiveness depends to a large extent on the classification of amino acids. See reference [21] for specific calculation methods.

2.2 K-Nearest Neighbor (KNN)

2.2.1 Distance measurement algorithm

The distance between two instance points in the KNN feature space is presented by the similarity of the two instance points, so the distance measurement of KNN often uses similarity measurement methods such as Euclidean distance and Manhattan distance to calculate the distance between instances. The KNN algorithm is based on the following assumption that most of the K nearest sample categories of all samples in the feature space determine the category of the sample itself, and have the characteristics of the K nearest sample categories. When determining the classification decision of the sample to be tested, the KNN algorithm only determines the category of the sample to be tested based on the category of one or more recent training samples. The KNN algorithm assumes that all samples correspond to points in the n-dimensional space, and the nearest neighbor of each sample is

defined according to the standard Euclidean distance. In the n-dimensional space, any sample x can be represented by the following feature vector:

$$\langle a_1(x), a_2(x), \ldots, a_n(x) \rangle \tag{3}$$

Among them, ar(x) represents the r-th attribute value of x. Therefore, the distance between xi and xj is as shown in Equation (4):

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \tag{4}$$

2.2.2 Implementation steps of KNN algorithm

The three models of KNN-CT, KNN-AC and KNN-LD were implemented on the Sklearn platform and python environment. The implementation steps of the KNN are as follows:

Table 1. Division of amino acids based on the dipoles and volumes of the side chains.

Algorithm KNN

Input: A[N] is the classification feature of N training samples; K is the number of neighbors.

Output: *The category of x.*

1 Choose A[1] to A[K] as the initial neighbors of *x*;

2 Calculate the Euclidean distance *d(x, A[i])* between the initial nearest neighbor and the test sample *x*, *i=1,2,3,...,k*;

3 Sort from small to large according to d(x, A[i]);

4 Calculate the distance D between the farthest sample and x, that is, max d(x, A[i]), i=1,2,3,…,k;

5 i=K+1;

6 ***while*** *i<n+1* ***do****;*

7 | *Calculate the distance between A[i] and x d(x, A[i]);*

8 | if *d(x, A[i])<D;*

9 | | *Replace the farthest sample with A[i]*;

10 | | *Sort from small to large according to d(x, A[i]);*

11 | | *Calculate the Euclidean distance D between the farthest sample and x, that is, d(x, A[i]), i=1,2,3,...,k;*

12 | | *Calculate the probability of the category of the first k samples A[i], i=1,2,3,...,k;*

13 | | *The category with the greatest probability is the category of sample x;*

14 | *end*

15 end

2.3 Evaluation of predictive performance

In this experiment, four indicators: Accuracy, Recall, Loss, and Area Under the Receiver Operating Characteristic Curve (AUC) are used to evaluate prediction performance. The calculation formulas for accuracy and recall are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$\mathrm{Re}\,call = \frac{TP}{TP + FN} \tag{6}$$

Among them, TP, TN, FP and FN represent true positive, true negative, false positive and false negative respectively. AUC is calculated by open source code [23]. The loss is calculated by the cross-entropy function, the formula is as follows:

$$loss(y, y^*) = -\frac{1}{n}\sum_{i=1}^{n}(y_i^* \ln y_i + (1 - y_i^*)\ln(1 - y_i)) \tag{7}$$

where, y=(y1,y2,y3……yn), is the actual output; y*=$(y_1^*, y_2^*, y_3^*, \ldots\ldots, y_n^*)$ is the expected output.

3. Experiment and Result Analysis

3.1 Construction of the sample set

This experiment mainly uses human protein data, which comes from http://www.csbio.sjtu.edu.cn/bioinf/LR_PPI/Data.htm. Among them, the positive samples were taken from the Human Protein Reference Database (HPRD, 2007 edition), and the negative samples were constructed with subcellular location information. Most protein sequences range from 100 to 1000 in length, and we removed protein pairs containing less than 50 residues and unusual amino acid sequences, such as B, J, O, U, X, and Z. The resulting data set contains 36,591 pairs of positive samples and 36,324 pairs of negative samples, of which 30,000 positive samples and 30,000 negative samples are randomly selected each time to form a training data set, and the rest is used as a test set to validate the model.

3.2 Hyperparameter

Table 2. Recommended parameter of KNN models.

Parameter	Range	Value
N_Neighbors	1-10	5, 9
Weights	Uniform, Distance	Uniform
Algorithm	Auto, Ball_Tree, KD_Tree, Brute	Ball_Tree
Leaf_size	10, 20, 30, 40	30
Metric	Minkowski	Minkowski
P	1, 2	2
N_Jobs	1	1

Hyparameter adjustment is a very important step in model construction. Only by selecting the best parameters can an optimal model be constructed. After a lot of experiments and debugging, the parameters used in experiment were summarized, as shown in Table 2. K defaults to 5, here the k value was set to 1-10 to test separately. When the K value is 9, KNN-CT and KNN-AC have the best prediction performance; when the K value is 5, KNN-LD has the best prediction performance. Weights is used to identify the weight of each sample's neighbor samples. The default value of KNN in Sklearn is uniform, which assigns a uniform weight to each neighbor. There are four KNN algorithms in Sklearn, such as Brute, KD_Tree, Ball_Tree and Auto. Brute is mostly used for input samples with sparse features, KD_Tree is mostly used for datasets with large number of samples or features, and Ball_Tree is mostly used for datasets with large number of samples or features and uneven samples. After testing, the Ball_Tree algorithm is selected in experiment. Leaf_size is used to control the leaf node threshold of KD tree or ball tree to stop building subtrees. The smaller the threshold of the leaf node, the larger the generated KD tree or ball tree; The deeper the number of leaf nodes, the longer the establishment time. On the contrary, the shallower the number of establishment layers and the shorter the establishment time. After adjustments in experiment, Leaf_size is determined to be 30. Metric were used to measure distance, the default is "Minkowski". P is the

auxiliary parameter of the distance measurement parameter Metric. It is only used for the selection of P value in "Minkowski", and the default is 2. N_jobs is the number of parallel processing tasks, and default value is 1.

3.3 Predictive performance of PPIs

We combined KNN with CT, AC, and LD feature extraction methods to construct three PPIs prediction models: KNN-CT, KNN-AC, and KNN-LD. The average prediction performance of the three models of KNN-CT, KNN-AC and KNN-LD was shown in Table 3. It can be seen from Table 3 that the accuracy range of the three models of KNN-CT, KNN-AC and KNN-LD is 93.45-94.67%, and the range of change is 1.22%. The accuracy rates of the three models of KNN-CT, KNN-AC and KNN-LD are 94.29%, 94.67% and 93.45%, respectively. Among them, the accuracy, AUC, recall and precision of the model KNN-AC are slightly higher than those of the model KNN-CT, and are significantly higher than those of the KNN-LD.

Table 3. The average prediction performance of the different prediction models.

Model	Accuracy(%)	AUC(%)	Recall (%)	Precision (%)
KNN-CT	94.29	94.29	91.18	93.07
KNN-AC	94.67	94.68	92.19	93.42
KNN-LD	93.45	93.45	89.66	92.14

Figure 1 shows the accuracy of the three models of KNN-CT, KNN-AC and KNN-LD with different changes in Neighbors. It can be seen from the figure that the accuracy of KNN-AC and KNN-CT is significantly better than that of KNN-LD, and the accuracy of KNN-AC is slightly higher than that of KNN-LD. It can also be seen from the trend graph that the accuracy rate of Neighbors when odd numbers is higher than that when adjacent even numbers is close, which also confirms why the Neighbors chooses odd numbers more.

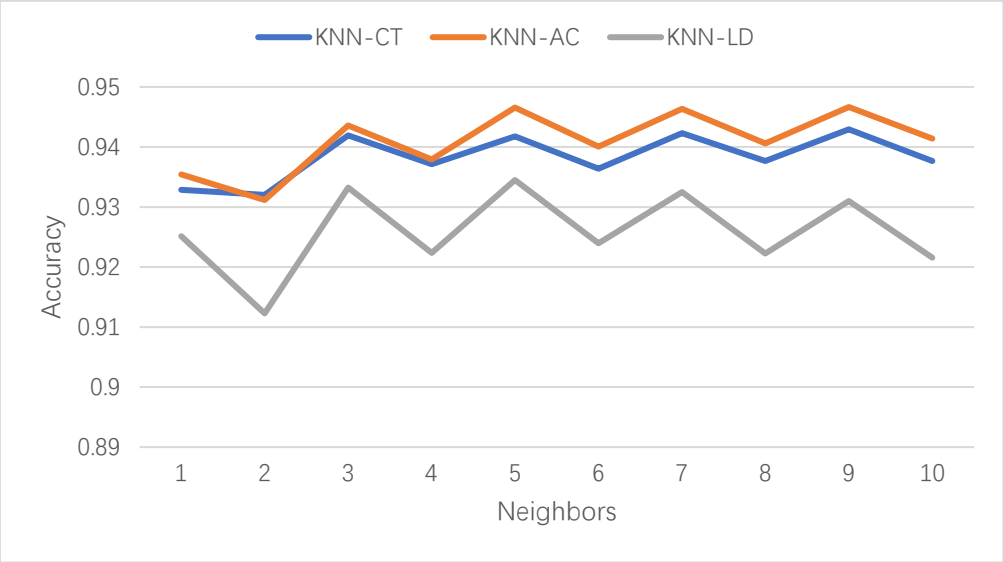

Figure 1. The accuracy values of different neighbors.

The prediction performance of KNN-AC is better in the three models of KNN-CT, KNN-AC and KNN-LD. Due to the same algorithm, the difference in the prediction performance of the three models mainly depends on the protein coding method. AC encoding can explain the interaction between amino acids and a certain number of amino acids in the sequence by selecting physical and chemical properties. This method takes into account the physical and chemical properties of the protein and retains more complete protein characteristic information [12]. In order to better capture the PPIs information from the amino acid fragments of the protein, LD divides a protein sequence into 10 local regions. In this way, the local information is not prominently highlighted, resulting in the loss of some key information, resulting in poor prediction performance [13].

3.4 Comparison with Existing Methods

In recent years, many researchers have proposed different PPIs prediction methods. In order to evaluate the prediction performance of KNN-CT, KNN-AC and KNN-LD, different methods were compared (The data sets are all human data sets), and the comparison results were shown in Table 4. It can be seen from Table 4 that the accuracy of these methods is between 83.90% and 94.69%. Among them, the KNN-CT and KNN-AC models are significantly better than SVM-CT, ELM, SVM-LD, SVM-AC and DNN-APAAC, are slightly better than the accuracy of SVM-CS. But the accuracy of KNN-LD is 0.75% lower than SVM-CS. Although the accuracy of KNN-LD is not as good as SVM-CS, KNN-LD is simple to implement, flexible to use, and requires fewer parameters to be adjusted. KNN achieved good prediction performance in PPIs prediction, and can be used as a beneficial supplement to PPIs prediction.

Table 4. Performance comparison of different methods on the human dataset.

References	Method	Accuracy
Shen's work[23]	SVM-CT	0.8390
You's work[24]	ELM	0.8480
Zhou's work[25]	SVM-LD	0.8876
Guo's work[26]	SVM-AC	0.9067
Du's work[27]	DNN-APAAC	0.8900
Zhang's work [28]	SVM-CS	0.9410
Our method	KNN-CT	0.9429
Our method	KNN-AC	0.9469
Our method	KNN-LD	0.9345

SVM: The shorthand of Support Vector Machine; ELM: The shorthand of Extreme Learning Machine; CT: The shorthand of conjoint triad method; AC: The shorthand of auto covariance; DNN: The shorthand of deep neural networks; CS: The shorthand of compressed sampling; APAAC: Amphiphilic pseudoamino acid composition.

4. Conclusion

The KNN algorithm has been applied in many fields, but it has not been found in the study of protein-protein interaction prediction. Therefore, this study is based on KNN, combined with CT, AC and LD to construct three models of KNN-CT, KNN-AC and KNN-LD, and obtain accuracy rates of 94.29%, 94.67% and 93.45%, respectively. The comparison found that the accuracy of the three models of KNN-CT, KNN-AC and KNN-LD is better than the results of SVM-CT, ELM, SVM-LD, SVM-AC and DNN-APAAC. For KNN, CT coding and AC coding obtain better prediction performance, while LD coding has a slightly worse prediction performance. The reason may be that when LD divides a protein sequence into 10 local regions, some key information is lost. This defect of LD can be used to reduce the loss of characteristic information by adding local regions in future research.

References

[1] UETZ P, Giot L, CAGNEY G, MANSFIELD T A, et al. A Comprehensive Analysis of Protein-protein Interactions in Saccharomyces Cerevisiae. Nature, 2000, 403(6770):623-627.

[2] LA COUNT DJ, VIGNALI M, CHETTIER R, et al. A Protein Interaction Network of the Malaria Parasite Plasmodium Falciparum. Nature, 2005, 438(7064):103-107.

[3] PARRISH J R, Yu J, LIU G, et al. A Proteome-wide Protein Interaction Map for Campylobacter Jejuni. Genome Biol., 2007, 8(7): R130.

[4] CHATTERJEE P, BASU S, KUNDU M, et al. Prediction of Protein-Protein Interactions Using Machine Learning, Domain-Domain Affinities and Frequency Tables. Cell Mol. Biol. Lett., 2011, 16: 264-278.

[5] RASHID M, RAMASAMY S, RAGHAVA G P, et al. A Simple Approach for Predicting Protein-Protein Interactions. Curr. Protein Pept. Sci., 2010, 11: 589-600.

[6] DOHKAN S, KOIKE A, TAKAGI T, et al. Improving the Performance of an SVM-Based Method for Predicting Protein-Protein Interactions. Silico Biol., 2006, 6: 515-529.

[7] FARISELLI P, PAZOS F, VALENCIA A, CASADIO R, et al. Prediction of Protein-Protein Interaction Sites in Heterocomplexes with Neural Networks.Eur. J. Biochem., 2002, 269: 1356-1361.

[8] VALENTE G T, ACENCIO M L, MARTINS C, et al. The Development of a Universal in Silico Predictor of Protein-Protein Interactions. PLoS One, 2013, 8(5): e65587.

[9] CHEN X W, LIU M. Prediction of Protein-Protein Interactions Using Random Decision Forest Framework. Bioinformatics, 2005, 21(24): 4394-4400.

[10] SAHA I, ZUBEK J, KLINGSTRÖM T, et al. Ensemble Learning Prediction of Protein-Protein Interactions Using Proteins Functional Annotations. Molecular Biosystems, 2014, 10(4): 820-830.

[11] 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.

[12] GUO Y, YU L, WEN Z, 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.

[13] YANG L, XIA J F, GUI J. Prediction of protein-protein interactions from protein sequence using local de scriptors. Protein and Peptide Letters, 2010, 17(9): 1085–1090.

[14] COVER T M, HART P E, et al. Nearest neighbor pattern classification. IEEE transactions on information theory, 1967, 13(1): 21–27.

[15] Liu Z G, Pan Q, Dezert J. A New Belief-Based K-Nearest Neighbor Classification Method. Pattern Recognition, 2013, 48(3): 834-844.

[16] Su M C, Chou C H. A Modified Version of the k-Means Algorithm with Distance Based on Cluster Symmetry[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(6): 674-680.

[17] Tian J, Li M Q, Chen F Z, et al. Coevolutionary Learning of Neural Network Ensemble for Complex

[18] Classification Tasks. Pattern Recognition, 2012, 45(4): 1373-1385.

[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.

[20] SUN T L, ZHOU B, LAI H H, et al. Sequence-Based Prediction of Protein Protein Interaction Using a Deep-Learning Algorithm. Bmc Bioinformatics, 2017, 18(1): 277-285.

[21] DAVIES M N, SECKER A, FREITAS A A, et al. Optimizing Amino Acid Groupings for GPCR Classification. Bioinformatics, 2008, 24(18):1980-1986.

[22] TONG J C, TAMMI M T. Prediction of Protein Allergenicity Using Local Description of Amino Acid Sequence. Front. Biosci., 2008, 13(16): 6072-6078.

[23] Van LT, Nabuurs SB, Marchiori E. Gaussian interaction profile kernels for predicting drug-target interaction. Bioinformatics. 2011; 27 (21): 3036-3043.

[24] SHEN J M, ZHANG J, LUO X M, et al. Predicting Protein-Protein Interactions Based Only on Sequences Information. Proc. Natl Acad. Sci., 2007, 104 (11): 4337-4341.

[25] YOU Z H, LI S, GAO X, LUO X, et al. Large-scale Protein-Protein Interactions Detection by Integrating Big Biosensing Data with Computational Model. Biomed Res Int., 2014, 2014(2):598129.

[26] ZHOU Y Z, GAO Y, ZHENG Y Y. Prediction of Protein-Protein Interactions Using Local Description of Amino Acid Sequence. Adv. Comput. Sci. Edu. Appl., 2011, 202: 254-262.

[27] GUO Y Z, LI M L, PU X M, et al. PRED_PPI: A Server for Predicting Protein-Protein Interactions Based on Sequence Data with Probability Assignment. Bmc Research Notes, 2010, 3(1): 145-152.

[28] DU X Q, SUN S W, HU C L, et al. DeepPPI: Boosting Prediction of Protein-Protein Interactions with Deep Neural Networks. Journal of Chemical Information & Modeling, 2017, 57 (6):1499-1510.

[29] Zhang YN, Pan XY, Huang Y, et al. Adaptive compressive learning for prediction of protein-protein interactions from primary sequence, Journal of Theoretical Biology, 2011; 283(1):44-52. pmid: 21635901.