Mathematical Biosciences and Engineering

MBE, 19(3): 2471–2488.
DOI: 10.3934/mbe.2022114
Received: 18 July 2021
Received: 08 November 2021
Accepted: 10 December 2021
Published: 07 January 2022

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

Research article

Deep neural learning based protein function prediction

Wenjun Xu[2,3,4], Zihao Zhao[1,2,3], Hongwei Zhang[1,2,3], Minglei Hu[1,2,3], Ning Yang[1,2,3], Hui Wang[1,2,3], Chao Wang[1,2,3], Jun Jiao[1,2,3] and Lichuan Gu[1,2,3,4,*]

[1] School of Information and Computer, Anhui Agricultural University, Hefei 230036, China
[2] Key Laboratory of Agricultural Electronic Commerce, Ministry of Agriculture, Hefei 230036, China
[3] Institute of Intelligent Agriculture, Anhui Agricultural University, Hefei 230036, China
[4] School of Life Sciences, Anhui Agricultural University, Hefei 230036, China

* **Correspondence:** Email: glc@ahau.edu.cn; Tel: +86055165786188; Fax: +86055165786188.

Abstract: It is vital for the annotation of uncharacterized proteins by protein function prediction. At present, Deep Neural Network based protein function prediction is mainly carried out for dataset of small scale proteins or Gene Ontology, and usually explore the relationships between single protein feature and function tags. The practical methods for large-scale multi-features protein prediction still need to be studied in depth. This paper proposes a DNN based protein function prediction approach IGP-DNN. This method uses Grasshopper Optimization Algorithm (GOA) and Intuitionistic Fuzzy c-Means clustering (IFCM) based protein function modules extracting algorithm to extract the features of protein modules, utilizing Kernel Principal Component Analysis (KPCA) method to reduce the dimensionality of the protein attribute information, and integrating module features and attribute features. Inputting integrated data into DNN through multiple hidden layers to classify proteins and predict protein functions. In the experiments, the F-measure value of IGP-DNN on the DIP dataset reaches 0.4436, which shows better performance.

Keywords: protein function prediction; protein-protein Interaction(PPI); DEEP Neural Network(DNN); Kernel Principal Component Analysis(KPCA); Grasshopper Optimization Algorithm(GOA)

1. Introduction

Protein function prediction is a classification problem of multiple labels that fills up the gap between a large number of protein sequences and known functions. The prediction is a challenging

research direction in biology and plays an important role in grasping the tissues and functions of the biological system [1]. Traditional biology experiments predict protein functions by extracting useful information from protein sequences. However, these approaches have a slow speed and high cost. On the contrary, computational methods are widely used in protein function prediction because of their low cost and ease of implementation [2].

Protein functions are mainly predicted by using protein features, including amino acid sequence [2,3], 3-D protein structure [4], protein-protein interaction (PPI) network [5] and the other molecular and functions [6,7]. Machine learning, which is widely used to predict protein function, uses features extracted from protein properties to train classification models, such as Artificial Neural Networks (ANNs) [8–10]. Deep Neural Network (DNN) is a subclass of ANNs which builds more advanced features in each subsequent layer with the input of initial features.

This paper proposes a protein function prediction approach that combines Kernel Principal Component Analysis (KPCA) with DNN. The approach firstly uses a feature extraction algorithm called IFCM-GOA that extracts the initial features of protein function modules and protein properties by using Grasshopper Optimization Algorithm and Intuitionistic Fuzzy c-Means. In order to remove the redundant information, KPCA is used to reduce the dimension of the initial features. The processed features are input into DNN to predict protein function.

2. Related work

There are four types of computation approaches for protein function prediction, including prediction based on sequence similarity, prediction based on PPI network, prediction based on protein structure similarity and prediction based on the other protein information [11]. Altschul [12] proposed BLAST. The BLAST programs are widely used tools for searching protein and DNA databases for sequence similarities. For protein comparisons, a variety of definitional, algorithmic and statistical refinements described here permits the execution time of the BLAST programs to be decreased substantially while enhancing their sensitivity to weak similarities. Yunes [13] proposed a protein function prediction approach called Effusion. Sovan [14] explored a dynamic PPI network that was made up of neighboring protein of level 1 and 2 at different times. Hoffmann [15] proposed a new method to quantify the similarity between pockets and studied its correlation with ligand prediction. Yang [16] predicted protein function by using the digital features of the protein sequence.

DNN has the ability to the representation of multiple hidden layers and data abstractions. It has been widely used in computer vision, natural language process, protein function prediction and other fields [17–20]. DNN is able to mine a more complex correlation between protein features and labels by setting activation function, depth of hiding level and other parameters. The common DNN algorithms are single task or multiple task feed-forward DNN [6], Auto-encoder [5], restricted Boltzmann [7], convolution neural network [3] and other DNN algorithms. In recent years, several scholars gradually studied the prediction of useful protein functions in large-scale protein based on DNN. In 2017, Cao [2] proposed ProLanGo that used recurrent neural network and protein sequence. In 2018, Kulmanov [21] proposed DeepGO that used deep learning to learn features from protein sequences as well as a cross-species protein-protein interaction network. This approach specifically outputs information in the structure of the GO and utilizes the dependencies between GO classes as background information to construct a deep learning model. In 2019, to address the problems during training features of the large-scale protein, Ahmet [8] proposed a layered stack based on multitasking

feed-forward DNN that is a solution of prediction based on GO. The above DNN-based approaches just mine a single protein feature and ignore the correlation between protein features with multiple biology information and labels. In 2019, You [22] proposed NetGO, which is based on GOLabeler-a state-of-the-art method for the third critical assessment of functional annotation (CAFA3). NetGO is a web server that is able to further improve the performance of the large-scale AFP by incorporating massive protein-protein network information. Experimental results have clearly demonstrated that NetGO significantly outperforms GOLabeler and other competing methods. In 2021, Yao [23] proposed the updated version as NetGO 2.0, which further improves the performance of large-scale AFP. NetGO 2.0 also incorporates literature information by logistic regression and deep sequence information by recurrent neural network (RNN) into the framework.

3. DNN-based protein function prediction

3.1. Algorithm process

This paper proposes a DNN-based protein functions prediction approach IGP-DNN of which input includes the features of PPI network module and protein property and the annotation terms of protein function. The approach builds a DNN model to predict the annotation terms of unknown protein function. Figure 1 shows the algorithm process.

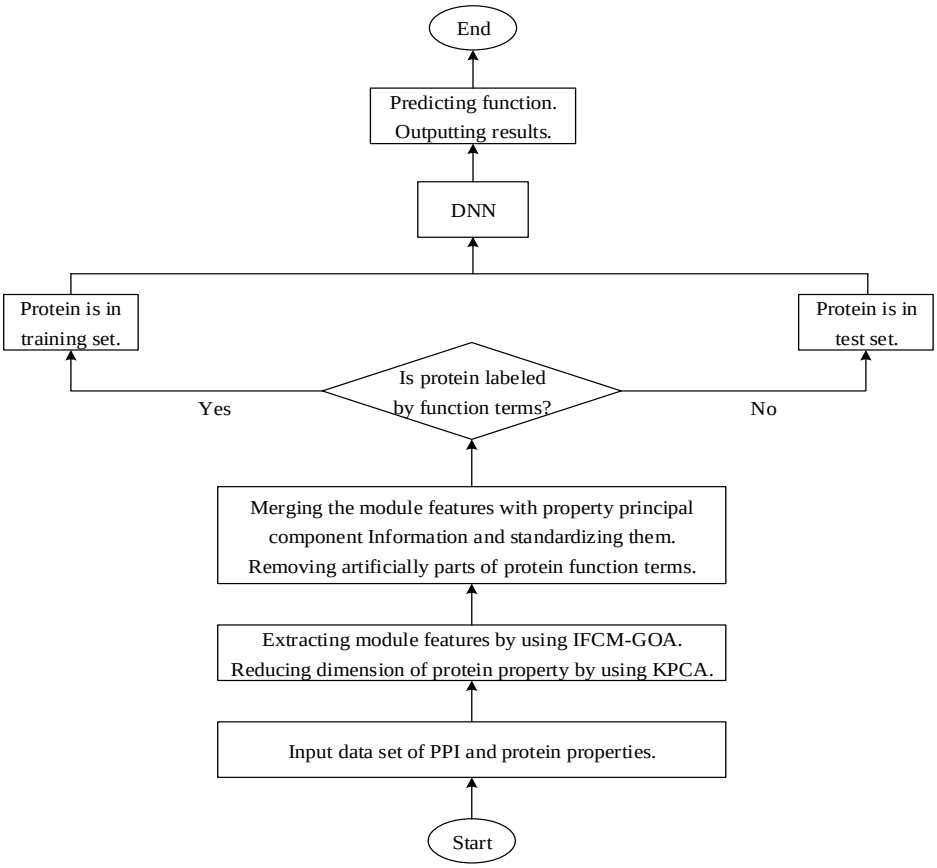


Figure 1. IGP-DNN algorithm process.

3.2. Vector construction of protein features

Vectors of protein features are built by using IFCM-GOA and KPCA. IFCM-GOA is used to extract module features. KPCA is used to reduce the dimension of protein property. The features that are reduced dimension and standardized are inputted to the DNN model.

3.2.1. IFCM-GOA

IFCM-GOA uses the idea that the grasshopper optimization algorithm (GOA) ingeniously balances the two processes of exploration and development to optimize and search for the best cluster center. According to the best cluster center, the intuitionistic fuzzy c-means (IFCM) cluster calculates the intuitionistic fuzzy membership matrix. Cluster results are obtained by dividing the matrix.

Assume undirected graph $G=(V,E)$ denotes data of PPI network, where $V=\{v_1,v_2,\cdots,v_n\}$ is vertexes set and v denotes proteins. And $V_a=\{v_1,v_2,\cdots,v_{nl}\}$ is proteins set with known functions. $V_b=\{v_{nl+1},v_{nl+2},\cdots,v_n\}$ is proteins set with unknown function. The vertexes in G are sort from v_1 to v_n. $E=\{e_{ij}\,|\,e_{ij}=<v_i,v_j>,v_i,v_j\in V\}$ is the set of edges. Element e_{ij} represents the interaction between protein v_i and v_j. Therefore, adjacency matrix G is represented by $A=(a_{ij})$. Element a_{ij} is $a_{ij}=\begin{cases}1,(v_i,v_j)\in E\\0,(v_i,v_j)\notin E\end{cases}$. IFCM-GOA input adjacency matrix $A=(a_{ij})$ to PPI network. The output is function module matrix $\Psi=(\psi_{ij}),i=1,2,\cdots,n,\ j=1,2,\cdots,l_1$ where the value of ψ_{ij} is 1 or 0. The value of ψ_{ij} means whether the protein belongs to a function module. Algorithm 1 shows the detailed steps of the IFCM-GOA.

Algorithm 1 IFCM-GOA Algorithm

Input: PPI network data
Output: Functional module
1: Initialize the population X_i (i=1,2,…,n), number of clustering, clustering center
2: Initialize cmax, cmin, maximum number of iterations
3: Calculate the membership matrix and the fitness of each search agent
4: **while** (l<Max number of iterations) **do**
5: GOA algorithm updates parameter and optimizes the best solution
6: **end while**
7: The best clustering center
8: Using the best clustering center to clustering

3.2.2. Feature vector building

IFCM-GOA is used to dig some function modules that are represented by a set $\Psi = (\psi_{ij}), i = 1, 2, \cdots, n, j = 1, 2, \cdots, l_1$. The protein properties, which including protein family, structure domain and binding sit, are integrated with function module features to generate new features that are used to train DNN. It improves the generalization of the prediction model. Assume $H(h_1, h_2, \cdots, h_m)$ represents for the set of protein properties. $Q = (q_{ij}), i = 1, 2, \cdots, n, j = 1, 2, \cdots, m$ denotes whether the protein has a property. The value of q_{ij} is 1 or 0, which denotes whether the protein has a property. The performance of the DNN model could lose the generalization because of the high-dimension and discreteness of protein. This paper uses KPCA to extract the principal component of the properties and reduce the dimension of $Q = (q_{ij})$. It reduces the impact of noise and improves the probability of success and efficiency. Firstly, kernel matrix $\Phi(Q) = \Phi((q_{ij}))$ is obtained by using non-linear function Φ(Gaussian kernel function) to map $Q = (q_{ij})$ to high dimension. The matrix is centralized and satisfies $\sum_{i=1}^{n} \Phi(q_i) = 0$. Secondly, the feature values of Φ(Q) are calculated in Jacobin iteration and sorted in descending to extract the correlate feature vectors. Thirdly, the feature vectors are orthogonalized by using Schmidt. The values of cumulative contribution rate of the features are calculated. More than 90% of former l_2 principal component are chosen. Finally, the principal component matrix $Q' = (q'_{ij}), i = 1, 2, \cdots, n, j = 1, 2, \cdots, l_2$ is obtained by calculating the project of the kernel matrix on feature vectors.

The feature matrix of the protein is generated by horizontally merging the features $\Psi = (\psi_{ij}), i = 1, 2, \cdots, n, j = 1, 2, \cdots, l_1$ of the function module with the principle matrix $Q' = (q'_{ij}), i = 1, 2, \cdots, n, j = 1, 2, \cdots, l_2$ of the property. The feature matrix of the protein reflects the protein information on both the macro and micro levels. The element x_{ij} is represented as follow.

$$x_{ij} = \begin{cases} \psi_{ij}, i = 1, 2, \cdots, n, j = 1, 2, \cdots, l_1 \\ q'_{ij}, i = 1, 2, \cdots, n, j = 1, 2, \cdots, l_1 \end{cases} \tag{1}$$

Defining $C = \{c_1, c_2, \cdots, c_w\}$ is the term set of function. $Y = (y_{ij}), i = 1, 2, \cdots, n, j = 1, 2, \cdots, w$ is information label matrix of the known protein function module. The element is represented as the follow.

$$y_{ij} = \begin{cases} 1, if \ \ protein\ v_i\ is\ commented\ by\ c_j \\ 0, if \ \ protein\ v_i\ is\ not\ commented\ by\ c_j \end{cases} \tag{2}$$

The prediction model transforms protein function prediction into a binary classification problem with multiple labels of which the samples are proteins and the sample labels are function module terms.

The feature matrix $X=(x_{ij}), i=1,2,\cdots,n, j=1,2,\cdots,l_1+l_2$ is min-max standardized to improve the training convergence speed of the DNN. According to formula (3), the standardized feature matrix $X'=(x'_{ij}), i=1,2,\cdots,n, j=1,2,\cdots,l_1+l_2$ is obtained and x'_{ij} are mapped to the interval [0,1].

$$x'_{ij}=\frac{x_{ij}-\min(x_j)}{\max(x_j)-\min(x_j)} \tag{3}$$

3.2.3. Protein function prediction

Deep learning solves the regression and classification problems by extracting the feature representation from the input data with different abstraction levels. In this paper, the DNN model IGP-DNN that predicts protein function is built. The input of IGP-DNN is the standardized feature matrix $X'=(x'_{ij})$ of the protein with known function in PPI network. And the output is the information label matrix of the corresponding protein function module.

The performance of the protein function prediction by using DNN is affected by many factors. It is important for the accuracy of the final prediction results to select the suitable hyper-parameters. In this paper, we determine the several hyper parameters by using enumeration. The hyper-parameters include the number of hidden layer and the number of neurons in each hidden layer. In the experiments, the number of the hiding levels is set as {5, 10, 15, 20, 25} and the number of neurons in each hidden layer is set as {1000, 2000, 3000, 4000, 5000}. DNN model with different hyper-parameters is trained in the three training sets. The IGP-DNN is finally determined by comparing the model performance on test set. According to the experience of DNN, the other hyper-parameters of the IGP-DNN are determined. For example, the number of nodes in the input level is the number of dimensions of the standardized feature matrix $X'=(x'_{ij})$ and the number of nodes in the output level is the number of each protein dataset function comment. The function *ReLU*, of which the formula is shown in the formula (4), is selected as the activation function of the hidden level.

$$ReLU=\max(x,0) \tag{4}$$

The function *tanh*, shown as formula (5), is selected as the activation function of the output level.

$$tanh=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{5}$$

The loss function is the cross-entropy loss function of which the basis is the maximum likelihood estimation. The formula of the loss function is shown as the formula (6).

$$L=-[y\log\hat{y}+(1-y)\log(1-\hat{y})] \tag{6}$$

The adaptive learning algorithm is selected as the optimization algorithm of the experiment. The value of the β_1 is 0.9, the β_2 is 0.999, the ε is 10^{-8}. The IGP-DNN is trained by using batch learning of which

the size is 20% of the number of the protein in the training set. The number of learning iteration *Epoch* is 200. The regularization coefficient is 0.0005 and the proportion of *Dropout* is 30%. The IGP-DNN predicts the probability that an unknown protein has a certain function and selects the *K* comment terms with the highest prediction probability as protein functions. *K* is the average of each protein functions.

4. Experiments

4.1. The Effectiveness of GOA

We use Matlab software to simulate the effectiveness of the GOA in single-mode, multi-mode and compound test functions.

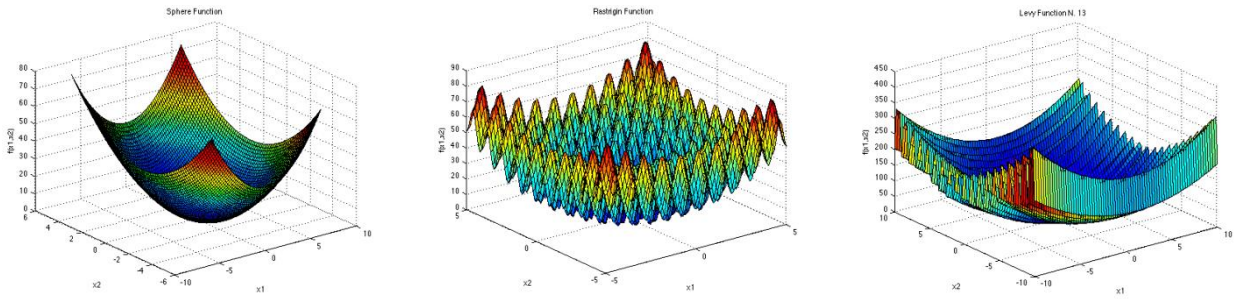


Figure 2. Unimodal, multi-modal and composite test functions.

In Figure 2, unimodal test function has no local solution, only a globally optimal solution, and the entire search space tends to be globally optimal, so it can be used as a benchmark. Multi-modal and composite test functions have many local optima, which makes them very suitable for benchmarking GOA in terms of local optima avoidance and exploration. At the same time, the composite test function usually simulates the real search space better than the multi-modal test function, so the potential performance of the GOA in solving practical problems can be inferred from this.

In Figure 3, five graphs were drawn, including the shape of the test function; Search history, which shows the historical position of the grasshopper during the optimization process; Trajectory of the first grasshopper, which shows the first grasshopper in each iteration value of the first variable in; Average fitness of all grasshoppers, the figure shows the average target value of all grasshoppers in each iteration; Convergence curve, the figure shows the target value of the best solution obtained in each iteration. The trajectory curve in Figure 3 shown that in the initial iteration process, the grasshopper fluctuates greatly due to the higher repulsive force rate. At this time, the grasshopper is in the exploratory stage and is more inclined to move quickly to a new place to find the optimal solution. However, due to the adaptive comfort zone and attractiveness between the grasshoppers, the fluctuations gradually decrease as the optimization process progresses. This ensures that the GOA cleverly weighs the two processes of exploration and development, and finally converges to the global optimal solution. The graphs of the average fitness of all grasshoppers and the convergence curve show the descent behavior of the three test functions, which proves that the GOA improves the initial random population of the test function and improves the accuracy of the approximate optimal solution in the

iterative process. Generally, for different test functions, grasshoppers tend to explore promising areas in the search space, so that the grasshoppers develop in the direction of the global optimum, and finally gather around the global optimal value, effectively solving the local optimal problem.

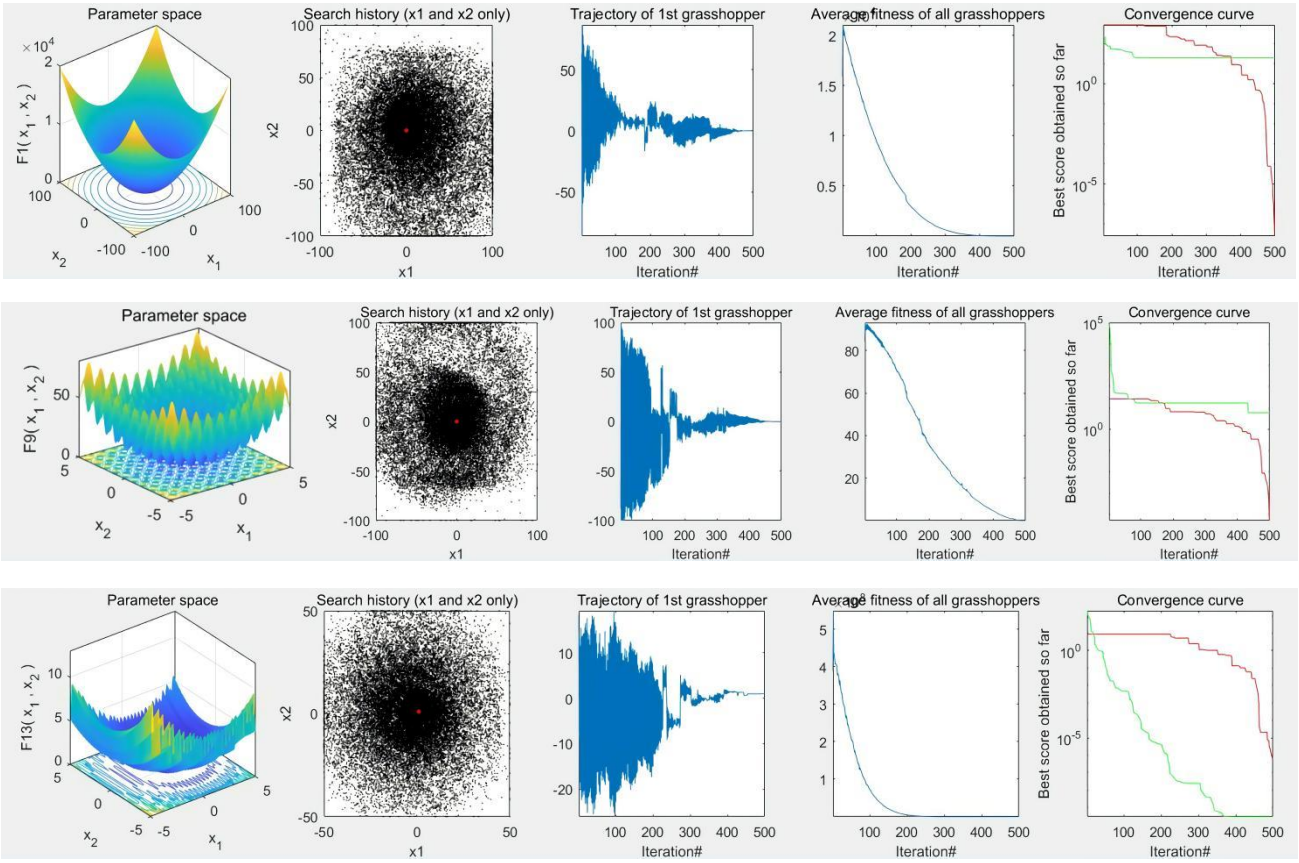


Figure 3. GOA tests on the standard functions.

4.2. Dataset

The experiment selects the Yeast Protein dataset from Database of Interacting Protein (DIP) [24] as the dataset of PPI network. The PPI network data of the Yeast Protein is downloaded from DIP and the protein IDs are transformed by using UniProtKB/Swiss-Prot. According to the protein number in UniProtKB/Swiss-Prot, database GO [25] and database InterPro [26] are obtained the corresponding GO term number and InterPro number. Database GO and InterPro provide the protein function comment and property information, respectively. The database Gavin [27] and Kragon [28] are the other two datasets for experimental verification. The former provides a small-scale data collection of budding yeast proteins that is often used to test the effectiveness of algorithms. The latter contains 2674 proteins and 7075 interactions. Table 1 shows the detailed information of the 3 PPI network datasets.

Table 1. Caption of the table.

Dataset	Number of nodes	Number of edges	Average of nodes	Number of GO terms	Number of property features
DIP	4579	20845	6.98	5879	10221
Gavin	1430	6531	9.13	1963	3297
Kragon	2674	7075	5.29	3505	5996

The deep learning framework is Tensorflow1.8. According to the IFCM-GOA, the function modules of which the number of clusters is 410, 130, 250 are selected as function module features.

4.3. Measure

The experiment uses cross-validation. The functions of protein in the test set are predicted on the basis of the training set. The performance of IGP-DNN is verified by comparing the predicted functions with the actual functions. The common approaches for measuring protein function prediction include the value of Precision, Recall and F-measure. The prediction results of protein function are divided into positive data that is verified by the experiment and negative data which is verified as no function. The right prediction in positive data is called true positive (*TP*), the wrong prediction is called false positive (*FP*), the right prediction and wrong prediction in the negative data are called true negative (*TN*) and false negative (*FN*), respectively. The average value ΔTP, ΔFP, ΔTN, ΔFN of *TP*, *FP*, *TN*, *FN* are calculated to measure the performance of IGP-DNN. The formula (7–9) shows the definition of the measure approaches.

$$Precision = \frac{\Delta TP}{\Delta TP + \Delta FP} \tag{7}$$

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

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

4.4. Results

To verify the performance advantage of the model that combines protein function module features with property principle features, this paper firstly uses KPCA to reduce features dimension on the datasets DIP, Gavin and Kragon. The results of dimension reduction are compared with the results by using the kernel independent principal component analysis (KICA) and the local linear embedding (LLE). Its effectiveness is verified. And then the different hyper-parameters are set and compared to choose the most suitable hyper-parameters to build the model IGP-DNN. Thereby, the performance of KPCA that reduces dimension is verified again. Finally, the IGP-DNN is compared with BLAST [12], IGP-SVM, HPMM [29], FFPred [30], and DeepGo [21] on three datasets. We use the BLAST as a baseline. The IGP-SVM combines the features of protein functional module extracted by IFCM-GOA with the protein attribute information obtained after KPCA dimension reduction as the protein feature,

and the SVM classifier is used for classification; The HPMM develops a protein feature extraction based on hierarchical clustering and principal component analysis, And combined with the method of multi-layer perceptron for protein function prediction; The FFPred uses machine learning methods, using the features predicted from the amino acid sequence to perform function prediction in the protein feature space. The analysis of experimental results verifies the superiority of IGP-DNN in predicting the function of unknown proteins.

The high dimensions of the protein property feature will affect the results of the algorithm. Therefore, IGP-DNN firstly reduces the dimension of the protein property by using KPCA. The results of dimension reduction are compared with the results by using KICA and LLE. The KICA found hidden components from the multiple dimension data. The LLE reflected global non-linear by using local linear. The results of dimension reduction are illustrated in Tables 2–4.

Table 2. The results of dimension reduction on DIP.

Algorithm	Number of property feature	Number of property principal component feature	Rate of dimension reduction
KPCA	10221	2698	73.60%
KICA	10221	3475	66.00%
LLE	10221	4793	46.90%

Table 3. The results of dimension reduction on Gavin.

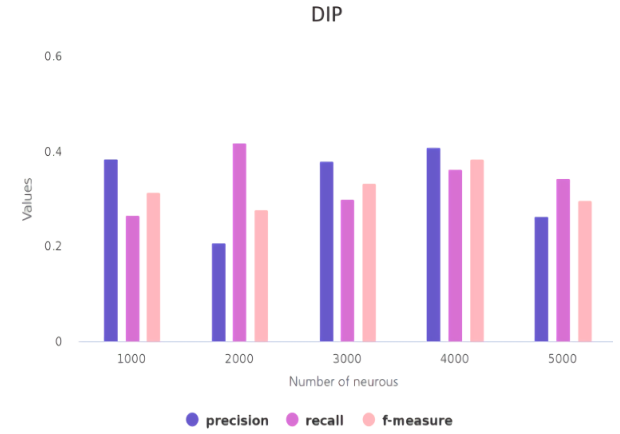
Algorithm	Number of property feature	Number of property principal component feature	Rate of dimension reduction
KPCA	3297	910	72.40%
KICA	3297	930	71.80%
LLE	3297	2008	39.10%

Table 4. The results of dimension reduction on Krogan.

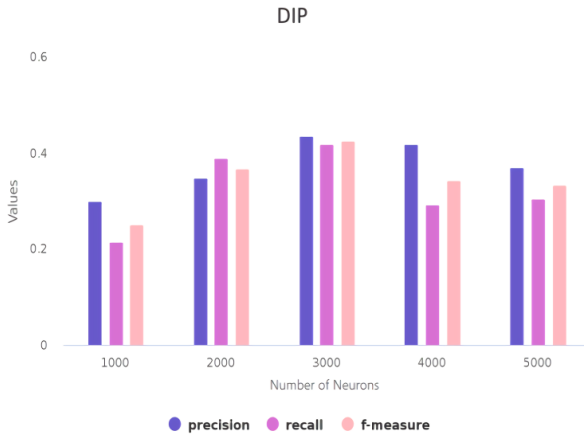
Algorithm	Number of property feature	Number of property principal component feature	Rate of dimension reduction
KPCA	5996	1625	72.90%
KICA	5996	2272	62.10%
LLE	5996	3610	39.80%

As shown in Table 2–4, the rate of dimension reduction of KPCA is 73.6%, 72.4%, 72.9% on the three datasets, respectively. The results of KPCA is slightly better than KICA and obviously better than LLE. The KPCA reduced the number of initial property features from 10221 to 2698, but KICA and LLE reduced to 3475 and 4793, respectively. The results show that KPCA can effectively reduce dimension, but it is not able to ensure the precision of prediction.

To verify the effectiveness of KPCA, the different dimension reduction approaches and hyper-parameters are chosen to carry out a comparative experiment and determine the most suitable hyper-parameters for building IGP-DNN. For two hyper-parameters and three-dimension reduction approaches, the multi-group comparative experiments are carried out on three datasets. The results are illustrated in Figures 4–6.

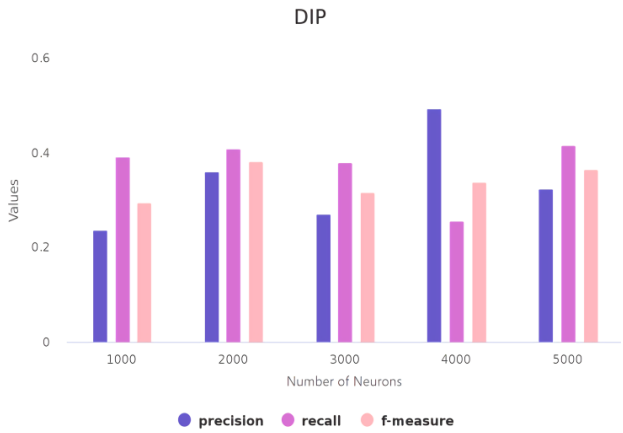


(a) five hidden layers

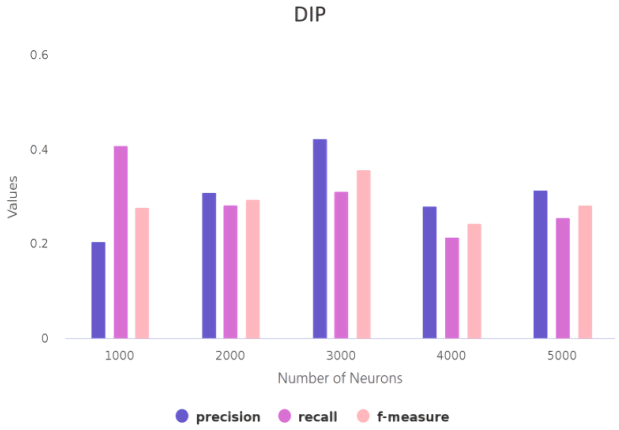


(b) ten hidden layers

(c) fifteen hidden layers

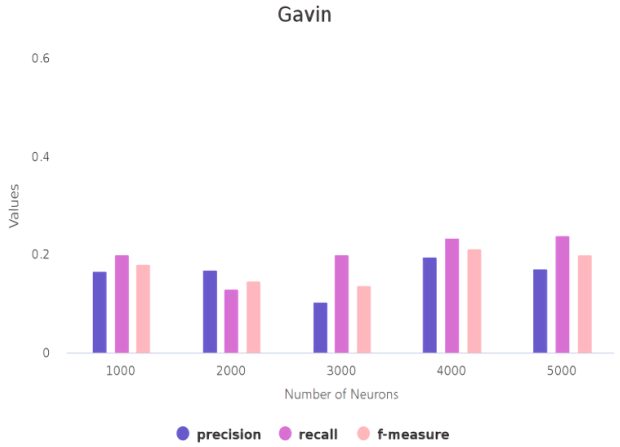


(d) twenty hidden layers

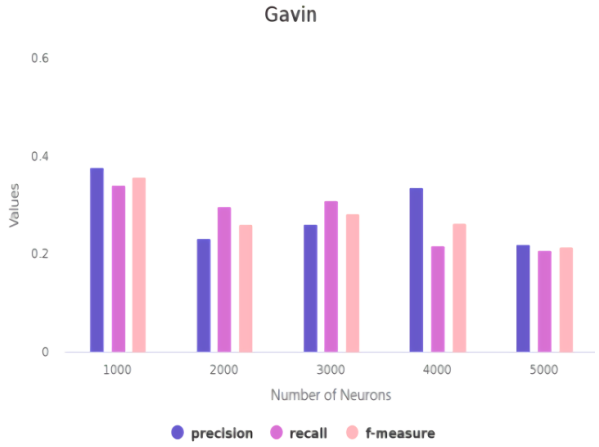


(e) twenty-five hidden layers

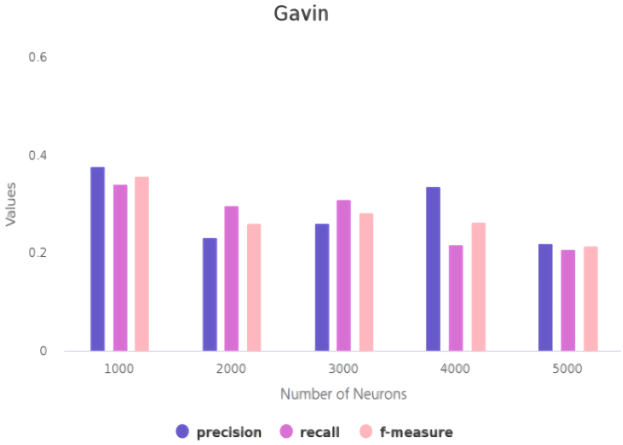
Figure 4. Results with different hyper parameter on DIP.



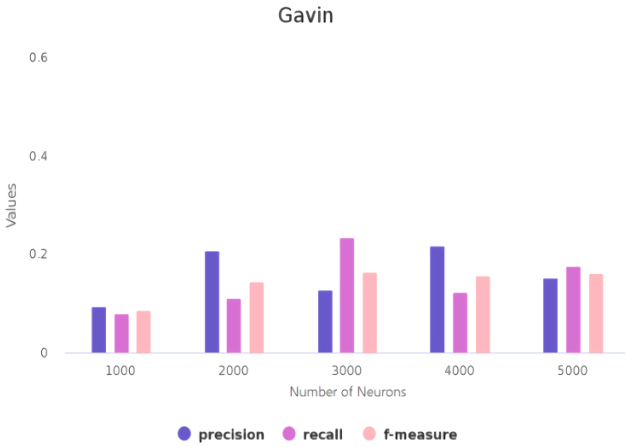
(a) five hidden layers



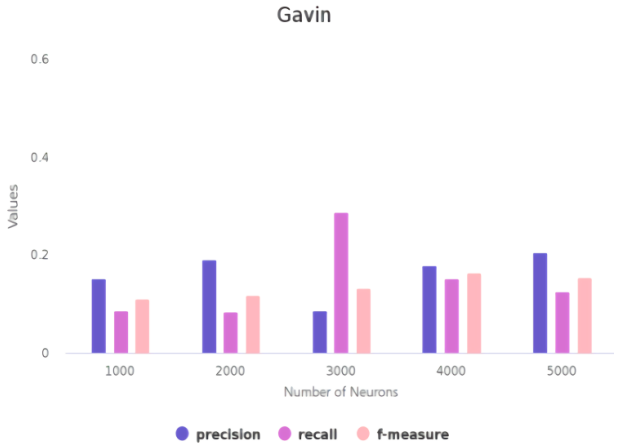
(b) ten hidden layers



(c) fifteen hidden layers

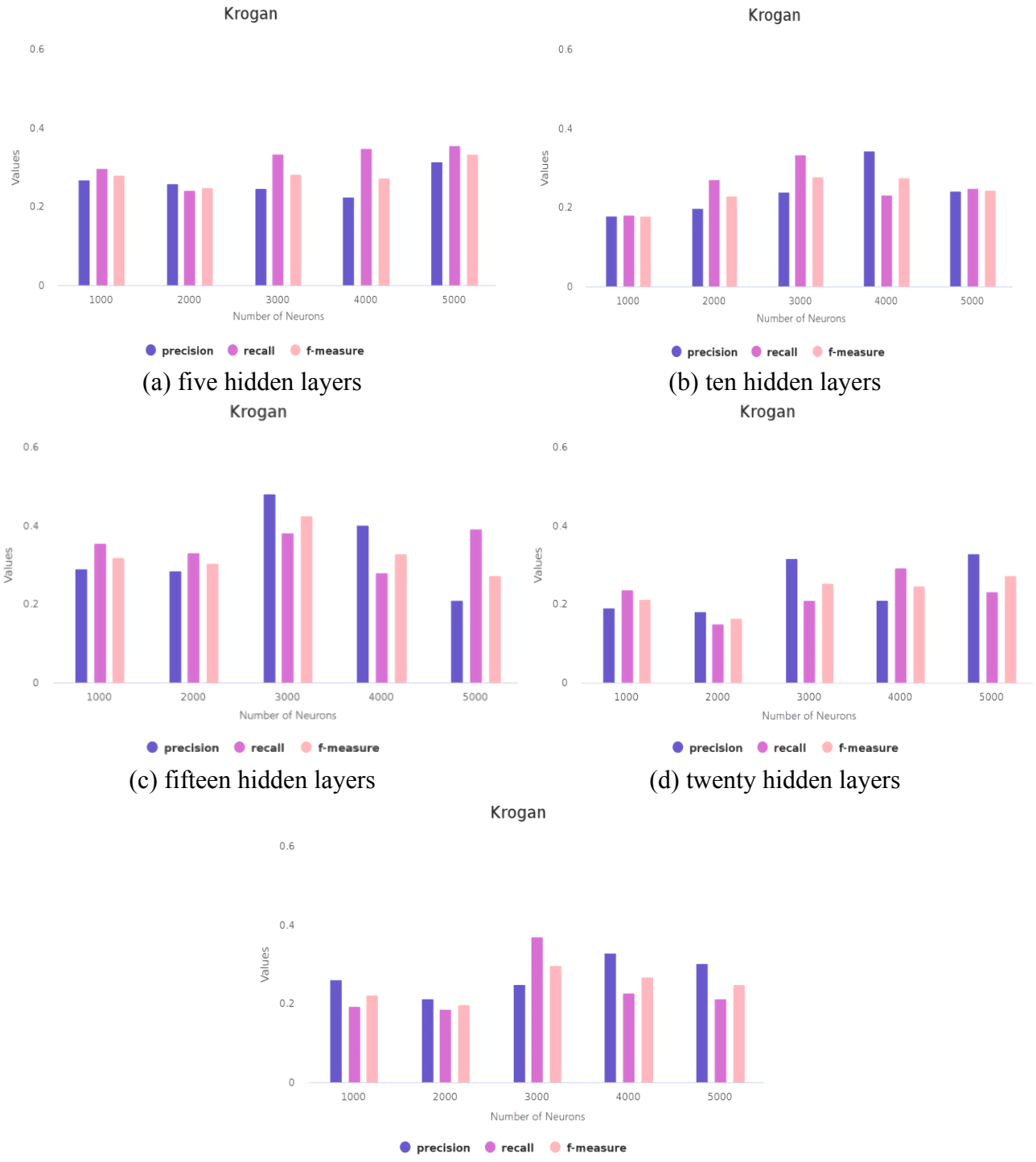


(d) twenty hidden layers



(e) twenty-five hidden layers

Figure 5. Results with different hyper parameter on Gavin.



(a) five hidden layers

(b) ten hidden layers

(c) fifteen hidden layers

(d) twenty hidden layers

(e) twenty-five hidden layers

Figure 6. Results with different hyper parameter on Krogan.

Figures 4–6 illustrate the results of the IGP-DNN model that the dimension reduction approach is KPCA. As shown in Figure 4, the performance of the model is the best on DIP when the number of hiding level is 20 and the number of neurons is 4000. As shown in Figure 5, the performance of the model is the best on Krogan when the number of hiding level is 20 and the number of neurons is 3000. As shown in Figure 6, the performance of the model is the best on Gavin when the number of hiding

levels is 15 and the number of neurons is 1000. It can be seen from the experimental results that the combination of the different number of hiding level and neurons have a great impact on the results. The reason is that the precision of the DNN prediction model is mainly impacted by the number of hiding levels and neurons. Compared with the model that has a single hidden layer, the model with the multiple hidden layers significantly improves the precision.

Table 5 shown the performance of IGP-DNN with different dimension reduction methods and hyperparameters on three datasets. It can be seen that the IGP-DNN model built on the DIP and Krogan datasets after using the KPCA method to reduce the dimensionality is significantly better than the two dimensionality reduction methods of KICA and LLE in predicting protein function. At the same time, it is also better than the method without dimensionality reduction. Only the Precision and Recall on the Gavin dataset is slightly lower than the KICA method, but higher than the LLE method and the method without dimensionality reduction. The experimental results show that the KPCA method can not only effectively reduce the feature dimensions and reduce the impact of excessively high dimensions, but also KPCA can effectively retain the important features of protein attribute information when reducing the dimensionality, ensuring the accuracy of protein function algorithm prediction.

Table 5. Experimental results of dimension reduction method.

Dimension Reduction Method	Dataset	Number of Hidden Layers	Number of Neurons	Precision	Recall	F1
KPCA	DIP	20	4000	0.4903	0.4051	0.4436
	Krogan	20	3000	0.4809	0.3817	0.4256
	Gavin	15	1000	0.3776	0.3409	0.3583
KICA	DIP	20	2000	0.2631	0.4252	0.3251
	Krogan	10	2000	0.3744	0.2488	0.2989
	Gavin	5	1000	0.4109	0.3267	0.364
LLE	DIP	10	4000	0.1396	0.1199	0.129
	Krogan	10	1000	0.2417	0.2847	0.2614
	Gavin	15	2000	0.2406	0.1085	0.1496
Naive	DIP	5	1000	0.1061	0.0899	0.0973
	Krogan	5	2000	0.1249	0.1947	0.1522
	Gavin	15	1000	0.2037	0.1175	0.149

As shown in Table 6, the IGP-DNN model with the best hyper parameters was compared with the BLAST, IGP-SVM, HPMM, FFPred and DeepGO on three datasets.

It can be seen from Table 6 that the Recall of IGP-DNN is slightly lower than HPMM on Gavin, but it has more advantages on the other two datasets. The Precision and F-measure of IGP-DNN are higher than HPMM on the three datasets. The reason is that IGP-DNN uses IFCM and GOA to solve the problems that the PPI network is easy to fall into local optimal and be disturbed by noise points during clustering when extracting the protein function features. In addition, KPCA can deal with nonlinear data better. The Precision, Recall and F-measure of the IGP-DNN model are better than IGP-SVM, because DNN is better than SVM and has stronger nonlinear fitting ability while they process large scale dataset. The performance of the IGP-DNN model is obviously better than FFPred and more stable. The Recall of IGP-DNN is slightly lower than DeepGO on Krogan, and the Precision is lower

than DeepGO on DIP, but it has more advantages on other dataset. The F-measure of IGP-DNN is higher than DeepGO on the three datasets. The false positive rate of IGP-DNN is relatively low, although DeepGO has a better Recall on the dataset Krogan, its performance superiority is limited. This situation may be due to the superiorities of IFCM-GOA and DNN are integrated into IGP-DNN. Through the integration of IFCM and GOA, the problem of falling into local optimal when extracting function module features is avoided, and lowering the over-sensitivity of IFCM for clustering center, meanwhile, the precision of the protein function module features extraction is obviously improved. The integration of multiple protein properties, including protein family, structure domain and binding site, and features of protein function module can better reflect the micro and macro level information of protein and improve the generalization ability of predictive models. Meanwhile, KPCA reserves the important information of principle features. It finally improves the precision of the model to use the enumeration to choose the optimal hyper parameters. The feature values extracted by KPCA are more accurate and reliable, the final prediction accuracy is higher, and false positives can be better screened.

Table 6. Performance comparison of the protein function prediction approaches on three datasets.

Dataset	Method	Precision	Recall	F1
DIP	BLAST	0.3672	0.3771	0.372
	IGP-SVM	0.3825	0.1203	0.183
	HPMM	0.3897	0.2651	0.3155
	FFPred	0.1624	0.3208	0.2156
	DeepGo	**0.5121**	0.3758	0.4334
	IGP-DNN	0.4903	**0.4051**	**0.4436**
Krogan	BLAST	0.3212	0.3823	0.3490
	IGP-SVM	0.2846	0.1853	0.2245
	HPMM	0.3134	0.3825	0.3445
	FFPred	0.2669	0.2783	0.2725
	DeepGo	0.4544	**0.3823**	0.4152
	IGP-DNN	**0.4809**	0.3817	**0.4256**
Gavin	BLAST	0.3018	0.3272	0.3140
	IGP-SVM	0.1139	0.2605	0.1585
	HPMM	0.2179	0.4223	0.2875
	FFPred	0.2769	0.1346	0.1811
	DeepGo	0.3759	0.317	0.3439
	IGP-DNN	**0.3776**	**0.3409**	**0.3583**

5. Conclusions

A protein function prediction method based on DNN is proposed in this paper. In the proposed model, protein features consist of the protein function module features and the protein property features, the former is extracted by using IFCM-GOA and the latter are reduced dimensions by using KPCA. These protein features are aiming to settle the noise sensitivity and the other problems during the process of predicting protein function. In addition, enumeration is used to choose the optimal hyper

parameters that are the basis of building the DNN model. Then, the IGP-DNN is compared with the BLAST, IGP-SVM, HPMM, FFPred and DeepGO on three different datasets of the yeast PPI network. The experimental results demonstrate that the Precision, Recall, F-measure of IGP-DNN are better than BLAST, IGP-SVM, HPMM, FFPred and DeepGO, and IGP-DNN can effectively predict the unknown protein function.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (No.31771679), Major Scientific and Technological Projects in Anhui Province, China (No.201903a06020009, 201904e01020006, 202103b06020013), Natural Science Research Project of Anhui Provincial Department of Education (No.KJ2020A0107), Natural Science Foundation of Anhui Province (General Program) (2108085MF209), the 2019 Anhui University collaborative innovation project (GXXT-2019-013).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. L. C. Gu, Y. Y. Han, C. Wang, W. Chen, J. Jiao, X. Yuan, Module overlapping structure detection in PPI using an improved link similarity-based Markov clustering algorithm, *Neural. Comput. Appl.*, **31** (2019), 1481–1490. https://doi.org/10.1007/s00521-018-3508-z
2. R. Cao, C. Freitas, L. Chan, M. Sun, H. Jiang, Z. Chen, ProLanGO: protein function prediction using neural machine translation based on a recurrent neural network, *Molecules*, **22** (2017), 1732. https://doi.org/10.3390/molecules22101732
3. B. Szalkai, V. Grolmusz, SECLAF: a webserver and deep neural network design tool for hierarchical biological sequence classification, *Bioinformatics*, **34** (2018), 2487–2489. https://doi.org/10.1093/bioinformatics/bty116
4. A. Tavanaei, A. S. Maida, A. Kaniymattam, R. Loganantharaj, Towards recognition of protein function based on its structure using deep convolutional networks, In *2016 IEEE Int. Conf. Bioinform. Biomed.* (*BIBM*). *IEEE*, 2016, 145–149. https://doi.org/10.1109/BIBM.2016.7822509
5. V. Gligorijević, M. Barot, R. Bonneau, deepNF: deep network fusion for protein function prediction, *Bioinformatics*, **34** (2018), 3873–3881. https://doi.org/10.1093/bioinformatics/bty440
6. R. Fa, D. Cozzetto, C. Wan, D. T. Jones, Predicting human protein function with multi-task deep neural networks, *PloS one*, **13** (2018), e0198216. https://doi.org/10.1371/journal.pone.0198216
7. X. Zou, G. Wang, G. Yu, Protein function prediction using deep restricted Boltzmann machines, *BioMed Res. Int.*, **2017** (2017), 1729301. https://doi.org/10.1371/journal.pone.0198216
8. A. S. Rifaioglu, T. Doğan, M. J. Martin, R. Cetin-Atalay, V. Atalay, DEEPred: automated protein function prediction with multi-task feed-forward deep neural networks, *Sci. Rep.*, **9** (2019), 1–16. https://doi.org/10.1038/s41598-019-43708-3

9. C. J. Zhang, H. Tang, W. C. Li, H. Lin, W. Chen, K. C. Chou, iOri-Human: identify human origin of replication by incorporating dinucleotide physicochemical properties into pseudo nucleotide composition, *Oncotarget*, **7** (2016), 69783. https://doi.org/10.18632/oncotarget.11975
10. Y. Pan, D. Liu, L. Deng, Accurate prediction of functional effects for variants by combining gradient tree boosting with optimal neighborhood properties, *PloS one*, **12** (2017), e0179314. https://doi.org/10.1371/journal.pone.0179314
11. Y. Liu, S. Shen, H. Fang, K. X. Chen, An overview of protein function prediction methods, *Chin. J. Bioinform.*, **11** (2013), 33–38.
12. S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, et al., Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, *Nucleic Acids Res.*, **25** (1997), 3389–3402. https://doi.org/10.1093/nar/25.17.3389
13. J. M. Yunes, P. C. Babbitt, Effusion: prediction of protein function from sequence similarity networks, *Bioinformatics*, **35** (2019), 442–451. https://doi.org/10.1093/bioinformatics/bty672
14. S. Saha, A. Prasad, P. Chatterjee, S. Basu, M. Nasipuri, Protein function prediction from dynamic protein interaction network using gene expression data, *J. Bioinform. Comput. Biol.*, **17** (2019), 1950025. https://doi.org/10.1142/S0219720019500252
15. B. Hoffmann, M. Zaslavskiy, J. P. Vert, V. Stoven, A new protein binding pocket similarity measure based on comparison of clouds of atoms in 3D: application to ligand prediction, *BMC bioinform.*, **11** (2010), 99. https://doi.org/10.1186/1471-2105-11-99
16. A. Yang, R. Li, W. Zhu, G. Yue, A novel method for protein function prediction based on sequence numerical features, *Match-Commun. Math. Comput. Chem.*, **67** (2012), 833.
17. L. Deng, G. Hinton, B. Kingsbury, New types of deep neural network learning for speech recognition and related applications: An overview, *2013 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, 8599–8603. https://doi.org/10.1109/ICASSP.2013.6639344
18. C. Angermueller, T. Pärnamaa, L. Parts, O. Stegle, Deep learning for computational biology, *Mol. Syst. Boil.*, **12** (2016), 878. https://doi.org/10.15252/msb.20156651
19. S. Min, B. Lee, S. Yoon, Deep learning in bioinformatics, *Briefings Bioinform.*, **18** (2017), 851–869. https://doi.org/10.1093/bib/bbw068
20. R. Cao, B. Adhikari, D. Bhattacharya, M. Sun, J. Hou, J. Cheng, QAcon: single model quality assessment using protein structural and contact information with machine learning techniques, *Bioinformatics*, **33** (2017), 586–588. https://doi.org/10.1093/bioinformatics/btw694
21. M. Kulmanov, M. A. Khan, R. Hoehndorf, DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier, *Bioinformatics*, **34** (2018), 660–668. https://doi.org/10.1093/bioinformatics/btx624
22. R. You, S. Yao, Y. Xiong, X. Huang, F. Sun, H. Mamitsuka, et al., NetGO: improving large-scale protein function prediction with massive network information, *Nucleic Acids Res.*, **47** (2019), W379–W387. https://doi.org/10.1093/nar/gkz388
23. S. Yao, R. You, S. Wang, Y. Xiong, X. Huang, S. Zhu, NetGO 2.0: improving large-scale protein function prediction with massive sequence, text, domain, family and network information, *Nucleic Acids Res.*, 2021. https://doi.org/10.1093/nar/gkab398
24. I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, D. Eisenberg, DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions, *Nucleic Acids Res.*, **30** (2002), 303–305. https://doi.org/10.1093/nar/30.1.303

25. UniProt Consortium, The universal protein resource (UniProt) in 2010, *Nucleic Acids Res.*, **38** (2010), D142–D148. https://doi.org/10.1093/nar/gkp846
26. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. Cherry, et al., Gene ontology: tool for the unification of biology, *Nat. Genet.*, **25** (2000), 25–29. https://doi.org/10.1038/75556
27. S. Pu, J. Wong, B. Turner, E. Cho, S. J. Wodak, Up-to-date catalogues of yeast protein complexes, *Nucleic Acids Res.*, **37** (2009), 825–831. https://doi.org/10.1093/nar/gkn1005
28. A. C. Gavin, M. Bösche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, et al., Functional organization of the yeast proteome by systematic analysis of protein complexes, *Nature*, **415** (2002), 141–147. https://doi.org/10.1038/415141a
29. J. Q. Tang, J. L. Wu, Protein function prediction method based on PPI network and machine learning, *J. Comput. Appl.*, **38** (2018), 722–727.
30. A. E. Lobley, T. Nugent, C. A. Orengo, D. T. Jones, FFPred: an integrated feature-based function prediction server for vertebrate proteomes, *Nucleic Acids Res.*, **36** (2008), W297–W302. https://doi.org/10.1093/nar/gkn193

AIMS Press