Protein Function Prediction with Deep Neural Learning

Zihao Zhao
Anhui Agricultural University
Hongwei Zhang
Anhui Agricultural University
Minglei Hu
Anhui Agricultural University
Ning Yang
Anhui Agricultural University
Hui Wang
Anhui Agricultural University
Chao Wang
Anhui Agricultural University
Jun Jiao
Anhui Agricultural University
Lichuan Gu (glc@ahau.edu.cn)
Anhui Agricultural University

Research Article

Keywords: Protein function prediction, Deep Neural Networks (DNN), Kernel Principal Components Analysis (KPCA), Grasshopper Optimization Algorithm (GOA)

Posted Date: January 27th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-148762/v1

Protein Function Prediction with Deep Neural Learning

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], Lichuan Gu [1,2,3,*]

[*] Correspondence: glc@ahau.edu.cn

[1] School of Information and Computer, Anhui Agricultural University, Hefei 230036, China.

Full list of author information is available at the end of the article.

Abstract

Background: The function of protein is directly related to its structure, and plays a pivotal role in the entire life process. The protein interaction network controls almost all biological cell processes while fulfilling most of the biological functions. In fact, protein function prediction can be regarded as a multi-label classification problem to fill the gap between a huge number of protein sequences and known functions. It is not only a key issue in related research fields, but also a long-standing challenge. Protein function prediction with Deep Neural Network (DNN) almost study data set with small scale proteins based on Gene Ontology (GO). They usually dig relationships between protein features and function tags. It still needs further study for large-scale protein to find useful prediction approaches.

Methods: This paper proposed a protein function prediction approach with DNN which used Grasshopper Optimization Algorithm (GOA), Intuitionistic Fuzzy c-Means (IFCM), Kernel Principal Component Analysis (KPCA) and DNN (IGP-DNN). The features in protein function modules were extracted by combining GOA and IFCM. The KPCA was used to reduce the dimensions of features in protein properties. Both features were integrated to enrich the features information and the integrated

features were input into the DNN model. The protein function modules were classified to predict function by computing in hiding level of DNN.

Results and conclusion: IGP-DNN combines the advantages of IFCM-GOA and DNN. The combination of IFCM and GOA not only avoids falling into local optimal when extracting function module feature and reduces the over-sensitivity of IFCM for clustering center, but also improves the precision of the protein function module feature extraction. This paper proposes a protein function prediction approach based on DNN. In the model, protein features are composed of the protein function module features that are extracted by using IFCM-GOA and the protein property features that are reduced dimensions by using KPCA to address the noise sensitivity and the other problems during predicting protein function.

Key words: Protein function prediction; Deep Neural Networks (DNN); Kernel Principal Components 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 20. The traditional biology experiment predicts protein functions by extracting useful information from protein sequences. However, the approaches have a slow speed and high cost. On the contrary, computational methods is widely used in protein function prediction because of its 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][9] [10]. Deep Neural Network (DNN) is a subclass of ANNs which builds the more advanced features in each subsequent layer with 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 from 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, the prediction based on PPI network, the prediction based on protein structure similarity and prediction based on the other protein information [11]. Yunes [12] proposed a protein function protection approach called Effusion. Sovan [13] explored a dynamic PPI network that was made up of neighboring protein of level 1 and level 2 at different times. Hoffmann [14] proposed a new method to quantify the similarity between pockets and studied its correlation with ligand prediction. Yang [15] predicted protein function by using the digital features of the protein sequence.

DNN has the ability to the representation of multiple hide levels and data abstractions. It has been widely used in computer vision, natural language process, protein function prediction and other fields [16][17][18][19]. DNN is able to mine a more complex correlation between protein features and labels by setting activation function, depth of hiding level and the 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 the other DNN algorithms. In recent years, several scholars gradually studied useful protein function prediction of large-scale protein based on DNN. In 2017, Cao [2] proposed ProLanGo that used recurrent neural network and protein sequence. In 2019, to address the problems during training features of the large-scale protein, Ahmet [8] proposed a layered stack based on multitask 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 feature with multiple biology information and labels.

3. DNN-BASED PROTEIN FUNCTION PREDICTION

3.1. Algorithm Process

The paper proposes a DNN-based protein function prediction approaches IGP-DNN of which input is 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.

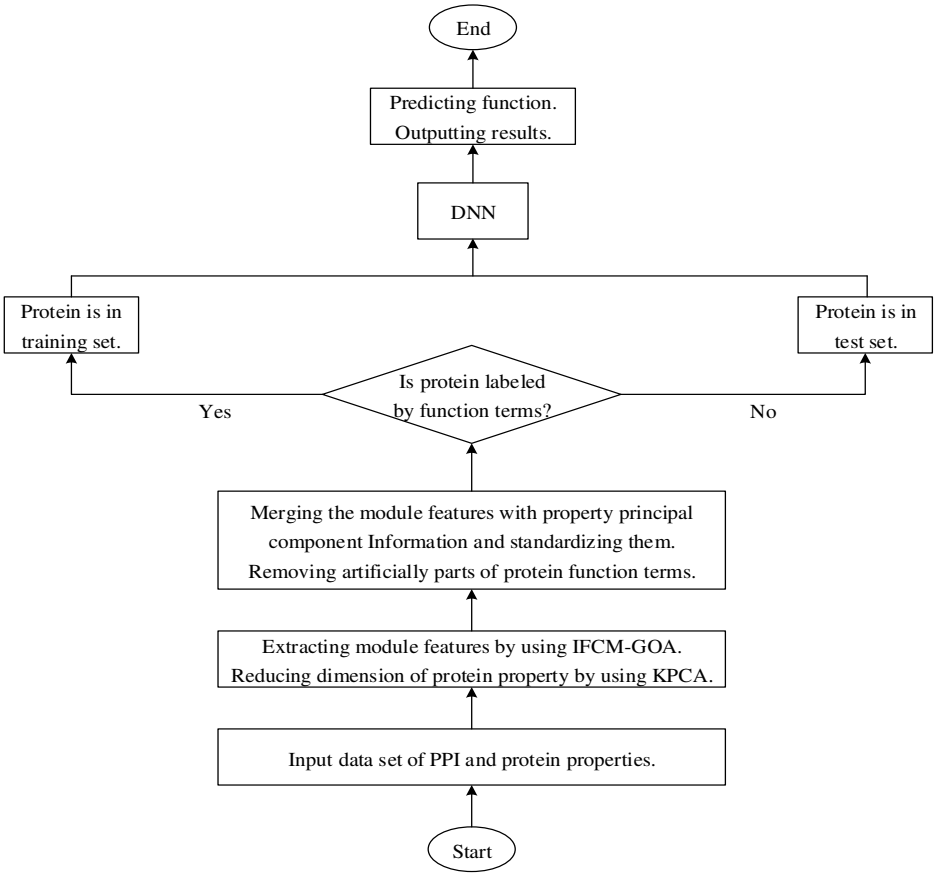


Fig 1 Process of DNN-based Protein function prediction

3.2. Vector construction of protein features

Vector of protein features is 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 input 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 $V=\{v_1,v_2,\ldots,v_n\}$ is vertex set and v denotes protein. And $V_a=\{v_1,v_2,\ldots,v_{nl}\}$ is protein set with known functions. $V_b=\{v_{nl+1},v_{nl+2},\ldots,v_n\}$ is protein set with unknown function. The vertex in G is 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 able to be 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 $\Psi=(\psi_{ij})$, $i=1,2,\ldots,n$, $j=1,2,\ldots,l_1$ where the value ψ_{ij} of is 1 or 0. The value of ψ_{ij} means whether the protein belongs to a function module. Table 1 shows the detailed steps of the IFCM-GOA.

Table 1. IFCM-GOA

Algorithm IFCM-GOA Algorithm
Input: PPI network data
Output: Functional module
1: Initialize the population X_i ($i=1,2,\ldots,n$), number of clustering, clustering center
2: Initialize c_{max}, c_{min}, 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 $\Psi=(\psi_{ij})$, $i=1,2,\ldots,n$, $j=1,2,\ldots,l_1$. 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 $H(h_1,h_2,\ldots,h_m)$ represents for the set of protein properties. $Q=(q_{ij})$, $i=1,2,\ldots,n$, $j=1,2,\ldots,m$ 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. The 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 $\Phi(Q)$ 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 cumulative contribution rate of the feature values is calculated. The former l_2 principal component of which is more than 90% is chosen. Finally, the principal component matrix $Q'=(q'_{ij})$, $i=1,2,\ldots,n$, $j=1,2,\ldots,l_2$ 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 $\Psi=(\psi_{ij})$, $i=1,2,\ldots,n$, $j=1,2,\ldots,l_1$ of the function module with the principle matrix $Q'=(q'_{ij})$, $i=1,2,\ldots,n$, $j=1,2,\ldots,l_2$

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.

$$x_{ij} = \begin{cases} \Psi_{ij}, i = 1,2,\dots,n, j = 1,2,\dots,l_1 \\ q'_{ij}, i = 1,2,\dots,n, j = 1,2,\dots,l_1 \end{cases} \quad (1)$$

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

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

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

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

3.3. Protein function prediction

Deep learning solves the regression and classification 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 hide level and the number of neurons in each hide level. In the experiments, the number of the hiding levels is set as {5,10,15,20,25} and the number of neurons in each hide level 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 data set 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.

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

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

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

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 function and selects the *K* comment terms

with the highest prediction probability as protein functions. *K* is the average of each protein functions.

4. EXPERIMENT

4.1. Data set

The experiment selects the Yeast Protein data set from the database Database of Interacting Protein [20] (DIP) as the data set 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 [21] and database InterPro [22] are obtained the corresponding GO term number and InterPro number. Database GO and database InterPro provide the protein function comment and the protein property information, respectively. The database Gavin [23] and Kragon [24] are the other two data set 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 2 shows the detailed information of the 3 PPI network data sets.

Table 2. Data sets

Data set	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.

4.2. 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 that is 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.

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

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

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

4.3. 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 data set 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.

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. KICA found hidden components from the multiple dimension data. LEE reflected global non-linear by using local linear. The results of dimension reduction are shown in Table 3-5.

Table 3. 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 4. The results of dimension reduction on Gavin

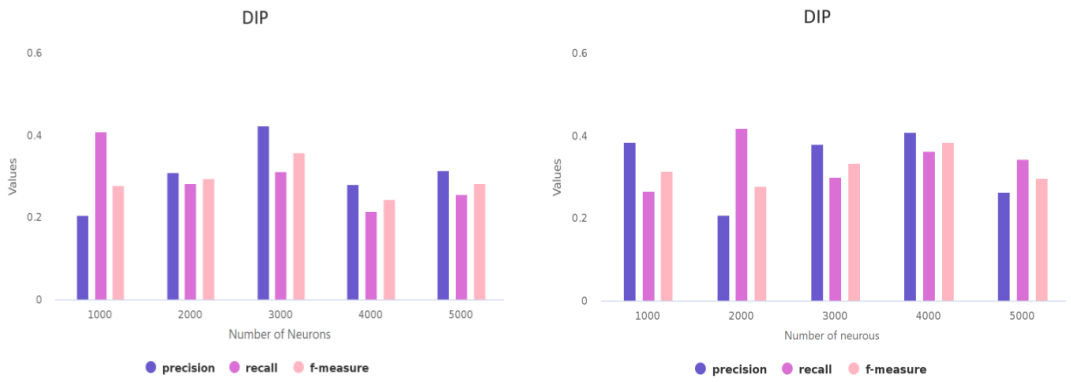
Algorithm	Number of property feature	Number of property principal component feature	Rate of dimension reduction
KPCA	3297	910	72.40%
KICA	3297	930	71.80%
LLE	3297	2008	39.10%

Table 5. 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%

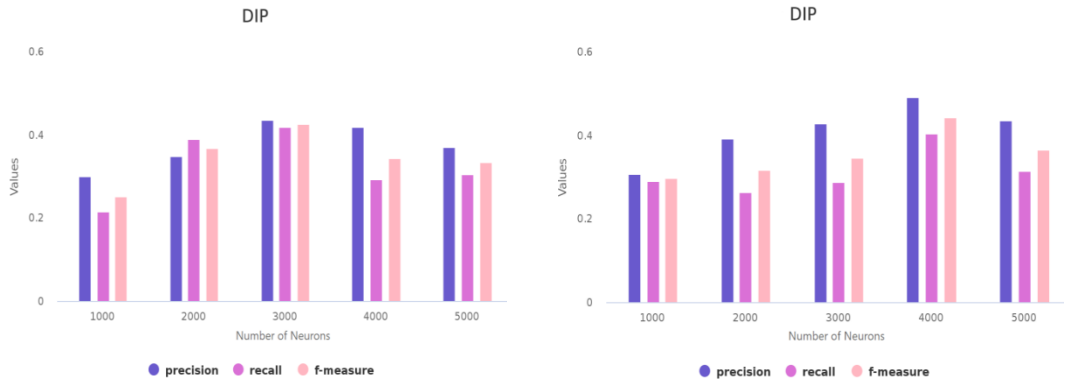
As shown in Table 3-5, the rate of dimension reduction of KPCA is 73.6%, 72.4%, 72.9% on the three data sets, respectively. The results of KPCA is slightly better than KICA and obviously better than LLE. The KPCA reduced the number of initial property features from 10221 to 2698, but KICA and LLE reduced 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 data sets. The results are shown in Figure 2-4.



(a) five hide levels

(b) ten hide levels

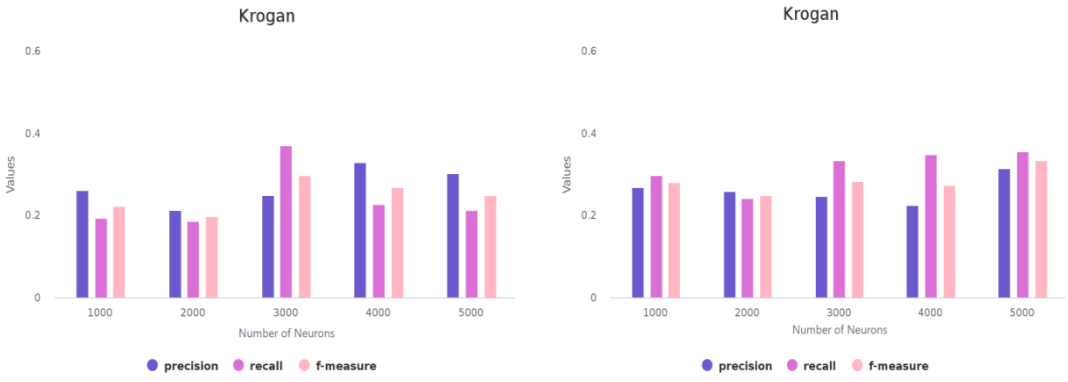


(c) fifteen hide levels

(d) twenty hide levels

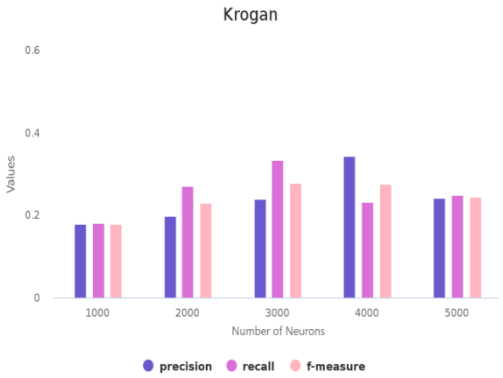
(e) twenty-five hide levels

Fig 2 a-e Results with different hyper parameter on DIP

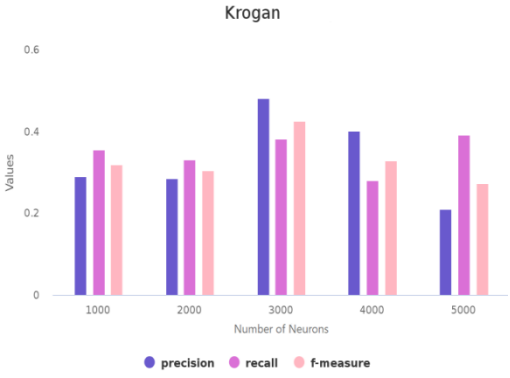


(a) five hide levels

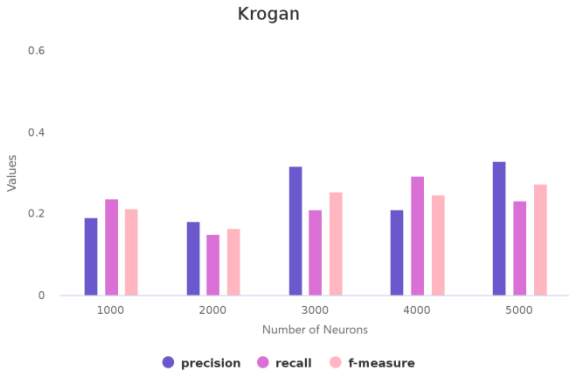
(b) ten hide levels



(c) fifteen hide levels

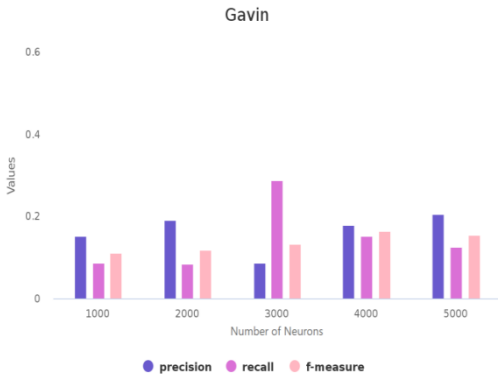


(d) twenty hide levels

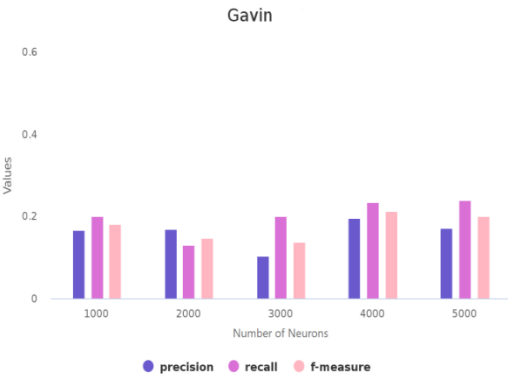


(e) twenty-five hide levels

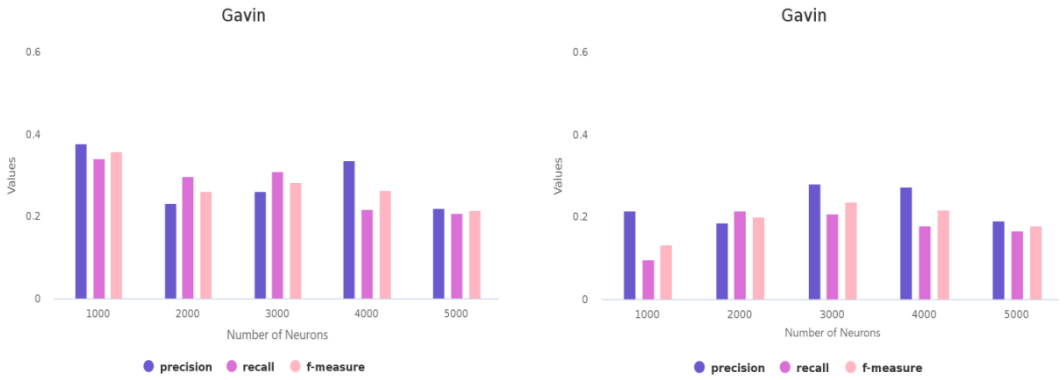
Fig 3 a-e Results with different hyper parameter on Krogan



(a) five hide levels

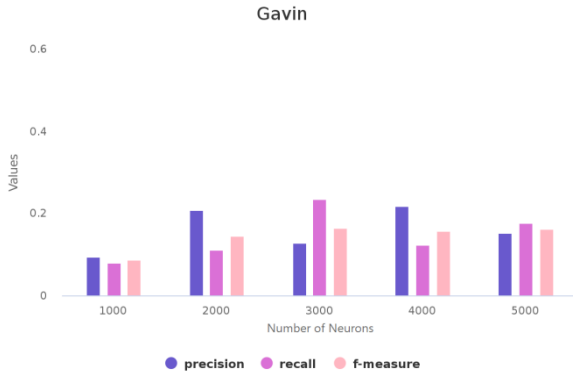


(b) ten hide levels



(c) fifteen hide levels

(d) twenty hide levels



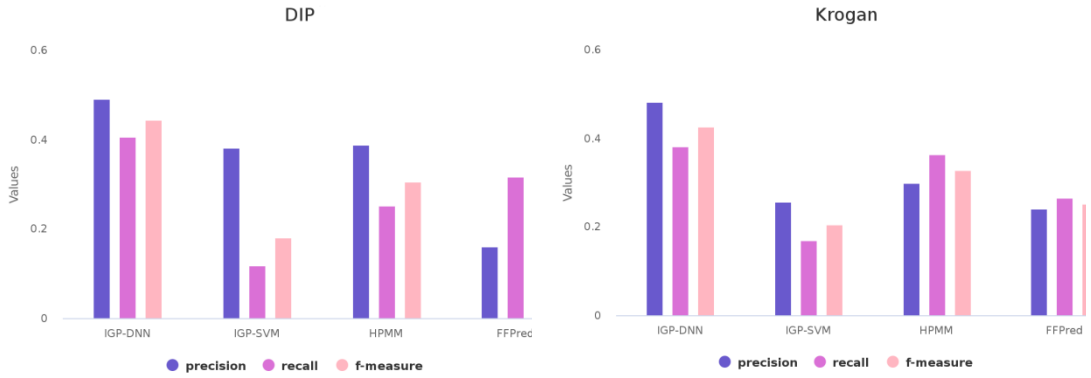
(e) twenty-five hide levels

Fig 4 a-e Results with different hyper parameter on Gavin

Figure 2-4 shows the results of the IGP-DNN model that the dimension reduction approach is KPCA. As shown in Figure 2, 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 3, 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 4, 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 hide level, the model with the multiple

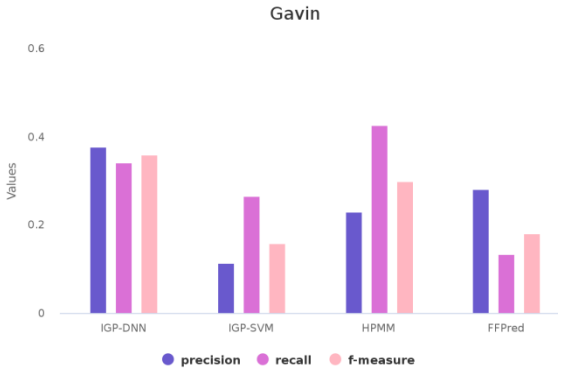
hide levels significantly improves the precision.

As shown in Figure 5, the IGP-DNN model that is chosen the best hyper parameters is compared with the DNN model IGP-SVM, HPMM and FFPred on three data sets, respectively.



(a) Four DNN model on DIP

(b) Four DNN model on Krogan



(c) Four DNN model on Gavin

Fig 5 a-c Performance comparison of the protein function prediction approaches on three data sets

It can be seen from Figure 5 that the Recall of IGP-DNN is slightly lower than HPMM on Gavin, but it has more advantage on the other two data sets. The *Precision* and *F-measure* of IGP-DNN are higher than HPMM on the three data sets. 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. *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 data set. The performance of IGP-DNN model is obviously better than FFPred and more stable. In summary, IGP-DNN has more advantages than IGP-SVM, HPMM and FFPred when predicting unknown protein function. IGP-DNN combines the advantages of IFCM-GOA and DNN. The combination of IFCM and GOA not only avoids falling into local optimal when extracting function module feature and reduces the over-sensitivity of IFCM for clustering center, but also improves the precision of the protein function module feature extraction. The integration of multiple protein properties, including protein family, structure domain and binding site, and protein function module feature 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.

5. CONCLUSION

This paper proposes a protein function prediction approach based on DNN. In the model, protein features are composed of the protein function module features that are extracted by using IFCM-GOA and the protein property features that are reduced dimensions by using KPCA to address the noise sensitivity and the other problems during predicting protein function. In addition, the 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 IGP-SVM, HPMM and FFPred on three different data sets of the yeast PPI network. The experimental results demonstrate that the *Precision*, *Recall*, *F-measure* of IGP-DNN are

better than IGP-SVM, HPMM and FFPred and IGP-DNN can effectively the unknown predict protein function.

Compliance with Ethical Standards

Conflicts of Interest:

We declare that we have no proprietary, financial, professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, "Protein Function Prediction with Deep Neural Learning".

Corresponding author: Lichuan Gu

Supplementary information

Acknowledgments

Not applicable.

Funding

This work is partially supported by the National Natural Science Foundation of China under Grant (31771679, 31671589), the Anhui Foundation for Science and Technology Major Project, China, under Grant (201903a06020009, 18030901034, 201904e01020006), the Key Laboratory of Agricultural Electronic Commerce, Ministry of Agriculture of China under Grant (AEC2018003, AEC2018006), the 2019 Anhui University Collaborative Innovation Project (SN: GXXT-2019-013), the Hefei Major Research Project of Key Technology (J2018G14).

Abbreviations

DNN: Deep neural network; GO: Gene ontology; KPCA: Kernel principal components analysis; GOA: Grasshopper optimization algorithm; PPI: Protein-protein interaction; IFCM: Intuitionistic fuzzy c-means; ANNs: Artificial neural networks; KICA: Kernel

independent principal component analysis; LLE: Local linear embedding; DIP: Database of interacting protein; SVM: Support vector machine.

Availability of data and materials

The DIP is freely available from https://dip.doe-mbi.ucla.edu/dip/Main.cgi.

Ethics approval and consent to participate

No ethics approval and consent were required for the study.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Authors' contributions

Zhao Z and Yang N designed and studied the experiments, Zhang H and Hu M analyzed and interpreted the data, Wang H edited the manuscript, Jiao J and Wang C revised and examined the manuscript, Gu L approved the final version of the manuscript. All authors reviewed and approved the final manuscript.

Authors' information

[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.

References

[1] Gu L, Han Y, Wang C, et al. Module overlapping structure detection in PPI using an improved link similarity-based Markov clustering algorithm[J]. Neural Computing and Applications, 2019, 31(5): 1481-1490

[2] Cao R, Freitas C, Chan L, et al. ProLanGO: protein function prediction using neural machine translation based on a recurrent neural network[J]. Molecules, 2017, 22(10): 1732

[3] Szalkai B, Grolmusz V. SECLAF: a webserver and deep neural network design tool for hierarchical biological sequence classification[J]. Bioinformatics, 2018, 34(14): 2487-2489

[4] Tavanaei A, Maida A S, Kaniymattam A, et al. Towards recognition of protein function based on its structure using deep convolutional networks[C]//2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2016: 145-149

[5] Gligorijević V, Barot M, Bonneau R. deepNF: deep network fusion for protein function prediction[J]. Bioinformatics, 2018, 34(22): 3873-3881

[6] Fa R, Cozzetto D, Wan C, et al. Predicting human protein function with multi-task deep neural networks[J]. PloS one, 2018, 13(6): e0198216

[7] Zou X, Wang G, Yu G. Protein function prediction using deep restricted Boltzmann machines[J]. BioMed research international, 2017, 2017

[8] Rifaioglu A S, Doğan T, Martin M J, et al. DEEPred: automated protein function prediction with multi-task feed-forward deep neural networks[J]. Scientific reports, 2019, 9(1): 1-16

[9] Zhang C J, Tang H, Li W C, et al. iOri-Human: identify human origin of replication by incorporating dinucleotide physicochemical properties into pseudo nucleotide composition[J]. Oncotarget, 2016, 7(43): 69783

[10] Pan Y, Liu D, Deng L. Accurate prediction of functional effects for variants by combining gradient tree boosting with optimal neighborhood properties[J]. PloS one, 2017, 12(6): e0179314

[11] Liu Y, Shen S, Fang H, Chen K.An overview of protein function prediction methods[J]. Chinese Journal of Bioinformatics, 2013, 11(01):33-38

[12] Yunes J M, Babbitt P C. Effusion: prediction of protein function from sequence similarity networks[J]. Bioinformatics, 2019, 35(3): 442-451

[13] Saha S, Prasad A, Chatterjee P, et al. Protein function prediction from dynamic protein interaction network using gene expression data[J]. Journal of Bioinformatics and Computational Biology, 2019, 17(04): 1950025

[14] Hoffmann B, Zaslavskiy M, Vert J P, et al. A new protein binding pocket similarity measure based on comparison of clouds of atoms in 3D: application to ligand prediction[J]. BMC bioinformatics, 2010, 11(1): 99

[15] Yang A, Li R, Zhu W, et al. A novel method for protein function prediction based on sequence numerical features[J]. Match-Communications in Mathematical and Computer Chemistry, 2012, 67(3): 833

[16] Deng L, Hinton G, Kingsbury B. New types of deep neural network learning for speech recognition and related applications: An overview[C]//2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013: 8599-8603

[17] Angermueller C, Pärnamaa T, Parts L, et al. Deep learning for computational biology[J]. Molecular systems biology, 2016, 12(7): 878

[18] Min S, Lee B, Yoon S. Deep learning in bioinformatics[J]. Briefings in bioinformatics, 2017, 18(5): 851-869

[19] Cao R, Adhikari B, Bhattacharya D, et al. QAcon: single model quality assessment using protein structural and contact information with machine learning techniques[J]. Bioinformatics, 2017, 33(4): 586-588

[20] Xenarios I, Salwinski L, Duan X J, et al. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions[J]. Nucleic acids research, 2002, 30(1): 303-305

[21] UniProt Consortium. The universal protein resource (UniProt) in 2010[J]. Nucleic acids research, 2010, 38(suppl_1): D142-D148

[22] Ashburner M, Ball C A, Blake J A, et al. Gene ontology: tool for the unification of biology[J]. Nature genetics, 2000, 25(1): 25-29

[23] Pu S, Wong J, Turner B, et al. Up-to-date catalogues of yeast protein complexes[J]. Nucleic acids research, 2009, 37(3): 825-831

[24] Gavin A C, Bösche M, Krause R, et al. Functional organization of the yeast proteome by systematic analysis of protein complexes[J]. Nature, 2002, 415(6868): 141-147

Figures

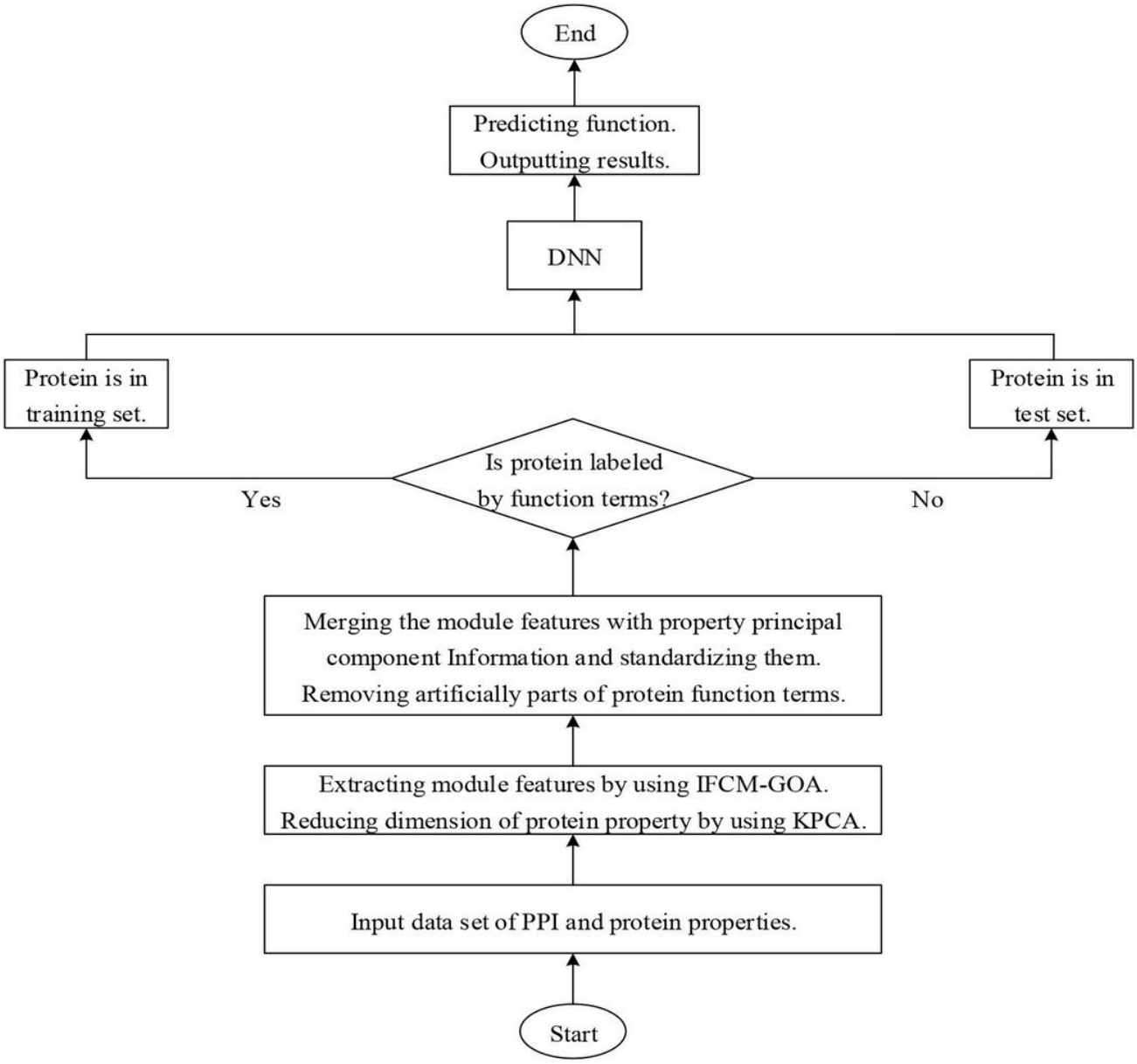


Figure 1

Process of DNN-based Protein function prediction

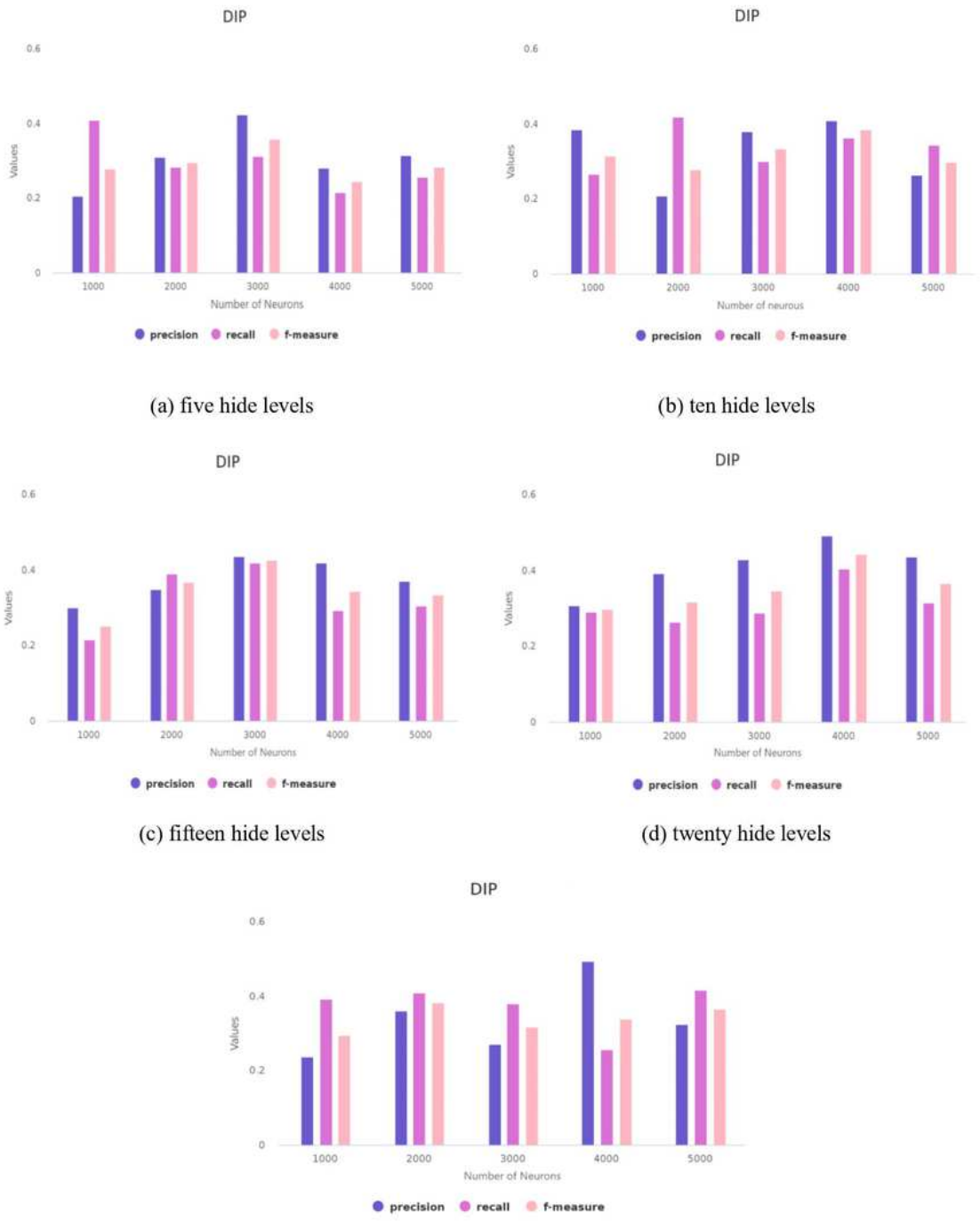


Figure 2

a-e Results with different hyper parameter on DIP

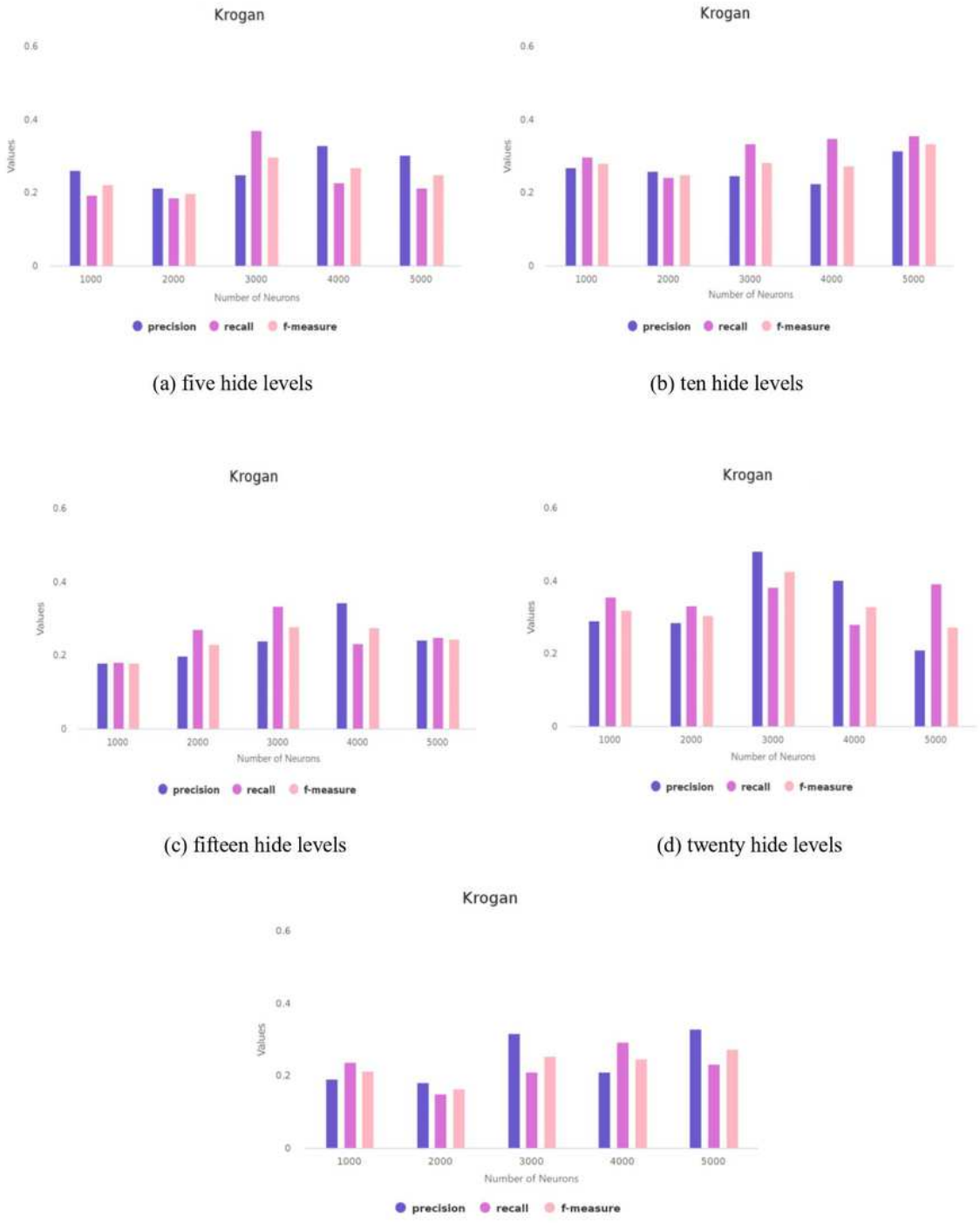


Figure 3

a-e Results with different hyper parameter on Krogan

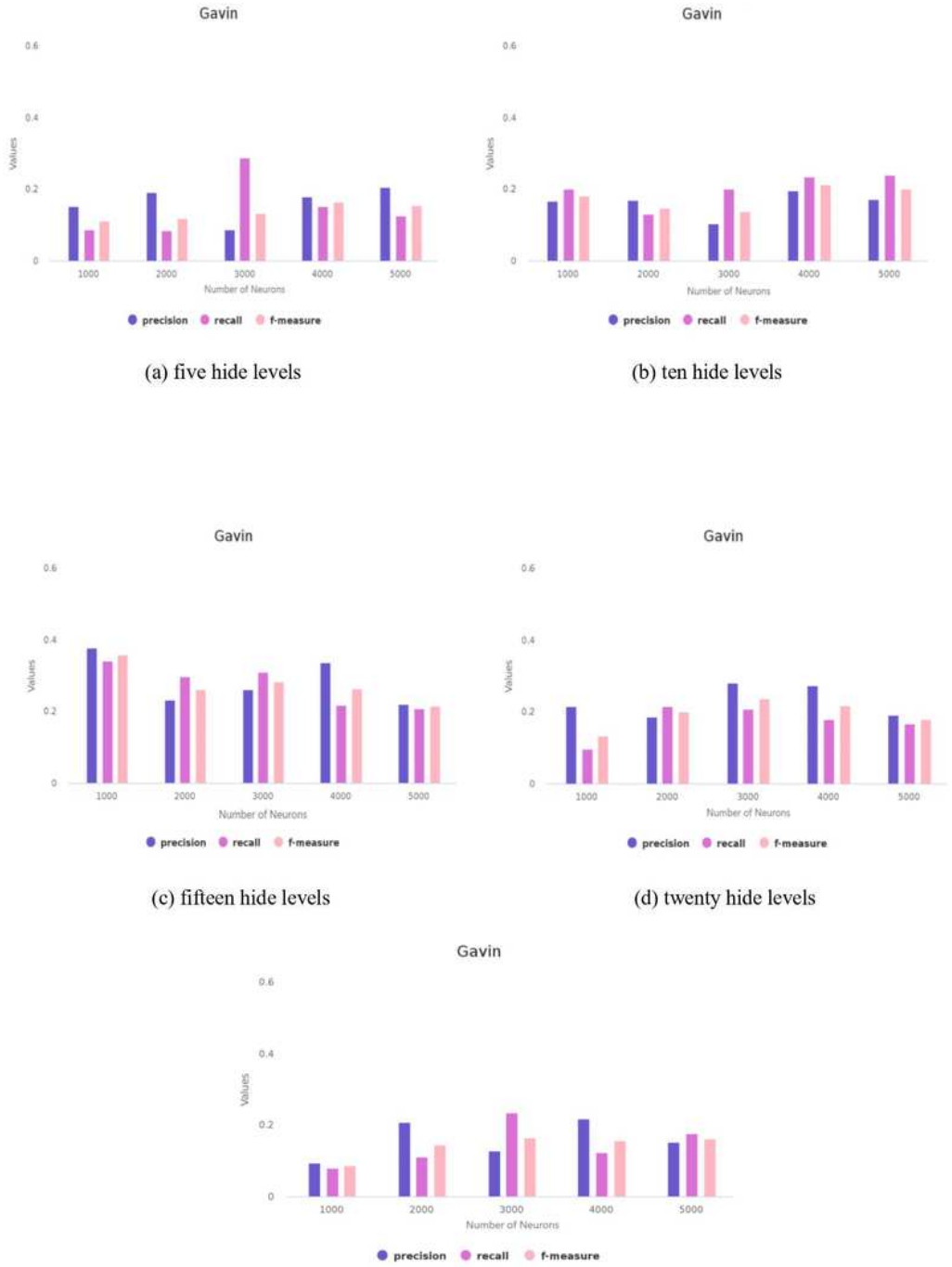
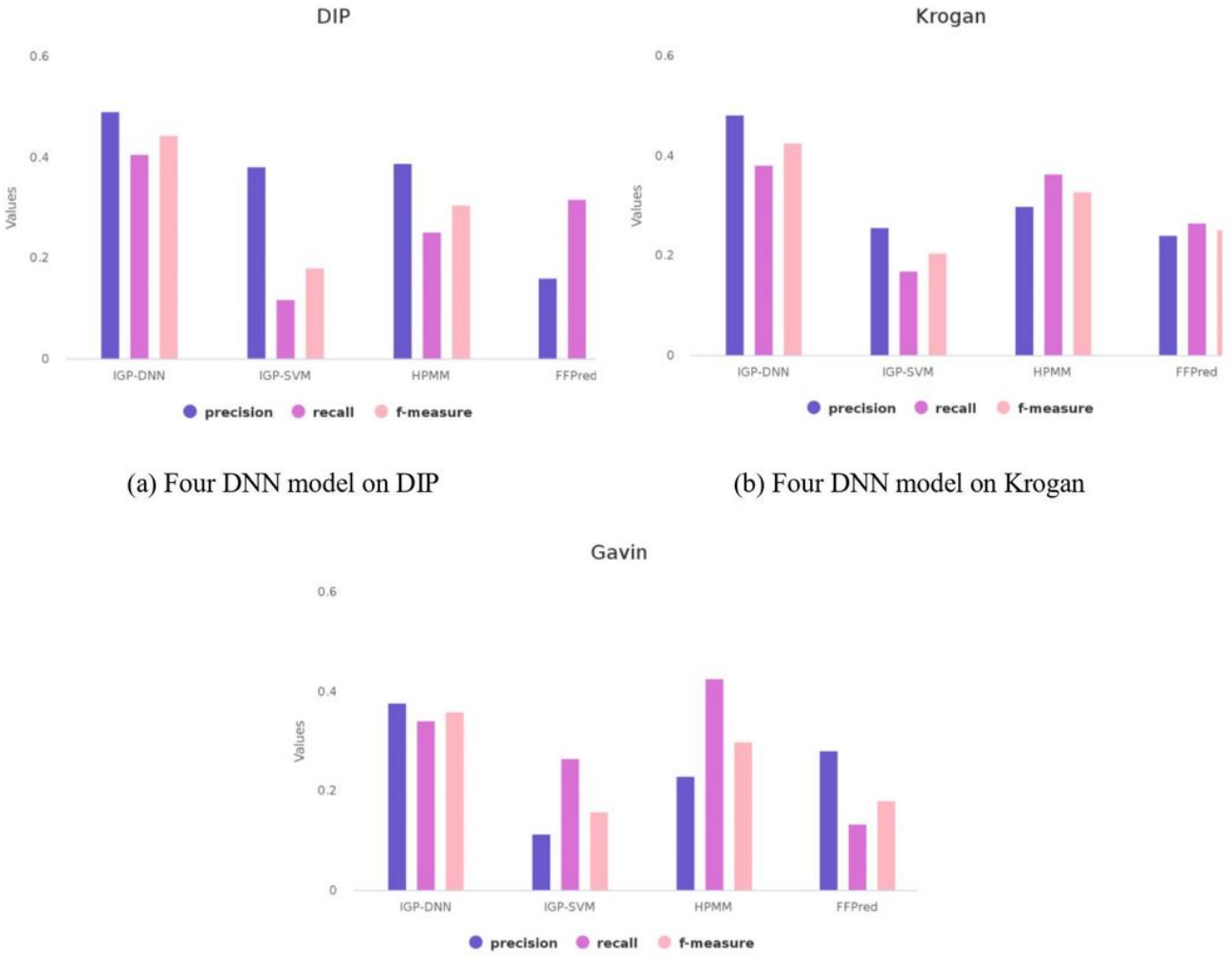


Figure 4

a-e Results with different hyper parameter on Gavin



(a) Four DNN model on DIP

(b) Four DNN model on Krogan

(c) Four DNN model on Gavin

Figure 5

a-c Performance comparison of the protein function prediction approaches on three data sets