frontiers | Frontiers in Pharmacology

TYPE Methods
PUBLISHED 28 October 2022
DOI 10.3389/fphar.2022.1018294

DPB-NBFnet: Using neural Bellman-Ford networks to predict DNA-protein binding

Jing Li[1], Linlin Zhuo[1]*, Xinze Lian[1], Shiyao Pan[1] and Lei Xu[2]*

[1]School of Data Science and Artificial Intelligence, Wenzhou University of Techonology, Wenzhou, China, [2]School of Electronic and Communication Engineering, Shenzhen Polytechnic, Shenzhen, China

Check for updates

OPEN ACCESS

EDITED BY
Pan Zheng,
University of Canterbury, New Zealand

REVIEWED BY
Zongda 吴,
Shaoxing University, China
Liu Mingzhe,
Chengdu University of Technology, China

*CORRESPONDENCE
Linlin Zhuo,
zhuoninnin@163.com
Lei Xu,
csleixu@szpt.edu.cn

SPECIALTY SECTION
This article was submitted to Experimental Pharmacology and Drug Discovery,
a section of the journal
Frontiers in Pharmacology

RECEIVED 13 August 2022
ACCEPTED 28 September 2022
PUBLISHED 28 October 2022

CITATION
Li J, Zhuo L, Lian X, Pan S and Xu L (2022), DPB-NBFnet: Using neural Bellman-Ford networks to predict DNA-protein binding.
Front. Pharmacol. 13:1018294.
doi: 10.3389/fphar.2022.1018294

DNA is a hereditary material that plays an essential role in micro-organisms and almost all other organisms. Meanwhile, proteins are a vital composition and principal undertaker of microbe movement. Therefore, studying the bindings between DNA and proteins is of high significance from the micro-biological point of view. In addition, the binding affinity prediction is beneficial for the study of drug design. However, existing experimental methods to identifying DNA-protein bindings are extremely expensive and time consuming. To solve this problem, many deep learning methods (including graph neural networks) have been developed to predict DNA-protein interactions. Our work possesses the same motivation and we put the latest Neural Bellman-Ford neural networks (NBFnets) into use to build pair representations of DNA and protein to predict the existence of DNA-protein binding (DPB). NBFnet is a graph neural network model that uses the Bellman-Ford algorithms to get pair representations and has been proven to have a state-of-the-art performance when used to solve the link prediction problem. After building the pair representations, we designed a feed-forward neural network structure and got a 2-D vector output as a predicted value of positive or negative samples. We conducted our experiments on 100 datasets from ENCODE datasets. Our experiments indicate that the performance of DPB-NBFnet is competitive when compared with the baseline models. We have also executed parameter tuning with different architectures to explore the structure of our framework.

KEYWORDS

bioinformatics, DNA-protein binding, graph neural networks, neural Bellman-Ford algorithms, DPB-NBFnet, computational biology

1 Introduction

DNA is a hereditary material that plays an essential role in human metabolism and almost all organisms. Meanwhile, proteins are a vital composition and principal undertaker of microbe movement. Therefore, studying the interactions between DNA and proteins is highly significant from the biological point of view because the influence of DNA-binding proteins on a large number of biological processes is conclusive, especially in gene transcription and regulation. However, traditional experimental methods to detect

DNA-protein binding (DPB), such as CHIP-seq or new methods such as ProNA 2020 (Qiu et al., 2020), are extremely expensive and time consuming. To cut the cost, computational biologists have used the deep learning architecture to predict the binary label of sequence-based data, which indicates the relationship of sequences. These learning tasks often have large amount of training examples, which allow scientists to adapt them to deep learning structures, especially graph neural networks (GNNs), without experiencing the over-fitting problem.

Many deep learning architectures have been used to predict DNA-protein binding (Dong et al., 2022). For example, DeepRAM (Trabelsi et al., 2019) first obtained the sequence representation by word2vec embedding, and then used convolutional layers and recurrent layers to process the data. DeepBind built a Long-Short-Term-Memory (LTSM) and Convolutional Neural Network (CNN) structure to model the sequence data (Alipanahi et al., 2015). Hierarchical Attention Networks (HANs) are another kind of architecture that is based on the natural language processing method for document classification (Yu et al., 2019). Trabelsi et al. (2019) demonstrated a comprehensive evaluation of deep learning architectures, including CNN (Zeiler and Fergus, 2014) and Recurrent Neural Networks (RNNs; Medsker and Jain, 2001), to predict DNA/RNA binding specificities.

Computational biologists have recently built GNNs to predict DNA- or RNA-protein interactions. For example, Guo et al. (2021) developed a method called DNA-GCN, which utilized a graph CNN architecture to first build a large graph containing the neighborhood information and then turn this problem into a node classification task. In another example, Shen et al. (2021) developed a method called NPI-GNN (which is composed of GraphSAGE, top-k pooling, and global pooling layers) to predict non-coding RNA (ncRNA)-protein interactions. Other work related with predicting molecular interactions has also built a knowledge graph and then utilized a GNN model (Song et al., 2022; Zeng et al., 2022).

Despite these studies, there is still a gap in the research of predicting DNA-protein binding with GNNs. In general, predicting DNA-protein binding could be regarded as a link prediction problem on a graph. In this graph, we regard different DNA and proteins as vertices with different attributes. After building the graph, we transfer predicting DNA and protein binding into predicting the edge existence of different vertices, which is a link prediction problem in a homogeneous graph.

We now wonder if we could combine the advanced methodology for link prediction problem with our real biological data. In our work, we propose a novel GNN method, which is called DPB-NBFnet and is based on Neural Bellman-Ford neural networks (NBFnets) (Zhu et al., 2021). The NBFnet is a novel GNN architecture that was developed by Zhu for the link prediction problem, which unified the link prediction methods in both heterogeneous and homogeneous graphs. NBFnet includes three neural functions—IND (INDICATOR), MES (MESSAGE), and AGG (AGGREGATE) functions—and it has been proved to have a state-of-the-art performance when compared to other methods for the link prediction problem, including GraIL (Teru et al., 2020). We evaluated our model on the 100 chosen datasets from ENCODE. Compared with other GNN models for the link prediction problem, the final accuracy and time consumption of our DNA-NBFnet framework for prediction of DNA-protein binding have been shown to be superior. We believe that our work could make some contributions to the study of DNA-protein binding and will also inspire other computational biological models.

2 Related work

2.1 Homogeneous and heterogeneous graphs

Given the real sequence data of DNA and protein, we first obtain a graph in which each vertex represents a DNA or a protein, and each edge represents an interaction or binding between DNAs and proteins. In this way, we could keep the topological structure of initial data (Cai et al., 2020b). Generally, a graph (West et al., 2001) is denoted as an ordered triplet $(\mathcal{V}, \mathcal{E}, \mathcal{R})$. This triplet is composed of a nonempty set $\mathcal{V}$ representing vertices, a set $\mathcal{E}$ representing edges, and $\mathcal{R}$ representing the relation types. Moreover, we use $\mathcal{N}(u)$ to denote the set of neighborhood nodes of node u, $\mathcal{E}(u)$ to denote the set of edges whose endpoint is u. If there are various kinds of nodes or edges, then this graph is categorized as a heterogeneous graph, which is represented as $(\mathcal{V}, \mathcal{E}, \mathcal{R})$, otherwise it is categorized a homogeneous graph $(\mathcal{V}, \mathcal{E})$. In our study, we build a homogeneous graph where we initialize DNA and protein as the same kind of vertices but with different labels in actual node representations. After building our graph, we turn predicting the DNA-protein binding into a link prediction problem between these nodes. Our goal is to predict the edge existence between different vertices of this graph.

2.2 Different methods for link prediction problems

There are currently three methods to solve this link prediction problem in our graph, which are path-based methods, embedding methods, and GNNs (Liu et al., 2022a; Liu et al., 2022b; Peng et al., 2022). Among these methods, GNNs are a growing family of methods and have shown the most advanced performance. GNN models are a set of representation learning functions. The highlight of GNNs is that they have the ability to encode topological structures of graphs (Cai et al., 2020a; Wu et al., 2020a; Fu et al., 2020). There are several kinds of GNNs for DNA-protein link prediction problem. DNA Graph

Convolution Networks (DNA-GCN) encapsulate each node's hidden representation by aggregating the feature information from its neighbors (Guo et al., 2021). DNA Graph Attention Networks (DNA-GANs) adopt attention mechanism to aggregating feature information and concatenate the outputs of multiple models (Xiao et al., 2017). However, these methods require the information of global graph and can only be used in transductive learning. Variational Graph Auto-encoders (VGAE) learns a graph embedding to get node embeddings for all nodes, and then aggregates the embeddings of the source and target nodes as their link representation (Kipf and Welling, 2016). These frameworks encode node representations by different GNN models and decode edges as functions over node pairs. SEAL is another mainstream framework, which has an end-to-end structure that encodes the enclosing subgraphs of each node explicitly. However, these structures require a subgraph to be created for each node, resulting in a high cost for large graphs, especially in DNA-protein binding graphs. In comparison with these methods, our DNA-NBFnet encapsulates the paths between two vertices at a relatively low cost.

2.3 Path formulation for DNA-protein binding prediction

The goal of link prediction is to predict the existence of a relationship r between two vertices u and v. For DNA-protein binding, the vertices could be proteins or DNA and the relationship is their binding. To capture the paths between two nodes, we regard it as a message passing process, and its power for link prediction has been proven by previous work. For example, PageCon considers different edge features without node difference in the graph and then passes relational messages among edges iteratively to aggregate neighborhood information (Wang et al., 2021). For a given entity pair u and v, PageCon models neighborhood topology including relational context and relational paths, and combines them for link prediction. In another example, MPNN learns representations on graph data by a message passing algorithm and executes an aggregation procedure to compute a function of their entire input graph (Gilmer et al., 2020). Based on these inspirations, our method combines the message passing heuristics and Bellman-Ford algorithm to efficiently solve the link prediction problem.

In our case, we only consider one kind of relation (i.e., if a DNA-protein binding exists). This requires a pair representation $\mathbf{h}\,(u, v)$ to be learned. Inspired by the message passing methods, this algorithm should pass relational messages among edges iteratively and finally aggregate neighborhood information. This pair representation should capture the local topological structure between the nodes u and v. Traditional methods, such as the Katz index (Katz, 1953) and PageRank (Page et al., 1999), encode such structure by counting various kinds of random walks from u to v. Based on this intuition, the pair representation $\mathbf{h}\,(u, v)$ is formulated as a generalized sum of path representations between these two nodes, with ⊕ being a summation calculator. Each path representation is formulated as a generalized product of edge representations in the path between them with the multiplication operator ⊗.

$$\mathbf{h}(u,v) = \mathbf{h}(P_1) \oplus \mathbf{h}(P_2) \oplus \cdots \oplus \mathbf{h}\left(P_{|\mathcal{P}_{uv}|}\right) := \bigoplus_{i=1}^{\mathcal{P}_{uv}} \mathbf{h}(P_i) \tag{1}$$

$$\mathbf{h}\left(P_i = \left(e_1, e_2, \ldots, e_{|P_i|}\right)\right) = \mathbf{w}(e_1) \otimes \mathbf{w}(e_2) \otimes \cdots \otimes \mathbf{w}\left(e_{|P_i|}\right) := \bigotimes_{j=1}^{|P_i|} \mathbf{w}\left(e_j\right) \tag{2}$$

where $\mathcal{P}_{uv}$ represents the set of paths from u to v, $\mathbf{h}(P_i)$, $(i = 1, 2, \ldots, P_{|\mathcal{P}_{uv}}|)$ represents the numbered path representation from u to v, and each path P_i is composed of several edges $e_1, e_2, \ldots, e_{|P_i|}$ $\mathbf{w}\,(e_j)$ denotes the representation of j-th edge e_j on the path $\mathbf{h}\,(P_i)$. In our formulation, there are two operators, a multiplication operator ⊗ and a summation operator ⊕. We need ⊕ to be commutative, but ⊗ not because it is defined to compute the exact order of the product.

The path formulation could be interpreted explicitly as follows. We first search all possible paths from u to v. We then compute the path representations by multiplication (Equation 2). Finally, we aggregate the path representations as the final pair representation (Equation 1). This path formulation is able to model several traditional link prediction methods, including PageRank and Katz index, which has been proven (Zhu et al., 2021).

2.4 Generalized Bellman-Ford algorithm

As shown in Equations 1, 2, the number of paths increases exponentially as the length of the path grows. Therefore, the computational cost grows drastically. Here, a flexible solution is provided using the generalized Bellman-Ford algorithm. Assuming that the multiplication operator ⊗ and summation operator ⊗ satisfy the semi-ring system (Rowen, 2012), with multiplication identity *{1}*○ and summation identity *{0}*○ respectively, we have the following algorithm, which is called the generalized Bellman-Ford algorithm.

$$\mathbf{h}^{(0)}(u,v) \leftarrow \mathbf{1}_{(u=v)} \tag{3}$$

$$\mathbf{h}^{(t)}(u,v) \leftarrow \left(\bigoplus_{(x,v)\in\varepsilon(v)} \mathbf{h}^{(t-1)}(u,x) \otimes \mathbf{w}(x,v)\right) \oplus \mathbf{h}^{(0)}(u,v) \tag{4}$$

where $\mathbf{1}_{(u=v)}$ is an indicator function. We define it to be equal to *{1}*○ if $u = v$ and *{0}*○ in other cases. $\mathbf{w}\,(x, v)$ is the representation for edge $e = (x, v)$. Equation 3 is also called the boundary condition and Equation 4 is called the Bellman-Ford iteration. It has been proven that this algorithm can solve the traditional algorithm, including graph distance, Katz index, widest path and most reliable path algorithms, personalized PageRank with different multiplication, and summation operators (Zhu et al., 2021).

In summary, this algorithm is able to obtain a pair representation $\mathbf{h}$ (u, v) for a given node u and all $v \in \mathcal{V}$. This method reduces the computational costs by the distributive property of multiplication over summation. Because u and r are fixed, we can abbreviate $\mathbf{h}^{(t)}(u, v)$ as $\mathbf{h}_v^{(t)}$. Finally, we get a source-specific pair representation $\mathbf{h}_v^{(t)}$.

3 Methodology

3.1 Data representations

We collected 100 datasets from ENCODE datasets. ENCODE is an encyclopedia of DNA datasets, which contains about 503,038 datasets in total. From an economical and practical view, we randomly chose 100 datasets among them. Each dataset is related to one specific DNA-binding protein like regulation factor or transcription factor. For each protein, amino acids are divided into seven groups. The datasets contain both positive and negative samples, among which the positive ones were DNA sequences that were experimentally verified and confirmed to bind to this protein. The negative samples were generated by corrupting one of the entities of these positive samples. After collecting the datasets, we get the graph data representation using large matrix M, where M contains the vertex information, topological information and true label l_i (being positive or negative) for the i-th sample.

3.2 Neural parametrization

Given the source node u and the number of setup layers T, the neural Bellman-Ford networks output the pair representation $\mathbf{h}$ (u, v). We parameterize the generalized Bellman-Ford algorithm with three neural functions—which are called IND function, MES function and AGG function here—and we get the following NBFnet algorithm:

Algorithm 1 NBFnet-Neural Parametrization

```
Input: head entity $u$ ,number of layers $L$
Output: pair representations $\mathbf{h}(u, v)$ for all $v \in \mathcal{V}$
for $v \in \mathcal{V}$ do
    $\mathbf{h}^{(0)}(u, v)$ by Equation 5
end for
for $t \leftarrow 1$ to $L$ do
    for $v \in \mathcal{V}$ do
        $\mathcal{M}_v^t \leftarrow \{\mathbf{h}^{(0)}(u, v)\}$
        for $(x, v) \in \mathcal{E}(v)$ do
            $\mathbf{m}(x, v)^{(t)} \leftarrow \mathbf{MES}^{(t)}(\mathbf{h}_x^{(t-1)}, \mathbf{w}(x, v))$
            $\mathcal{M}_v^t \leftarrow \mathcal{M}_v^t \cup \{\mathbf{m}^t(x, v)\}$
        end for
        $h_v^t \leftarrow \mathbf{AGG}^t(\mathcal{M}_v^t)$
    end for
end for
return $\mathbf{h}_v^L$ as $\mathbf{h}(u, v)$ for all $v \in \mathcal{V}$
```

By substituting the neural functions **IND**, **MES**, **AGG** for normal functions in Equations 3, 4, we get the final formulas of NBFnet:

$$\mathbf{h}^{(0)}(u, v) \leftarrow \mathbf{IND}(u, v) \tag{5}$$

$$\mathbf{h}^{(t)}(u, v) \leftarrow \mathbf{AGG}\Big(\Big\{\mathbf{MES}\Big(\mathbf{h}_x^{(t-1)}, \mathbf{w}(x, v)\Big)\Big|(x, v) \in \mathcal{E}(v)\Big\} \bigcup \Big\{\mathbf{h}_v^{(0)}\Big\}\Big) \tag{6}$$

In general, NBFnet could be regarded as a kind of GNN structure for learning pair representations. These neural functions (i.e., IND, MES, AGG functions) remain to be learned.

3.3 SortPooling

After obtaining the pair representations $\mathbf{h}$ (u, v) (given the head entity u) *via* our NBFnet, we process the data with a feed-forward neural network $f(\cdot)$. This network is first is built by a leaky rectified linear unit (ReLU) layer and then a SortPooling layer. The SortPooling layer, unlike simple global pooling layers (Zhang M. et al., 2018), is able to cut down the size of the graph in a flexible and smart manner and effectively extract features. In this pooling layer, the input is sorted by WL colors and it imposes the output as consistent ordering graph vertices. Assuming that the input of these layers $\tilde{\mathbf{h}}(u, v)$ is an $N \times d$ dimensional matrix, the output of the SortPooling layer is a $K \times d$ dimensional matrix. Here, K is a self-defined integer. SortPooling layer output the most refined k vertices by WL colors (Shervashidze et al., 2011; Wu et al., 2021b,a).

3.4 AlphaMEX: Global pooling

Unlike using a normal global pooling layer using the average function or maximum function to replace the whole layer, AlphaMEX is an end-to-end global pooling operator where a nonlinear log-mean-exp function is set up to more effectively process features (Zhang B. et al., 2018; Wu et al., 2020b; Qi et al., 2022). In DPB-NBFnet, we use the AlphaMEX to cut down the size of the graph because the whole network structure contains more layers. The AlphaMEX introduces a parameter α, whose function formula is designed as follows:

$$\mathbf{AlphaMEX}_\alpha\{\mathbf{h}_i\} := \frac{1}{log\left(\frac{\alpha}{1-\alpha}\right)} log\left(\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\alpha}{1-\alpha}\right)^{\mathbf{h}_i}\right) \tag{7}$$

where $\mathbf{h}_i$ denotes input matrix and n denotes the numbers of $\mathbf{h}_i$. α is limited from 0 to 1, which is a trainable parameter.

Together with NBFnet, the leaky ReLU and Sortpooling layer are consecutively composed of one feature extracting module. In our framework, we set up three feature extracting modules sharing this structure. We then process the data with three global pooling layers, here we implement AlphaMEX, which is a better global pooling

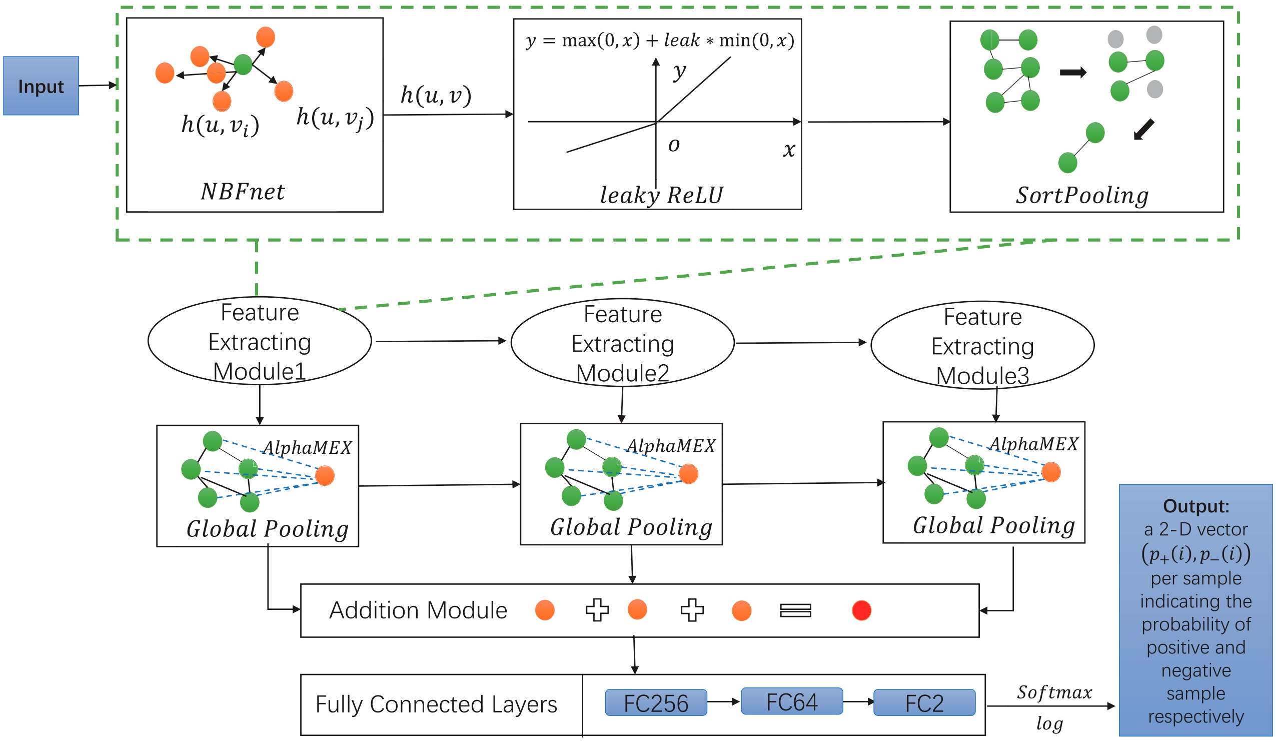
Input
$h(u,v_i)$
$h(u,v_j)$
NBFnet
$h(u,v)$
$y = \max(0,x) + leak * \min(0,x)$
y
o
x
leaky ReLU
SortPooling
Feature Extracting Module1
Feature Extracting Module2
Feature Extracting Module3
AlphaMEX
Global Pooling
AlphaMEX
Global Pooling
AlphaMEX
Global Pooling
Addition Module
Fully Connected Layers
FC256
FC64
FC2
Softmax
log
Output:
a 2-D vector $(p_+(i), p_-(i))$ per sample indicating the probability of positive and negative sample respectively

FIGURE 1
The framework of DPB-NBFnet is composed of three feature processing modules, each of which contains NBFnet, leaky ReLU (set leak as 0.2), and SortPooling. Consecutively, the framework contains three global pooling layers, one addition module, and FC layers (fully connected layers) with 256,64,2 neurons. A homogeneous graph is first input to the feature extracting module. The output is then processed by the following two feature extracting modules. The output of three feature extracting modules is then read by three AlphaMEX global pooling layers, whose outputs are put together by the addition module. The outcome of the addition module is then read by three FC layers with 256, 64, and 2 neurons, respectively. We then use a softmax function then a logarithmic function to give the final predicted value $p_+(i)$ and $p_-(i)$ to classify the positive and negative samples, respectively.

operator for convolutional neural networks. The additive module then sums up the results of the AlphaMEX pooling layers. Finally, the output of the additive module is presented to a network of fully connected layers. The outcome of the last FC layer is processed by a softmax function. We then take the logarithmic value as the final predicted result.

Eventually, we get a 2-D vector $(p_+(i), p_-(i))$ that indicates the probability of the positive sample and negative sample, respectively. The overall workflow of DPB-NBFnet framework is illustrated in Figure 1.

3.5 Loss function

For the i-th DNA-protein pair, if there is a binding in this pair, then we denote it as a positive sample, and otherwise negative. We aim to minimize the negative logarithmic likelihood of positive and negative pairs. Hence, we design and use such a loss function in our model:

$$\mathcal{L} = -\frac{1}{k}\sum_{i=1}^{k}\left[l_i p_+(i) + (1 - l_i) p_-(i)\right] \tag{8}$$

where k is the number of samples in the dataset, and l_i the binary label of the i-th sample.

3.6 Time complexity

DPB-NBFnet has a relative low time complexity compared with other GNN frameworks. We will discuss its time complexity roughly. Assuming that our model needs to infer the likelihood of a dataset containing $|\mathcal{V}|$ samples with d dimensions, where $\mathcal{V}$ is the set of all uncertain positive and negative samples, we need to implement the algorithm (Equations 5, 6) once to get the predictions. The time complexity here is $O(|\mathcal{E}|d)$. After this, a constant K is settled for Equations 5, 6 to converge. So far, it has a time complexity $O(|\mathcal{E}|d + |\mathcal{V}|d^2)$. In summary, for each sample, the average time complexity is $O(\frac{|\mathcal{E}|d}{\mathcal{V}} + |d^2)$.

TABLE 1 Performance comparison of different methods on ENCODE datasets.

Methods	PR[a] (%)	RE[b] (%)	AC[c] (%)	MCC[d]	AUROC[e] (%)
Katz index	81.2	72.5	81.0	74.9%	75.7
node2vec	83.4	76.6	87.2	89.6%	89.1
VGAE	91.0	90.0	92.0	86.8%	94.7
DRUM	91.7	90.5	93.7	83.9%s	96.9
SEAL	91.5	91.3	91.4	82.71%	97.5
DPB-NBFnet	93.7	92.6	97.2	89.1%	98.2

[a] PR stands for precision (Equation 9).
[b] RE stands for recall (Equation 10).
[c] AC stands for accuracy (Equation 11).
[d] MCC stands for Matthew correlation coefficient (Equation 12).
[e] AUROC stands for area under the receiver operating characteristic curve (Hosmer et al., 2013).

TABLE 2 AUROC results of different AGG and MES functions.

AGG\MES	Sum (%)	Mean (%)	Max (%)
DistMult	85.8	89.3	90.7
TransE	89.2	74.7	93.6
RotatE	94.6	96.5	98.4

3.7 Measurements and evaluation

In our work, we utilize the average precision (AP), the average recall (AR), the average accuracy (AC), the Matthew's correlation coefficient (MCC; Chicco and Jurman, 2020), and the area under the receiver operating characteristic curve (AUROC; Hosmer et al., 2013) to measure the performance of our DPB-NBFnet modules. These statistics are defined as follows:

$$\mathbf{PR} = \frac{TrP}{TrP + FaP} \tag{9}$$

$$\mathbf{RE} = \frac{TrP}{TrP + FaN} \tag{10}$$

$$\mathbf{AC} = \frac{TrP + TrN}{TrP + TrN + FaP + FaN} \tag{11}$$

$$\mathbf{MCC} = \frac{TrP \cdot TrNFaP \cdot FaN}{\sqrt{(TrP + FaP)(TrP + FaN)(FaN + FaP)(TrN + FaN)}} \tag{12}$$

Where TrP, TrN, FaP, FaN denotes the number of true positive samples, true negative samples, false positive samples, and false negative samples, respectively. The AUROC is calculated by its geometric meaning by Python, namely the area under the ROC curve (Wu et al., 2021c; Su et al., 2022).

4 Experiment

4.1 Experiment setup

We collected 100 datasets from ENCODE datasets. Each of these datasets corresponds to a specific DNA-binding protein like transcription factor or regulation factor. The positive samples are DNA sequences that were experimentally confirmed to bind to this protein, and the negative samples were generated by corrupting one of the entities of these positive samples.

After preparing the data, we implement the DPB-NBFnet by Python, with the main packages PyTorch 1.10.0 and PyTorch-Geometric (PyG) 2.0.4. PyG is a library for implementing GNN

TABLE 3 Results of different AGG and MES functions with multiple measurements.

#layers T\Methods	2 (%)	3 (%)	4 (%)	5 (%)	6 (%)	7 (%)	8 (%)
PR[a]	71.2	77.2	80.1	81.5	85.8	96.1	96.3
RE[b]	73.4	79.5	81.4	85.6	90.4	96.8	97.0
AC[c]	75.8	80.4	83.4	85.8	92.1	97.2	97.4
MCC[d]	73.3	82.1	85.4	86.6	91.4	96.9	97.1
AUROC[e]	72.6	81.4	86.4	91.3	94.4	97.6	97.6

[a] PR stands for precision (Equation 9).
[b] RE stands for recall (Equation 10).
[c] AC stands for accuracy (Equation 11).
[d] MCC stands for Matthew correlation coefficient (Equation 12).
[e] AUROC stands for area under the receiver operating characteristic curve (Hosmer et al., 2013).

models on structured data. Given that our NBFnet can be regarded as a GNN model, it is convenient for us to use this package.

4.2 Main results

We evaluated the DNA-protein binding prediction performance of our DPB-NBFnet framework on the 100 ENCODE datasets. We used a four-fold cross-validation strategy. We compared DPB-NBFnet model with other link prediction models, among which we chose a traditional path-based method Katz index, an embedding method node2vec (Grover and Leskovec, 2016), and three GNN models: VGAE (Kipf and Welling, 2016), DRUM (Sadeghian et al., 2019), and SEAL (Zhang and Chen, 2018). Table 1 presents the final prediction results. As can be seen in Table 1, our DPB-NBFnet framework was able to achieve 93.7% precision, 92.6% recall, 95.2% accuracy. Therefore, it has shown a state-of-the-art performance compared with other methods for link prediction problems on DNA-protein binding.

4.3 Exploration of the DPB-NBFnet structure

4.3.1 Neural functions

In general, DPB-NBFnet benefits from advanced embedding methods, such as DistMult (Yang et al., 2017), RotatE (Sun et al., 2019) and TransE (Bordes et al., 2013). Compared with explicit AGG functions (i.e., sum, max, mean), combinations of advanced AGG and MES functions achieve a better performance. Table 2 gives the results of AUROC choosing different MES and AGG functions.

4.3.2 Number of GNN layers

As can be seen in Algorithm 1, parameter L is required as an input, which represents the number of layers. Although some studies have reported that the GNN model usually has a better performance when the layers go deeper, we observed that the DPB-NBFnet does not behave in this way. At first the performance improves as more layers are included but it then reaches saturation at about six layers. We predict that this happen because paths no longer than six are enough for a link prediction problem. The results of AUROC are listed in Table 3.

5 Discussion and conclusion

5.1 Application

Predicting DNA-protein binding has a significant meaning from the micro-biological point of view, but it is extremely expensive to explore all kinds of DNA-binding proteins *via* experimental methods. Our DPB-NBFnet framework was able to achieve a prediction accuracy of 97.2%, which could be applied into predictions of DNA-protein binding on real datasets. This provides biologists with a cost-effective method to explore more DNA-binding proteins and also study their actual functions in micro-organisms.

5.2 Conclusion

In this work, we present a novel framework DPB-NBFnet, which is a GNN model that predicts DNA-protein binding. This framework uses NBFnet, SortPooling, and AlphaMEX, which are all technologies from modern machine-learning research. Our results show that DPB-NBFnet outperformed baseline models. We also explore the influence of different neural functions and number of layers on our DPB-NBFnet structure. DPB-NBFnet can be considered to be a substitute option to the existing link prediction methods on real datasets. We also hope that this method could inspire more computational biologists and could be put into use in diverse kinds of tasks in the future.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, and further inquiries can be directed to the corresponding authors.

Author contributions

Conception and design of study: JL, LZ, XL, and LX; acquisition of data: JL, LZ, and SP; analysis and/or interpretation of data: JL, XL, and LZ; drafting the manuscript: JL; revising the manuscript critically for important intellectual content: XL and LX; approval of the version of the manuscript to be published: JL, LZ, LX, XL, and SP.

Acknowledgments

We would like to thank Huimin Huang for discussion on DNA-binding proteins, and Rouyi Zhang and Huizuo Xu for discussion on the message passing algorithms.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nat. Biotechnol.* 33, 831–838. doi:10.1038/nbt.3300

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Adv. neural Inf. Process. Syst.* 26.

Cai, L., Ren, X., Fu, X., Peng, L., and Zeng, X. (2020a). ienhancer-xg: Interpretable sequence-based enhancers and their strength predictor. *Bioinformatics* 37, 1060–1067. doi:10.1093/bioinformatics/btaa914

Cai, L., Wang, L., Fu, X., Xia, C., Zeng, X., and Zou, Q. (2020b). ITP-pred: An interpretable method for predicting, therapeutic peptides with fused features low-dimension representation. *Briefings Bioinforma.* 22. doi:10.1093/bib/bbaa367

Chicco, D., and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics* 21, 6–13. doi:10.1186/s12864-019-6413-7

Dong, J., Zhao, M., Liu, Y., Su, Y., and Zeng, X. (2022). Deep learning in retrosynthesis planning: Datasets, models and tools. *Brief. Bioinform.* 23, bbab391. doi:10.1093/bib/bbab391

Fu, X., Cai, L., Zeng, X., and Zou, Q. (2020). Stackcppred: A stacking and pairwise energy content-based prediction of cell-penetrating peptides and their uptake efficiency. *Bioinformatics* 36, 3028–3034. doi:10.1093/bioinformatics/btaa131

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2020). "Message passing neural networks," in *Machine learning meets quantum physics* (Berlin, Germany: Springer), 199–214.

Grover, A., and Leskovec, J. (2016). "node2vec: Scalable feature learning for networks," in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, 855–864. doi:10.1145/2939672.2939754

Guo, Y., Luo, X., Chen, L., and Deng, M. (2021). "Dna-gcn: Graph convolutional networks for predicting dna-protein binding," in *International conference on intelligent computing* (Berlin, Germany: Springer), 458–466.

Hosmer, D. W., Lemeshow, S., and Sturdivant, R. (2013). "Area under the receiver operating characteristic curve," in *Applied logistic regression*. Third ed (Hoboken, NJ, USA: Wiley), 173–182.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika* 18, 39–43. doi:10.1007/bf02289026

Kipf, T. N., and Welling, M. (2016). *Variational graph auto-encoders. arXiv preprint arXiv:1611.07308*.

Liu, W., Lin, H., Huang, L., Peng, L., Tang, T., Zhao, Q., et al. (2022a). Identification of mirna–disease associations via deep forest ensemble learning based on autoencoder. *Brief. Bioinform.* 3, bbac104. doi:10.1093/bib/bbac104

Liu, W., Sun, X., Yang, L., Li, K., Yang, Y., and Fu, X. (2022b). Nscgrn: A network structure control method for gene regulatory network inference. *Brief. Bioinform.* 23, bbac156. doi:10.1093/bib/bbac156

Medsker, L. R., and Jain, L. (2001). Recurrent neural networks. *Des. Appl.* 5, 64–67.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). "The PageRank citation ranking: Bringing order to the web," in *Tech. rep.* (Stanford, CA, USA: Stanford InfoLab).

Peng, L., Yang, C., Huang, L., Chen, X., Fu, X., and Liu, W. (2022). Rnmflp: Predicting circrna–disease associations based on robust nonnegative matrix factorization and label propagation. *Brief. Bioinform.* 23, bbac155. doi:10.1093/bib/bbac155

Qi, A., Zhao, D., Yu, F., Heidari, A. A., Wu, Z., Cai, Z., et al. (2022). Directional mutation and crossover boosted ant colony optimization with application to Covid-19 x-ray image segmentation. *Comput. Biol. Med.* 148, 105810. doi:10.1016/j.compbiomed.2022.105810

Qiu, J., Bernhofer, M., Heinzinger, M., Kemper, S., Rost, B., Melo, F., et al. (2020). Prona2020 predicts protein-dna, protein-rna and protein-protein binding proteins and residues from sequence. *J. Mol. Biol.* 432, 2428–2443. doi:10.1016/j.jmb.2020.02.026

Rowen, L. H. (2012). *Ring theory, 83*. Cambridge, MA, USA: Academic Press.

Sadeghian, A., Armandpour, M., Ding, P., and Wang, D. Z. (2019). Drum: End-to-end differentiable rule mining on knowledge graphs. *Adv. Neural Inf. Process. Syst.* 32.

Shen, Z. A., Luo, T., Zhou, Y. K., Yu, H., and Du, P. F. (2021). NPI-GNN: Predicting ncRNA–protein interactions with deep graph neural networks. *Brief. Bioinform.* 22, bbab051. doi:10.1093/bib/bbab051

Shervashidze, N., Schweitzer, P., Van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.* 12.

Song, B., Luo, X., Luo, X., Liu, Y., Niu, Z., and Zeng, X. (2022). Learning spatial structures of proteins improves protein–protein interaction prediction. *Brief. Bioinform.* 23, bbab558. doi:10.1093/bib/bbab558

Su, H., Zhao, D., Elmannai, H., Heidari, A. A., Bourouis, S., Wu, Z., et al. (2022). Multilevel threshold image segmentation for Covid-19 chest radiography: A framework using horizontal and vertical multiverse optimization. *Comput. Biol. Med.* 146, 105618. doi:10.1016/j.compbiomed.2022.105618

Sun, Z., Deng, Z. H., Nie, J. Y., and Tang, J. (2019). *Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197*.

Teru, K., Denis, E., and Hamilton, W. (2020). "Inductive relation prediction by subgraph reasoning," in International Conference on Machine Learning (PMLR), 9448–9457.

Trabelsi, A., Chaabane, M., and Ben-Hur, A. (2019). Comprehensive evaluation of deep learning architectures for prediction of dna/rna sequence binding specificities. *Bioinformatics* 35, i269–i277. doi:10.1093/bioinformatics/btz339

Wang, H., Ren, H., and Leskovec, J. (2021). "Relational message passing for knowledge graph completion," in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 1697–1707.

West, D. B. (2001). *Introduction to graph theory, vol. 2*. Upper Saddle River, NJ: Prentice hall Upper Saddle River.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. (2020a). A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 4–24. doi:10.1109/TNNLS.2020.2978386

Wu, Z., Shen, S., Li, H., Zhou, H., and Lu, C. (2021a). A basic framework for privacy protection in personalized information retrieval: An effective framework for user privacy protection. *J. Organ. End User Comput. (JOEUC)* 33, 1–26. doi:10.4018/joeuc.292526

Wu, Z., Shen, S., Li, H., Zhou, H., and Zou, D. (2021b). A comprehensive study to the protection of digital library readers' privacy under an untrusted network environment. *Libr. Hi Tech*.

Wu, Z., Shen, S., Lu, C., Li, H., and Su, X. (2020b). How to protect reader lending privacy under a cloud environment: A technical method. *Libr. Hi Tech*.

Wu, Z., Shen, S., Zhou, H., Li, H., Lu, C., and Zou, D. (2021c). An effective approach for the protection of user commodity viewing privacy in e-commerce website. *Knowledge-Based Syst.* 220, 106952. doi:10.1016/j.knosys.2021.106952

Xiao, T., Hong, J., and Ma, J. (2017). *Dna-gan: Learning disentangled representations from multi-attribute images. arXiv preprint arXiv:1711.05415*.

Yang, F., Yang, Z., and Cohen, W. W. (2017). Differentiable learning of logical rules for knowledge base reasoning. *Adv. neural Inf. Process. Syst.* 30.

Yu, W., Yuan, C.-A., Qin, X., Huang, Z.-K., and Shang, L. (2019). "Hierarchical attention network for predicting dna-protein binding sites," in *International conference on intelligent computing* (Berlin, Germany: Springer), 366–373.

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *European conference on computer vision* (Berlin, Germany: Springer), 818–833.

Zeng, X., Tu, X., Liu, Y., Fu, X., and Su, Y. (2022). Toward better drug discovery with knowledge graph. *Curr. Opin. Struct. Biol.* 72, 114–126. doi:10.1016/j.sbi.2021.09.003

Zhang, B., Zhao, Q., Feng, W., and Lyu, S. (2018a). Alphamex: A smarter global pooling method for convolutional neural networks. *Neurocomputing* 321, 36–48. doi:10.1016/j.neucom.2018.07.079

Zhang, M., and Chen, Y. (2018). Link prediction based on graph neural networks. *Adv. neural Inf. Process. Syst.* 31.

Zhang, M., Cui, Z., Neumann, M., and Chen, Y. (2018b). "An end-to-end deep learning architecture for graph classification," in Proceedings of the AAAI conference on artificial intelligence. vol. 32.

Zhu, Z., Zhang, Z., Xhonneux, L. P., and Tang, J. (2021). Neural bellman-ford networks: A general graph neural network framework for link prediction. *Adv. Neural Inf. Process. Syst.* 34, 29476–29490.