

RESEARCH

Open Access



Predicting protein-protein interactions using high-quality non-interacting pairs

Long Zhang¹, Guoxian Yu¹, Maozu Guo^{2,3} and Jun Wang^{1*}

From 29th International Conference on Genome Informatics
Yunnan, China. 3-5 December 2018

Abstract

Background: Identifying protein-protein interactions (PPIs) is of paramount importance for understanding cellular processes. Machine learning-based approaches have been developed to predict PPIs, but the effectiveness of these approaches is unsatisfactory. One major reason is that they randomly choose non-interacting protein pairs (negative samples) or heuristically select non-interacting pairs with low quality.

Results: To boost the effectiveness of predicting PPIs, we propose two novel approaches (NIP-SS and NIP-RW) to generate high quality non-interacting pairs based on sequence similarity and random walk, respectively. Specifically, the known PPIs collected from public databases are used to generate the positive samples. NIP-SS then selects the top-*m* dissimilar protein pairs as negative examples and controls the degree distribution of selected proteins to construct the negative dataset. NIP-RW performs random walk on the PPI network to update the adjacency matrix of the network, and then selects protein pairs not connected in the updated network as negative samples. Next, we use auto covariance (AC) descriptor to encode the feature information of amino acid sequences. After that, we employ deep neural networks (DNNs) to predict PPIs based on extracted features, positive and negative examples. Extensive experiments show that NIP-SS and NIP-RW can generate negative samples with higher quality than existing strategies and thus enable more accurate prediction.

Conclusions: The experimental results prove that negative datasets constructed by NIP-SS and NIP-RW can reduce the bias and have good generalization ability. NIP-SS and NIP-RW can be used as a plugin to boost the effectiveness of PPIs prediction. Codes and datasets are available at <http://mlda.swu.edu.cn/codes.php?name=NIP>.

Keywords: Protein-protein interactions, Non-interacting proteins, Deep neural networks, Sequence similarity, Random walk

Background

As the essential component of all organisms, proteins form the very basis of life and carry out a variety of biological functions within living organisms. A protein rarely accomplishes its functions alone, instead it interacts with other proteins to accomplish biological functions. It is thus generally accepted that protein-protein interactions (PPIs) are responsible for most activities of living organisms. As a hotspot of proteomics research, detecting

PPIs can not only provide great insight for understanding various biological functions in cells, but also contribute to develop drugs for special diseases [1, 2]. In the past decades, different high-throughput technologies had developed to detect PPIs, such as tandem affinity purification (TAP) [3], co-immunoprecipitation (Co-IP) [4], x-ray crystallography [5], yeast two-hybrid (Y2H) screens [6, 7], and mass spectrometric protein complex identification (MS-PCI) [8]. However, these wet-experiment based solutions are costly and tedious. PPIs obtained from these biological experiments only cover a small fraction of the complete PPI network [9]. Furthermore, these high-throughput technologies generally suffer from high rates of false negatives and false positives [9–11].

*Correspondence: kingjun@swu.edu.cn

¹College of Computer and Information Sciences, Southwest University, Chongqing, China

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



Computational approaches have been developed to predict PPIs in an economic and reliable way. These approaches use different data types to predict PPIs, such as protein domains [12], protein structure information [13], gene neighborhood [14], gene fusion [15], and phylogenetic profiles [16, 17]. Nevertheless, these methods are barely achieved if the pre-knowledge of the proteins is not available, i.e., protein functional domains, 3D structure of proteins, and other information [18]. As the explosive growth of sequence data, more and more researchers have moved toward sequence data based approaches to predict PPIs. Experimental results show that it is adequate to predict new PPIs using amino acid sequences alone [19–27].

Martin et al. [19] extracted the feature information of amino acid sequences by the extended signature descriptor and used support vector machine (SVM) to predict PPIs [19]. Shen et al. [20] adopted SVM as the classifier and encoded the feature information of amino acid sequences by conjoint triad (CT), in which the 20 standard amino acids are grouped into 7 categories on the basis of their dipoles and volumes of the side chains. This SVM-based approach yields a high prediction accuracy of 83.9%. However, this approach can not sufficiently encode the feature information, since CT only takes into account the neighboring effect of amino acid sequences, but PPIs usually occur at the non-continuous segments of amino acid sequences. Guo et al. [21] employed the auto covariance (AC) to detect the correlation among discontinuous segments and obtained an accuracy of 86.55%. You et al. [24] combined a novel multi-scale continuous and discontinuous (MCD) feature representation and SVM to predict PPIs. MCD feature representation can adequately capture continuous and discontinuous feature information of segments within an amino acid sequence. This method yields a high accuracy of 91.36% [24]. Different from these SVM-based approaches, Yang et al. [22] combined k NN and local descriptor (LD) to predict PPIs and obtained an accuracy of 83.73%. Du et al. [27] applied deep neural networks (DNNs) and integrated diverse feature descriptors to encode the feature information of amino acid sequences to predict PPIs. This approach obtains a high accuracy of 92.5% on predicting PPIs of *Saccharomyces cerevisiae* [27]. Wang et al. [28] used DNNs and a novel feature descriptor named local conjoint triad descriptor (LCTD), which encodes continuous and discontinuous feature information of local segments within an amino acid sequence, to predict PPIs. This approach yields a high accuracy of 93.12% on PPIs of *Saccharomyces cerevisiae*.

However, the performance of all the aforementioned sequence-based methods heavily depend on the quality of PPIs datasets. Positive examples (interacting protein pairs) are generally chosen based on reliable methods (small scale experiments), interactions confirmed by Y2H

[6, 7], Co-IP [4], and other methods; or interactions confirmed by interacting paralogs [29, 30]. Therefore, given the public protein-protein interactions databases [31], the positive examples are readily available and can be easily constructed. The *difficulty* is that there are no ‘gold standard’ of non-interacting protein pairs (negative examples), which contribute to discriminatively predict PPIs. Two kinds of strategies are widely used by previous computational methods [19–21, 23–27]. The first one randomly pairs proteins and then removes the pairs included in the positive examples [21, 30]. The second constructs negative examples based on the subcellular localization of proteins [23, 25–27]. However, these two strategies have limitations and may compromise the prediction performance. The first strategy wrongly takes a large number of positive samples as negative samples, while the second strategy leads to a biased estimation of PPIs prediction [30].

In this paper, two novel approaches (NIP-SS and NIP-RW) are proposed to improve the performance of PPIs prediction. NIP-SS and NIP-RW separately generate reliable non-interacting pairs (negative dataset) based on sequence similarity and on random walk in the PPIs network. The basic idea of NIP-SS is: given a positive protein pair (i and j), and a protein k , the larger the sequence difference between i and k is, the smaller the probability that k interacts with j (i) is. In addition, we control the degree distribution of selected protein pairs to make it similar as that of the positive dataset. Given a PPI network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of proteins, and \mathcal{E} is the set of weighted undirected edges, where the weight reflects the interaction strength between a protein pair, 1 means an interaction, 0 means unknown. The basic idea of NIP-RW is: after a k -steps random walk on \mathcal{G} , if the edge weight between two proteins is larger than 0, there may be an interaction between them; otherwise, there may be no interaction.

To investigate the effectiveness of NIP-SS and NIP-RW, we firstly collected the positive sets from Database of Interacting Proteins (DIP) [31], and separately constructed negative sets using four strategies : 1) NIP-SS, 2) NIP-RW, 3) subcellular localization, 4) random pairing, and then merged the positive set and each negative set to form a training dataset. Next, we used the auto covariance (AC) [21] descriptor to extract the features from amino acid sequences and Deep neural networks (DNNs) to predict PPIs. AC can account for the interactions between residues with a certain distance apart in the sequence and encode the features by a lower dimensional vector [21], DNNs can automatically extract high-level abstractions and reduce the model training time [32]. We performed comparative and quantitative experiments on public benchmark datasets to study the effectiveness of negative datasets generated by different strategies. The experimental results show that NIP-SS and NIP-RW have

good generalization ability and contribute to a higher accuracy in predicting PPIs than other related and widely-used strategies.

Methods

PPIs datasets

To comprehensively evaluate the rationality of NIP-SS and NIP-RW, we constructed 3 non-redundant positive PPIs sets for *S. cerevisiae*, *H. sapiens*, and *M. musculus* from DIP [31]. Next, we separately generated negative PPIs (non-interacting protein pairs) for these three species using NIP-SS, NIP-RW, subcellular location, and random pairing. After that, we merged the positive and negative sets for each species. As a result, twelve PPIs datasets are obtained. In addition, another six datasets were collected as the independent test datasets to further assess the generalization ability of NIP-SS and NIP-RW, *Mammalian* dataset collected from Negatome 2.0 [33] only contains non-interacting protein pairs, they were generated by manual curation of literature. The procedure of constructing the negative dataset will be introduced later.

The twelve datasets are divided into three groups based on the species. The experimental-validated PPIs of these three groups are all from DIP [31]. The first group contains 17257 positive PPIs of *S. cerevisiae* (version 20160731) collected by Du et al. [27]. The second and third groups are processed by ourselves, they contain 3355 and 923 positive PPIs of *H. sapiens* and *M. musculus* (version 20170205), respectively. These positive PPIs are generated by excluding proteins with fewer than 50 amino acids and with $\geq 40\%$ sequence identity by cluster analysis via the CD-HIT program [34]. The excluded proteins have a heavy impact on the performance of PPIs prediction [21]. Each of these three groups contains four training sets and the difference between these four sets is the negative samples, which are generated by NIP-SS, NIP-RW, subcellular location, and random pairing. Table 1 gives the statistics of these 18 datasets.

Generating non-interacting protein pairs

Negative samples must be chosen with caution, which can heavily affect the performance of PPIs prediction. There are two primary strategies to construct negative samples, including random pairing and subcellular location. For the *first* strategy, after constructing the positive set from DIP, we count the number of proteins in the positive set and put these proteins into set \mathcal{P} . Next, we can randomly select two proteins from \mathcal{P} and take them as a non-interacting pair if they do not have an interaction in the positive set. Obviously, this random pairing is not completely reliable, it will produce a high rate of false negatives for generated negative examples, since the interactions between proteins in the DIP are far from complete.

Table 1 The 18 PPIs datasets used in this paper

Groups	Datasets	# Positive samples	# Negative samples
SC ^a	SC-SS ¹	17257	17257
	SC-RW ²	17257	17257
	SC-Sub ³	17257	17257
	SC-RP ⁴	17257	17257
HS ^b	HS-SS	3355	3355
	HS-RW	3355	3355
	HS-Sub	3355	3355
	HS-RP	3355	3355
MM ^c	MM-SS	923	923
	MM-RW	923	923
	MM-Sub	923	923
	MM-RP	923	923
Test ^d	<i>C. elegans</i>	4013	0
	<i>E. coli</i>	6984	0
	<i>H. sapiens</i>	1412	0
	<i>H. pylori</i>	1420	0
	<i>M. musculus</i>	313	0
	<i>Mammalian</i>	0	1937

^aSC: *S. cerevisiae*;

^bHS: *H. sapiens*;

^cMM: *M. musculus*;

^dTest: Six independent testing datasets;

¹NIP-SS;

²NIP-RW;

³Subcellular location;

⁴Random pairing

The *second* strategy is based on a hypothesis that proteins located in different subcellular localizations do not interact. A protein can be divided into seven groups based on subcellular location information extracted from Swiss-Prot (<http://www.expasy.org/sprot/>), including cytoplasm, nucleus, mitochondrion, endoplasmic reticulum, golgi apparatus, peroxisome and vacuole. The negative samples are obtained by pairing a protein from one group with another protein from the other groups. These negative samples further exclude the proteins pairs appeared in the positive set. However, Ben-Hur and Noble [30] proved that subcellular localization based approaches lead to a biased accuracy of PPIs prediction.

Motivated by the limitations of existing solutions, we proposed two novel approaches NIP-SS and NIP-RW to construct the negative datasets. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ encode a PPIs network, where \mathcal{V} is the set of proteins, and \mathcal{E} stores the known interactions. To construct a reliable negative dataset with good generalization ability, we hope that proteins in the negative dataset are as many as possible. The average repeatability can be employed to describe the generalization ability of a dataset, which is calculated by

$r = \sum_{i=1}^n (d(i) - 1) / n$, where $d(i)$ means the degree of protein i . Note, if a protein in the negative dataset does not ‘interact’ with five proteins, then this protein have a degree of five. The smaller the value of r , the larger the generalization ability of this dataset is. On the one hand, we also hope that the degrees of proteins in the negative dataset are not too small, proteins with low degrees contain little predictive information and are not conducive for predicting PPIs. On the other hand, the degrees of proteins should not be too large, which will lead to an overestimation of prediction results. In addition, the maximum degree of proteins, the proportion of proteins in different ranges of degrees, and the proportion of non-interactions in each range all have an impact on the prediction performance. Given these reasons, we need to construct a reliable negative dataset, in which the degree distribution of proteins and interaction distribution are similar to those in the positive dataset. Such a negative dataset contains more proteins and has less bias.

Generating non-interacting protein pairs based on sequence similarity

The basic idea of NIP-SS is that, for an experimental validated PPI between protein i and j , if a protein k is dissimilar to i , there is a low possibility that k interacts with j . Based on this idea, we firstly generate the positive set of proteins \mathcal{P} having confirmed interactions between another protein, and compute the sequential similarity between any two proteins in \mathcal{P} . Next, we sort the sequence similarity between all protein pairs in \mathcal{P} by the ascending order, and then select the top- m protein pairs with the lowest similarity as negative examples (non-interacting pairs), m is generally larger than the number of positive examples to facilitate the follow-up adjustment. If we employ these negative examples to form a negative dataset and then use this dataset to predict PPIs, it will lead to an over-estimation of PPIs prediction. This is because such negative dataset contains some proteins with very

large degrees, which occur more frequently in the negative dataset than in the positive dataset. For example, the maximum degree in the positive dataset is 252, but 1439 in the initial negative dataset (see “[Contribution of controlling degrees](#)” section). As such, the bias is introduced into the training set composed with positive samples and negative samples. To ensure a good generalization ability, the degree distribution of proteins needs to be controlled during constructing the negative dataset.

We advocate to make the degree distribution of proteins in the negative dataset similar with that of the positive dataset. We firstly calculate the degree distribution of proteins, maximum degree, the proportion of proteins and the number of interactions in different ranges of degrees (such as the degree ≤ 10 , the degree in $(11, 20]$, and so on) in the positive dataset. Similarly, we also compute the above values in the negative dataset. Next, we compare these values of positive and negative datasets, and then adjust the number of non-interacting partners of a protein by referring to the corresponding values of the positive dataset. Finally, we remove the protein pairs appeared in the positive dataset to generate the reliable negative dataset. The process of NIP-SS is shown in Fig. 1.

We collect the amino acid sequences data from the UniProt database [35]. Sequence similarity between two proteins i and j is calculated using blocks substitution matrix (BLOSUM), which is a substitution matrix used for sequence alignment of proteins [36]. BLOSUM matrices are used to score alignments between evolutionary divergent protein sequences. We adopt BLOSUM50 to compute the score between proteins, and then normalize the score as follows:

$$\tilde{bl}(i, j) = \frac{bl(i, j) - \min\{bl(i, 1), \dots, bl(i, n)\}}{\max\{bl(i, 1), \dots, bl(i, n)\}} \quad (1)$$

where n is the total number of proteins in \mathcal{P} , $bl(i, j)$ is the original BLOSUM50 score of protein i and j .

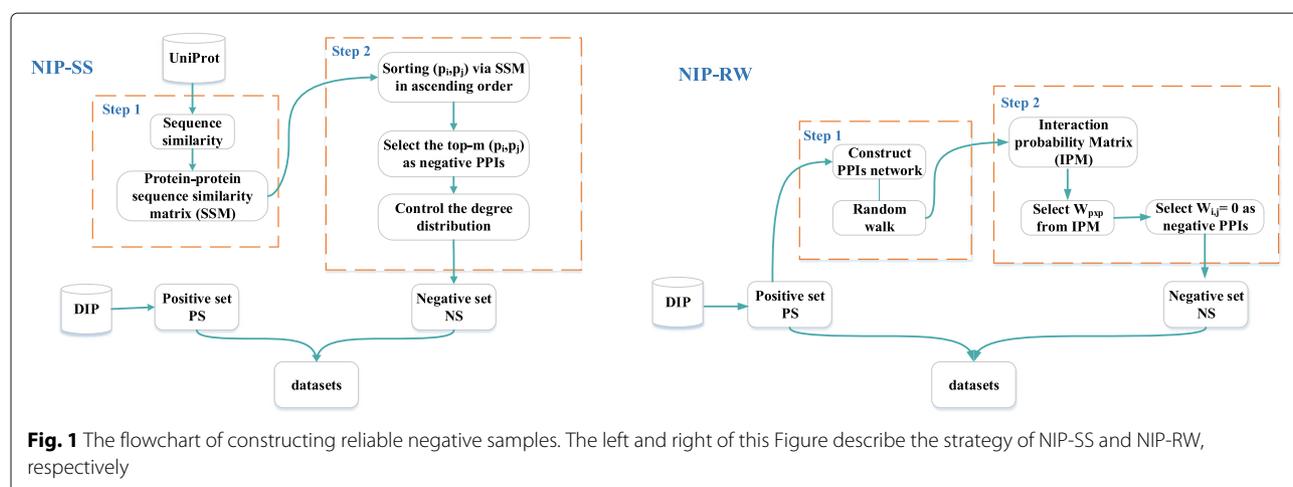


Fig. 1 The flowchart of constructing reliable negative samples. The left and right of this Figure describe the strategy of NIP-SS and NIP-RW, respectively

Generating non-interacting pairs based on random walk

NIP-RW is motivated by the observation that interacting proteins are likely to share similar functions, level-1 ($k = 1$) neighborhood (or directly interacting) proteins are more probable to share functions than level-2 ($k = 2$) neighborhood proteins, whose interactions are mediated by another protein. In other words, the probability of sharing similar functions reduces as the increase of k [37]. Given that, two proteins that can only be connected after a k -step random walk, is less likely to share functions and thus less probable to interact with each other. The flowchart of NIP-RW is shown in Fig. 1.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent a PPI network, where \mathcal{V} is the set of proteins, and \mathcal{E} is the set of edges. Each vertex $u \in \mathcal{V}$ stands for a unique protein, each edge $(u, v) \in \mathcal{E}$ represents an observed interaction between protein u and protein v , $\mathbf{E} \in \mathbb{R}^{n \times n}$ stores available interactions between n proteins. We define a pair of proteins (u and v) as level- k neighbors if there exists a path $\phi = (u, \dots, v)$ with length k in \mathcal{G} . The k -steps random walk process can be modeled as follows:

$$\mathbf{W}^{(k)} = \mathbf{W}^{(k-1)}\mathbf{E} \quad (2)$$

After k -steps random walk, we can obtain a updated adjacency matrix $\mathbf{W}^{(k)} \in \mathbb{R}^{n \times n}$, which reflects the inferred interaction probability (strength) between any pairwise proteins.

Since \mathbf{E} is generally very sparse, $\mathbf{W}^{(k)}$ still encodes a sparse matrix. As such, the selected negative examples are inclined to proteins connected with few proteins, and lead to a bias of negative examples. To generate a negative dataset with good generalization ability, we use a sub-matrix $\mathbf{W}_{p \times p}$ of $\mathbf{W}^{(k)}$ to control the number of proteins and degree distribution of these selected p proteins. After that, we select two proteins with $\mathbf{W}_{p \times p}(i, j) = 0$ and take these two proteins as a non-interacting pair. We will investigate the parameter sensitivity of p and provide a principal way to specify p in “Contribution of controlling degrees” section.

Feature vector extraction

To effectively predict PPIs based on amino acid sequences, we need to extract and represent the essential information of interacting proteins by a feature descriptor. Many feature descriptors have been utilized to predict PPIs. Among these descriptors, conjoint triad (CT) [20] only takes into account the neighboring effect of amino acid sequences. However, PPIs generally occur at discontinuous segments of amino acid sequences. Local descriptor (LD) [23], auto covariance (AC) [21], multi-scale continuous and discontinuous (MCD) [24] and local conjoint triad descriptor (LCTD) [28] can effectively address this problem and achieve better prediction. Among these four descriptors,

feature vectors encoded by AC have the lowest dimensionality. To balance the effectiveness and efficiency, we employ AC to encode the feature information of amino acid sequences, and then use DNNs to predict PPIs. To be self-inclusive, we introduce the AC feature descriptor in the following subsection.

Auto covariance (AC)

PPIs generally can be divided into four interaction modes: electrostatic, hydrophobic, hydrogen bond, and steric [38]. Seven physicochemical properties of amino acids can reflect these interaction modes whenever possible, including hydrophobicity [39], hydrophilicity [40], volumes of side chains of amino acids [41], polarity [42], polarizability [43], solvent-accessible surface area [44], net charge index of side chains [45]. The original values of these seven physicochemical properties for each amino acid are shown in Table 2. Feature normalization can improve the accuracy and efficiency of mining algorithms on the data [46]. Given that, we firstly normalize data with zero mean and unit standard deviation as follows:

$$P'_{ij} = \frac{P_{ij} - \bar{P}_j}{S_j} \quad (3)$$

where P_{ij} is the j -th physicochemical property value for the i -th amino acid, \bar{P}_j is the mean of the j -th physicochemical property over 20 amino acids and S_j is the corresponding standard deviation of the j -th physicochemical property. Then each amino acid sequence is translated into seven vectors with each amino acid represented by the normalized values.

AC is a statistical tool introduced by Wold et al. [38], it is adopted to transform amino acid sequences into uniform matrices. AC can account for the interactions between residues using a certain *lag* apart the entire sequence. To represent an amino acid sequence A with length l , the AC variables are computed as:

$$AC(lag, j) = \frac{1}{l-lag} \sum_{i=1}^{l-lag} \left(A_{ij} - \frac{1}{l} \sum_{i=1}^l A_{ij} \right) \times \left(A_{(i+lag),j} - \frac{1}{l} \sum_{i=1}^l A_{ij} \right) \quad (4)$$

lag is the distance between residues. A_{ij} is the j -th physicochemical property of the i -th amino acid of A , l is the length of the amino acid sequence A . In this way, the number of AC variables is $D = lg \times p$, where p is the number of descriptors, which is set as 7 according to seven properties of amino acids. lg is the maximum distance *lag* ($lag = 1, 2, \dots, lg$), which is set as 30 [21]. After that, each amino acid sequence is encoded by a 210-dimensional vector

Table 2 The original values of the seven physicochemical properties for each amino acid

Code	H_1	H_2	V	P_1	P_2	SASA	NCI
A	0.62	-0.5	27.5	8.1	0.046	1.181	0.007187
C	0.29	-1	44.6	5.5	0.128	1.461	-0.03661
D	-0.9	3	40	13	0.105	1.587	-0.02382
E	-0.74	3	62	12.3	0.151	1.862	0.006802
F	1.19	-2.5	115.5	5.2	0.29	2.228	0.037552
G	0.48	0	0	9	0	0.881	0.179052
H	-0.4	-0.5	79	10.4	0.23	2.025	-0.01069
I	1.38	-1.8	93.5	5.2	0.186	1.81	0.021631
K	-1.5	3	100	11.3	0.219	2.258	0.017708
L	1.06	-1.8	93.5	4.9	0.186	1.931	0.051672
M	0.64	-1.3	94.1	5.7	0.221	2.034	0.002683
N	-0.78	2	58.7	11.6	0.134	1.655	0.005392
P	0.12	0	41.9	8	0.131	1.468	0.239531
Q	-0.85	0.2	80.7	10.5	0.18	1.932	0.049211
R	-2.53	3	105	10.5	0.291	2.56	0.043587
S	-0.18	0.3	29.3	9.2	0.062	1.298	0.004627
T	-0.05	-0.4	51.3	8.6	0.108	1.525	0.003352
V	1.08	-1.5	71.5	5.9	0.14	1.645	0.057004
W	0.81	-3.4	145.5	5.4	0.409	2.663	0.037977
Y	0.26	-2.3	117.3	6.2	0.298	2.368	0.023599

H_1 : hydrophobicity; H_2 : hydrophilicity; V : volume of side chains; P_1 : polarity; P_2 : polarizability; SASA: solvent accessible surface area; NCI : net charge index of side chains

with AC variables. Finally, feature vectors of two individual proteins are taken as inputs of two separate DNNs, respectively.

Deep neural networks

Deep learning, the most active field in machine learning, attempts to learn multi-layered models of inputs. It has been achieving great success in many research areas, such as speech recognition [47], signal recognition [48], computer vision [49–51], natural language processing [52, 53] and so on. Meanwhile, it also has been widely employed in bioinformatics [54, 55]. Deep learning is not only good at automatically learning the high-level features from the original data, but also good at discovering intricate structures in high-dimensional data [56].

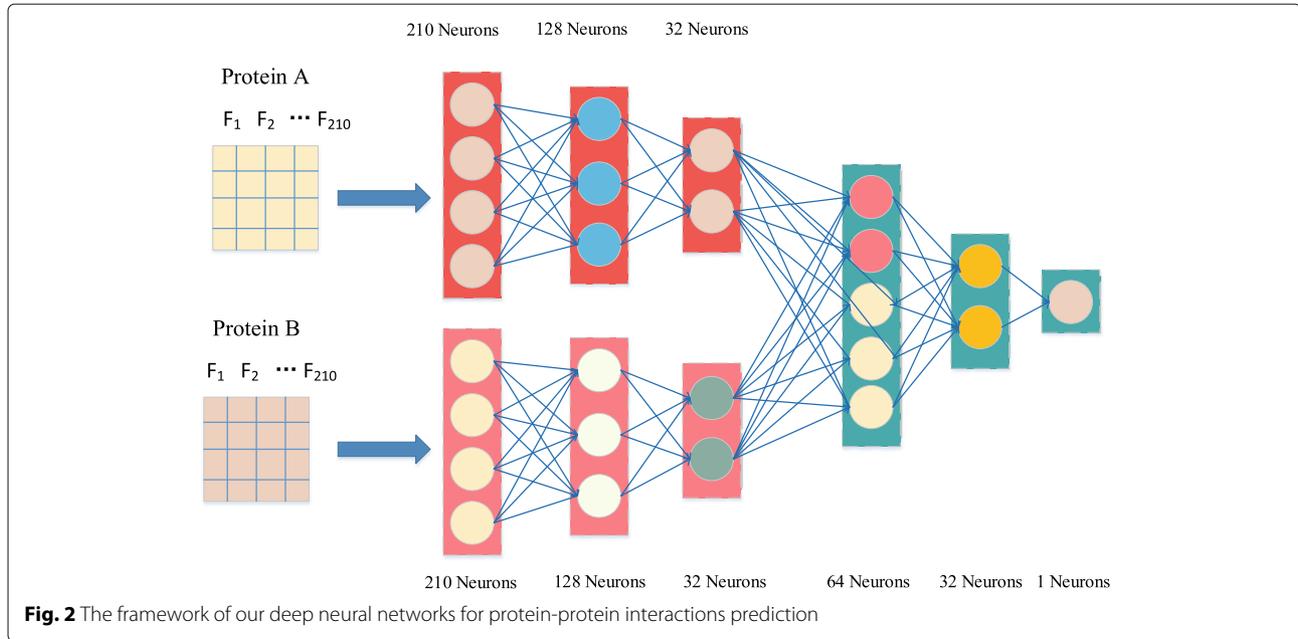
Deep neural networks (DNNs) are composed of an input layer, multiple hidden layers (three or more hidden layers), and an output layer, the configuration of adopted DNNs is shown in Fig. 2. In general, neural networks are fed data from the input layer (x), then the output values are sequentially computed along with hidden layers by transforming input data in a nonlinear way. Neurons of a hidden layer or output layer are connected to all neurons of the previous layer [32]. Each neuron computes a weighted sum of its inputs and applies a nonlinear activation function to calculate its outputs $f(x)$ [32]. The

nonlinear activation functions usually include sigmoid, hyperbolic tangent, or rectified linear unit (ReLU) [57]. ReLU and sigmoid are employed in this work.

We separately construct two DNNs using TensorFlow platform, as illustrated in Fig. 2. Next, the feature vectors of two individual proteins extracted by AC are employed as the inputs for these two DNNs, respectively. After that, these two separate DNNs were combined in a hidden layer to predict PPIs. Adam algorithm [58] (an adaptive learning rate methods) is applied to speed up training. Meanwhile, the dropout technique is employed to avoid overfitting. The ReLU activation function [57] and cross entropy loss are employed, since they can both accelerate the model training and obtain better prediction results [59]. The batch normalization approach is also applied to reduce the dependency of training with the parameter initialization, speed up training and minimize the risk of overfitting. The following equations are used to calculate the loss:

$$\mathbf{H}_{i1}^m = \sigma_1(\mathbf{W}_{i1}\mathbf{X}_{i1} + \mathbf{b}_{i1}) (i = 1, \dots, n; m = 1, 2) \quad (5)$$

$$\mathbf{H}_{ij}^m = \sigma_1(\mathbf{W}_{ij}\mathbf{H}_{i(j-1)} + \mathbf{b}_{ij}) \quad (i = 1, \dots, n; j = 2, \dots, h_1; m = 1, 2) \quad (6)$$



$$\mathbf{H}_{ik}^3 = \sigma_1 \left(\mathbf{W}_{ik} \left(\mathbf{H}_{ih_1}^1 \oplus \mathbf{H}_{ih_1}^2 \right) + \mathbf{b}_{ik} \right) \quad (7)$$

$(i = 1, \dots, n, k = h_1 + 1)$

$$\mathbf{H}_{ik}^3 = \sigma_1 \left(\mathbf{W}_{ik} \mathbf{H}_{i(k-1)} + \mathbf{b}_{ik} \right) \quad (8)$$

$(i = 1, \dots, n; k = h_1 + 2, \dots, h_2)$

$$L = -\frac{1}{n} \sum_{i=1}^n \left[\mathbf{y}_i \ln(\sigma_2(\mathbf{W}_{ih_2} \mathbf{H}_{ih_2} + \mathbf{b}_{ih_2})) + (1 - \mathbf{y}_i) \ln(1 - \sigma_2(\mathbf{W}_{ih_2} \mathbf{H}_{ih_2} + \mathbf{b}_{ih_2})) \right] \quad (9)$$

where n is the number of PPIs for batch training, m represents the individual network, and h_1 is the depth of two individual networks, h_2 is the depth of fused network. σ_1 is the activation function of ReLU, σ_2 is the activation function of the output layer with sigmoid, \oplus represents the concatenation operator. \mathbf{X} is the batch training inputs, \mathbf{H} is the output of hidden layer, and \mathbf{y} is the corresponding desired outputs. \mathbf{W} is the weight matrix between the input layer and output layer, \mathbf{b} is the bias.

Results and discussion

In this section, we briefly introduce several widely-used evaluation criteria for performance comparison, and the recommended configuration of experiments. Next, we analyze and discuss the experimental results and compare our results with those of other related work.

Evaluation metrics

To comprehensively compare the performance, six evaluation metrics are employed, accuracy (ACC), precision

(PE), sensitivity (SEN), specificity (SPE), matthews correlation coefficient (MCC), F_1 score values, and area under the receiver operating characteristic curve (AUC). These metrics (except AUC) are computed as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

$$SEN = \frac{TP}{TP + FN} \quad (11)$$

$$SPE = \frac{TN}{TN + FP} \quad (12)$$

$$PE = \frac{TP}{TP + FP} \quad (13)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (14)$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \quad (15)$$

where true positive (TP) stands for the number of true PPIs which are correctly predicted; false negative (FN) stands for the number of true PPIs which are incorrectly predicted as non-interacting pairs; false positive (FP) is the number of true non-interacting pairs which are predicted as interacting pairs; true negative (TN) represents the number of true non-interacting pairs which are correctly predicted. MCC is considered as the most robust metric of a binary classifier. MCC equal to 0 represents completely random prediction, whereas 1 means perfect prediction. F_1 score is a harmonic average of precision and sensitivity, and a larger score indicates a better performance. Receiver operating characteristic (ROC) curve

is also employed to assess the performance of prediction model. To summarize ROC curve in single quantity, the area under ROC curve (AUC) is used. AUC ranges from 0 to 1, the maximum value 1 stands for perfect prediction. For a random guess, the AUC value is close to 0.5.

Experimental setup

Our approach is implemented on TensorFlow platform <https://www.tensorflow.org>. We firstly constructed the negative datasets using four different strategies. We then encoded the amino acid sequences from the datasets using auto covariance (AC) [24]. After that, we trained two separate neural networks with graphics processing unit (GPU) based on the feature sets encoded by AC. Finally, we fused these two networks to predict new PPIs. Deep learning algorithms contains a number of hyper-parameters, which may heavily impact the experimental results. The recommended hyper-parameters configuration of our proposed model is summarized in Table 3. As to the parameter specification of the comparing methods, we employed the grid search to obtain the optimal parameters, which are shown in Table 4. For Du et al. [27] work, they also provided with a similar hyper-parameters configuration with ours, which can be accessed via the reference [27]. All the experiments are carried out on a server with configuration: CentOS 7.3, 256GB RAM, and Intel Exon E5-2678 v3. Meanwhile, we used NVIDIA Corporation GK110BGL [Tesla K40c] to accelerate training of DNNs.

Contribution of controlling degrees

For the negative dataset generated by NIP-SS, we select the top- m protein-protein pairs with the lowest sequential similarity as the negative PPIs. Among all protein

pairs, the similarity between these protein pairs is minimum. However, there are some proteins having very large degrees, which will lead to a bias and overestimation of prediction results. Therefore, we need to control the degree distribution of the negative dataset, and approximate the distribution with that of the positive dataset to guarantee the generalization ability of negative examples. Table 5 reports the degree distribution of proteins in *S. cerevisiae*, *H. sapiens* and *M. musculus*, and Fig. 3 reveals the prediction performance of NIP-SS with and without controlling the degree of proteins related to negative samples.

From Table 5, we can see that the maximum degree of proteins in the negative dataset (NIP-SS-NonControl) is 1439, and the proportion of non-interactions with degree larger than 150 is 27.39%, which may lead to a bias. As a result, using this datasets produce a higher accuracy of 97.05%. Compared to NIP-SS-NonControl, the negative dataset constructed by NIP-SS contains more proteins and smaller maximum degree. Meanwhile, non-interactions are mainly related to proteins whose degrees fewer than 50. As such, the negative dataset generated by NIP-SS has a better generalization ability and lower bias than that by NIP-SS-NonControl. The contribution of controlling the degree of proteins in the negative dataset is also significant on *H. sapiens* and *M. musculus* datasets.

If we directly select protein pairs whose corresponding entries equal to 0 in the updated $\mathbf{W}^{(k)}$ to generate the negative dataset, such a dataset brings less predictive information and is not conducive for predicting PPIs, since this dataset contains many proteins with low degrees. Therefore, a sub-matrix $\mathbf{W}_{p \times p}$ is employed to control the degree distribution of proteins. In addition, k also affects the degree distribution. Given that, we need to specify suitable input values of p and k . Particularly, we firstly fix k to 3, and then tune p from 500 to 4382 with an interval of 500. Next, we calculate the average repeatability (r), maximum degree of proteins, the proportion of proteins in different ranges of degrees, and the proportion of non-interactions in each range. We then choose p that makes the degree of proteins in the negative dataset similar to those of the positive dataset. After that, we adopt p selected in the first step and tune k within {1, 2, 3, 6, 10, 50, 300, 1000}.

The degree distribution and prediction results on *S. cerevisiae* are shown in Table 6 and Fig. 4, respectively. From Table 6, we can see that when $p \approx 2000$, the degree distribution of the negative dataset is most similar to that of the positive dataset. In addition, from Fig. 4, we can also observe that when $n < 2000$, the accuracy with the setting $p = 500, 1000$ and $k = 3$ is higher than 95%. This is because the average repeatability is large and leads to a bias.

Table 3 Recommended parameters of our model

Name	Range	Recommend
Learning rate	1, 0.1, 0.001, 0.002, 0.003, 0.0001	0.002
Batch size	32, 64, 128, 256, 512, 1024, 2056	1024, 2056
Weight initialization	uniform, normal, lecun_uniform, glorot_normal, glorot_uniform	glorot_normal
Per-parameter adaptive learning rate	SGD, RMSprop, Adagrad, Adadelata, Adam, Adamax, Nadam	Adam
Activation function	relu, tanh, sigmoid, softmax, softplus	relu, sigmoid
Dropout rate	0.5, 0.6, 0.7	0.6
Depth	2, 3, 4, 5, 6, 7, 8, 9	3
Width	16, 32, 64, 128, 256, 1024, 2048, 4096	128, 64, 32
GPU	Yes, No	Yes

Table 4 Optimal parameters of comparing methods

Method	Name	Parameters			
Guo's work [21]	SVM+AC	C	γ		Kernel
		32768.0	0.074325444687670064		Poly
Yang's work [22]	kNN+LD	n_neighbors	Weights	Algorithm	p
		3	Distance	Auto	1
Zhou's work [23]	SVM+LD	C	γ		Kernel
		3.1748021	0.07432544468767006		rbf
You's work [25]	RF+MCD	n_estimators	Max_features	Criterion	Bootstrap
		5000	Auto	Gini	True

The similar parameter selection strategy is also conducted on the other two datasets. The experimental results and the degree distribution of proteins are shown in Fig. 4, Tables 6, and 7. According to Table 6, we set $p = 700$ and $p = 300$ for *H. sapiens* and *M. musculus*, respectively. In addition, according to the right of Fig. 4 and Table 7, we fix $k = 3$ for *H. sapiens* and $k = 50$ for *M. musculus*. From Table 6, we can also observe that when p is set to $n/2$ (or $n/3$, n is the number of proteins in the positive set), the degree distribution generally approximates well with that of the positive dataset.

Results of different negative dataset construction strategies

To investigate the effectiveness of the proposed two strategies for constructing negative dataset, we conduct experiments on three prevalent PPIs datasets, including *S. cerevisiae*, *H. sapiens* and *M. musculus* datasets, and take the performance of PPIs prediction as the comparing index. To avoid over-fitting and data dependency, five-fold cross-validation is adopted. Table 8 reports the average prediction results on these three species using different negative dataset generation strategies.

We can see that for the *S. cerevisiae* dataset, the model based on the negative dataset generated by NIP-SS gives the average accuracy of 94.34%, precision of 95.62%, recall of 92.96%, specificity of 95.74%, MCC of 88.73%, F_1 of 94.27% and AUC of 98.24%, respectively. These values are higher than those of other strategies, which separately adopt random walk, random pairing, subcellular localization to generate the negative dataset. These results prove the effectiveness of NIP-SS in generating reliable non-interacting protein pairs for PPIs prediction. In addition, the negative dataset constructed by NIP-SS contain more proteins and have similar degree distributions to the positive dataset, which can effectively control the bias of the dataset. The model trained on the negative dataset generated by random pairing yields very low accuracy of 74.20%. That is because this negative dataset has a high

rate of false negatives, and the degree distribution mainly concentrates on proteins with degree smaller than 10. The model based on negative dataset generated by subcellular localization also yields a good performance with accuracy of 93.79%, MCC of 87.62%, and AUC of 98.13%. However, compared to the negative dataset generated by NIP-SS, this dataset covers fewer proteins and a larger proportion of non-interactions in the degree range 50-70, which are higher than those NIP-SS. Those will produce an over-optimistic estimate of prediction.

The model trained on the negative dataset generated by NIP-RW yields an average accuracy of 87.92%, MCC of 75.97% and AUC of 94.23%. These values are lower than those of NIP-SS. That is mainly because the proteins in the negative datasets generated by NIP-SS and NIP-RW have different degrees. 21.05% non-interacting protein pairs in the negative dataset generated by NIP-SS are located in range of degree larger than 50, but no non-interacting protein pairs in the negative dataset generated by NIP-RW are located in that range. Another reason is that random walk process is restricted by the connected positive examples. For the small network of *H. sapiens* and *M. musculus* datasets, NIP-RW yields good results.

As to the *H. sapiens* and *M. musculus* datasets, we can observe that the model based on the negative datasets of subcellular localization yields the best prediction accuracy of 93.34% and 91.82%, respectively. We find the negative datasets constructed by subcellular localization has the maximum average repeatability (r) and contains the fewest proteins, which lead to a bias and an overestimated performance. Since the degree distribution of negative datasets constructed by NIP-SS and NIP-RW are similar, the prediction performance using these two strategies are similar. The model based on negative datasets generated by random pairing again gives the lowest performance.

To further investigate the effectiveness of our model that uses two separate DNNs at first, we introduced a variant of our model called DNNs-Con. DNNs-Con firstly concatenates AC features of two individual proteins, and then

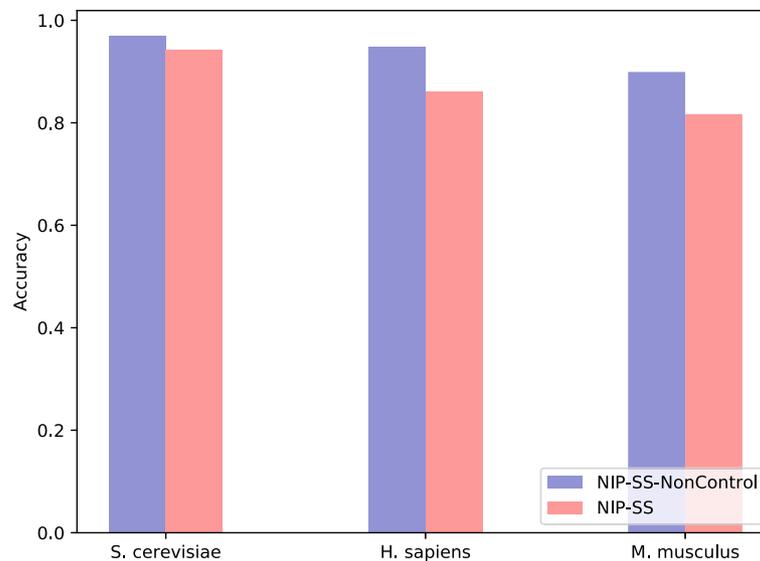


Fig. 3 The experimental results of NIP-SS-NonControl and NIP-SS on *S. cerevisiae*, *H. sapiens*, and *M. musculus*. The negative datasets constructed by NIP-SS control the degree distribution of proteins

takes the concatenated features as input of DNNs. The hidden layers for this network are fixed as 420-256-32. To check the statistical significance between our model and DNNs-Con, the pairwise *t*-test (at 95% significance level) is also used. The experimental results of five-fold cross validation are reported in Table 9. From Table 9, we can observe that the accuracy, MCC, F_1 and AUC of our model are 2.61%, 5.22%, 2.68% and 1.29% higher than those of DNNs-Con, respectively. In addition, we observe that our model converges faster than DNNs-Con during the training process, that is due to two separate networks can faster extract sequence information contained in each amino acid sequence. These results prove that our model (using two separate DNNs, instead of single one) is efficient and effective to predict PPIs.

Based on the above analysis, we fix $p = 2000$ and vary $k \in \{1, 2, 3, 6, 10, 50, 300, 1000\}$. Figure 4 (right of this Figure) reports the results under different values of k . We also calculate the degree distribution at different k , which are listed in Table 7. From the right of Fig. 4, we can observe that when $k \geq 6$, the result is close to 1. That is because there are more nonzero entries in $\mathbf{W}^{(k)}$ as k increases, which change the degree distribution of proteins and thus bring in a larger bias. Table 7 shows the degree distribution when $p = 2000$. Based on these results, we fix k to 3.

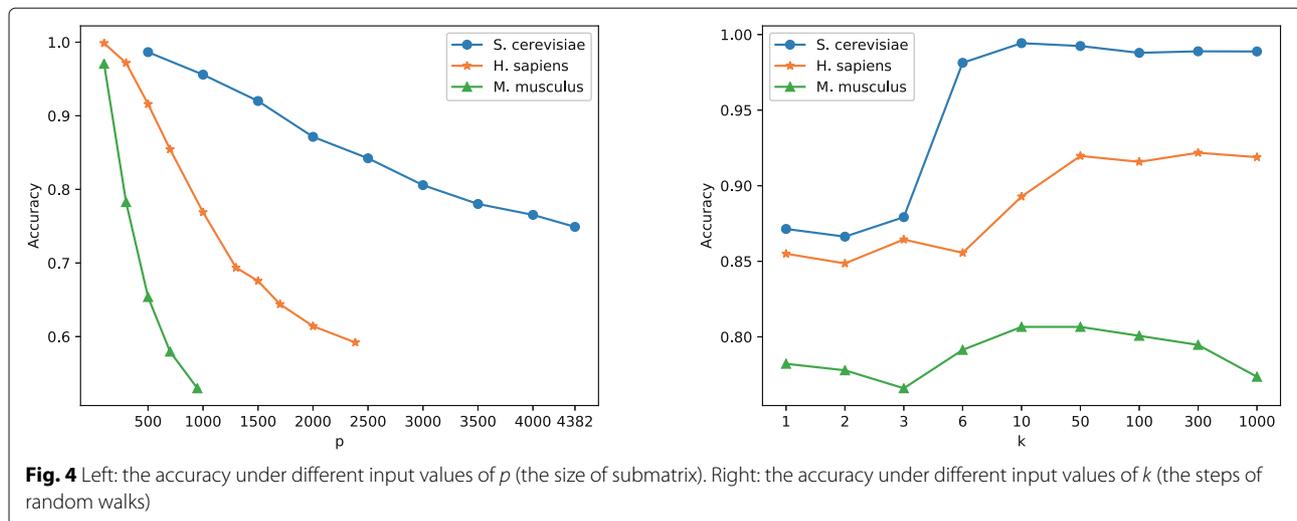
The impact of of imbalanced class

In general, the number of negative PPIs has a large impact on prediction performance. To investigate the impact of

imbalanced class on our proposed two strategies, three *H. sapiens* datasets are constructed with different numbers of negative samples for NIP-SS and NIP-RW, respectively. The ratios of positive samples (3355 interaction pairs) and negative samples in these three datasets are 1:1, 1:2 and 1:3, respectively. Four metrics of sensitivity (SEN), specificity (SPE), area under the receiver operating characteristic curve (AUC), and geometric mean (GM) are used to evaluate the prediction performance. GM is commonly used for class-imbalance learning [60], it can give a more accurate evaluation on imbalanced data. The GM is calculated by this formula: $GM = \sqrt{SEN \times SPE}$. The prediction results are shown in Table 10. From the Table 10, we can see that as the number of negative samples increases, the overall performance of the model shows a downward trend. In addition, the prediction values of AUC and GM decrease significantly. AUC is respectively decreased by 10.51% and 8.27% for NIP-SS and NIP-RW, and GM is decreased by 16.63% and 11.27%. Given that, to avoid the performance degradation caused by imbalanced class, we adopt the widely-used solution that uses the same number of negative PPIs as that of positive samples.

The impact of different feature descriptors

The extracted features can affect the performance of PPIs prediction [28]. To investigate the contribution of auto covariance (AC) [21] descriptor, we separately train DNNs on *S. cerevisiae* (the negative dataset constructed by NIP-SS) based on AC [21], CT [20], LD [23], and MCD [25]. Table 11 reports the results of five-fold cross



validation. Meanwhile, we also use pairwise t -test (at 95% significance level) to check the statistical significance between AC and CT, LD, MCD. From Table 11, we observe that DNNs-AC achieves an average accuracy as 94.25%, precision as 94.7%, recall as 93.75%, specificity as 94.74%, MCC as 88.5%, F_1 as 94.22%, and AUC as 98.15%. The performance difference of these descriptors is not significant, but AC descriptors have the smallest feature dimension. For this reason, we adopt AC to encode amino acid sequences.

Comparison with existing methods

To further study the performance of our model and the contribution of negative dataset generated by NIP-SS and NIP-RW, we compare our prediction results on *S. cerevisiae* with those of other competitive methods, including Guo et al. [21], Yang et al. [22], Zhou et al. [23], You et al. [25], and Du et al. [27]. These approaches were introduced in “Background” section.

Table 12 shows the experimental results. Our method yields average prediction accuracy of 94.34%, precision of 95.62%, recall of 92.96%, MCC of 88.73%, F_1 of 94.27%, and AUC of 98.24%. Compared to the other two negative datasets, the negative dataset constructed by NIP-SS covers more proteins and the degree distribution is close to the degree distribution of the positive dataset. In addition, we can observe that the comparing methods using the negative dataset constructed by NIP-RW also produces good results. However, for a large dataset, the degrees of proteins in the negative dataset generated by NIP-RW are almost always smaller than 50. This is because the distribution of degree is restricted by the collected positive examples and a large network makes the random walk process less

controlled. For this reason, the NIP-RW is reliable on *H. sapiens* and *M. musculus*. These results prove that the negative datasets constructed by NIP-SS and NIP-RW are rational and can boost the performance of PPI prediction.

Results on independent datasets

Six independent datasets, which just only contain the examples of interactions (non-interactions), including *Caenorhabditis elegans* (4013 interacting pairs), *Escherichia coli* (6954 interacting pairs), *Helicobacter pylori* (1420 interacting pairs), *Homo sapiens* (1412 interacting pairs), *Mus musculus* (313 interacting pairs), and *Mammalian* (1937 non-interacting pairs), are employed as test sets to evaluate the generalization ability, and to further assess the practical prediction ability of our model and the rationality of NIP-SS and NIP-RW. Three datasets of *H. sapiens* (3355 positive examples and 3355 negative examples) are constructed and the difference between these datasets is the negative samples, which are generated by NIP-SS, NIP-RW, and subcellular location, respectively. Then, three models with optimal configuration (provided in “Evaluation metrics” section) are trained on these three datasets. After that, these six independent datasets are used to test the generalization ability of these models. The prediction results are shown in Table 13. From Table 13, we can observe that the accuracy of our model using the negative datasets generated by NIP-SS and NIP-RW on *C. elegans*, *E. coli*, *H. sapiens*, *H. pylori*, *M. musculus*, and *Mammalian* are 86.10%, 85.34%, 86.20%, 81.86%, 85.64%, 15.69% and 78.113%, 79.65%, 85.03%, 79.15%, 80.66%, 18.58%, respectively. These prediction results indicate that the negative datasets generated by NIP-SS and NIP-RW

Table 8 Results based on different negative datasets on *S. cerevisiae*, *H. sapiens* and *M. musculus*

Species	Negative samples	ACC	PE	RE	SPE	MCC	F ₁	AUC
<i>S. cerevisiae</i>	NIP-SS	94.34%±0.38%	95.62%±0.75%	92.96%±0.40%	95.74%±0.75%	88.73%±0.77%	94.27%±0.34%	98.24%±0.11%
	NIP-RW	87.92%±0.24%	90.04%±1.69%	85.32%±1.90%	90.48%±2.20%	75.97%±0.55%	87.59%±0.35%	94.23%±0.12%
	Sub	93.79%±0.43%	95.18%±0.41%	92.25%±0.78%	95.33%±0.45%	87.62%±0.83%	93.69%±0.38%	98.13%±0.17%
	Random method	74.20%±0.78%	72.68%±1.45%	77.59%±0.89%	70.83%±1.77%	48.53%±1.47%	75.04%±0.76%	81.29%±0.34%
<i>H. sapiens</i>	NIP-SS	86.17%±0.93%	86.38%±1.27%	85.88%±1.55%	86.48%±1.00%	72.36%±1.85%	86.12%±1.05%	92.20%±0.82%
	NIP-RW	86.44%±0.59%	90.05%±0.48%	81.87%±2.35%	90.91%±1.09%	73.14%±1.12%	85.75%±1.28%	92.30%±0.70%
	Sub	93.34%±0.58%	93.19%±0.42%	93.51%±0.94%	93.17%±0.37%	86.68%±1.16%	93.35%±0.57%	96.22%±0.45%
	Random method	60.46%±1.54%	60.07%±1.74%	62.33%±1.91%	58.50%±3.24%	20.85%±3.14%	61.17%±1.63%	64.57%±1.35%
<i>M. musculus</i>	NIP-SS	81.69%±1.48%	80.57%±2.20%	83.73%±2.97%	79.51%±4.47%	63.44%±3.16%	82.06%±0.84%	87.04%±1.95%
	NIP-RW	80.66%±2.14%	84.89%±5.41%	74.83%±3.46%	86.72%±4.62%	61.97%±4.53%	79.41%±2.52%	87.75%±2.25%
	Sub	91.82%±1.26%	90.13%±2.57%	93.93%±2.38%	89.76%±2.41%	83.78%±2.40%	91.95%±1.44%	94.81%±0.74%
	Random method	50.76%±2.12%	50.80%±5.77%	52.17%±1.90%	49.44%±3.26%	1.60%±3.86%	51.37%±3.58%	51.40%±2.43%

Table 9 Results of different network architectures on *S. cerevisiae*, the adopted negative dataset is constructed by NIP-SS

Architectures	Data set	ACC	PE	RE	SPE	MCC	F ₁	AUC
DNNS	Fold 1	94.08%	94.04%	94.17%	93.98%	88.15%	94.11%	98.24%
	Fold 2	94.03%	94.36%	93.64%	94.42%	88.07%	94.00%	98.13%
	Fold 3	94.57%	95.25%	93.66%	95.45%	89.14%	94.45%	98.17%
	Fold 4	94.38%	94.99%	93.78%	94.98%	88.77%	94.38%	98.16%
	Fold 5	94.19%	94.84%	93.50%	94.88%	88.39%	94.17%	98.03%
	Average	94.25%±0.22%	94.70%±0.49%	93.75%±0.26%	94.74%±0.56%	88.5%±0.45%	94.22%±0.19%	98.15%±0.08%
DNNS-Con	Fold 1	91.92%	92.40%	91.27%	92.55%	83.84%	91.83%	97.15%
	Fold 2	91.86%	93.87%	89.21%	94.40%	83.79%	91.48%	96.90%
	Fold 3	91.58%	93.62%	89.32%	93.86%	83.26%	91.42%	96.83%
	Fold 4	91.86%	93.65%	90.07%	93.70%	83.79%	91.83%	96.92%
	Fold 5	91.42%	92.24%	90.53%	92.32%	82.86%	91.38%	96.93%
	Average	91.73%±0.21%●	93.16%±0.77%●	90.08%±0.86%●	93.37%±0.89%●	83.51%±0.43%●	91.59%±0.23%●	96.95%±0.12%●

●/○ indicates whether our model is statistically (according to pairwise t-test at 95% significance level) superior/inferior to the DNNS-Con

Table 10 Results on *H. sapiens* with different numbers of negative samples for NIP-SS and NIP-RW

Method	Dataset	SEN	SPE	AUC	GM
NIP-SS	<i>H.sapiens</i> _{1,1}	86.57%	87.08%	92.01%	86.57%
	<i>H.sapiens</i> _{1,2}	69.95%	89.25%	86.33%	79.00%
	<i>H.sapiens</i> _{1,3}	52.93%	92.48%	81.50%	69.94%
NIP-RW	<i>H.sapiens</i> _{1,1}	81.87%	90.91%	92.30%	86.27%
	<i>H.sapiens</i> _{1,2}	72.84%	90.13%	87.41%	81.02%
	<i>H.sapiens</i> _{1,3}	58.33%	94.51%	84.03%	75.00%

contribute to a good performance across species. We note that the accuracy on *Mammalian* using the NIP-SS and NIP-RW strategies are 3.36 and 3.98 times higher than that using subcellular localization (which is only 4.67%). Given that, we can conclude that the negative dataset generated by subcellular localization may produce a bias for predicting PPIs. In other words, subcellular localization based negative examples generation strategy is inclined to predict a new protein pair as interaction. To further demonstrate this discovery and the advantages of NIP-SS and NIP-RW, we constructed a dataset (named *Mammalian-imbalanced*), in which the number of negative samples is about 4 times than that of positive samples, since the number of protein pairs (non-interacting) is far greater than the number of interaction pairs in the real world. The negative samples are from *Mammalian* dataset (1937 negative samples), while the positive are from the *M. musculus* (313 positive samples). Finally, the dataset contains 313 + 1937 protein pairs. The prediction results are also shown in Table 13. From Table 13, we can see that the accuracy on *Mammalian-imbalanced* dataset using the NIP-SS and NIP-RW strategies are 23.45% and 27.56%, respectively, which are both higher than that using subcellular localization (only 17.75%). These prediction results show that NIP-SS and NIP-RW hold a good generalization ability and performance in predicting PPIs, and the strategies of subcellular location will lead to a bias in predicting.

Conclusion and future work

Effective PPIs prediction approaches depend on a high quality negative dataset (non-interacting protein pairs), which contributes to discriminative and accurate prediction. In this paper, we present two novel strategies (NIP-SS and NIP-RW) to generate high-quality negative dataset and to boost the performance of PPIs prediction. NIP-SS uses sequence similarity between proteins to guide the generation of negative examples, whereas NIP-RW utilizes the interaction profiles of proteins to select negative examples. To reduce the bias and enhance the generalization ability of the generated negative dataset, these two strategies separately adjust the degree of the non-interacting proteins and approximate the degree to that of the positive dataset. We found that NIP-SS is competent on all datasets and hold a good performance, whereas NIP-RW can only obtain a good performance on small dataset (positive samples ≤ 6000) because of the restriction of random walk and the results of extensive experiments. In addition, these experiments also indicate that the negative datasets constructed by NIP-SS and NIP-RW can significantly improve the performance of PPIs prediction and these two strategies work better than other two widely adopted strategies.

We will fuse multiple types of biological data, including the sequence similarity, functional similarity and domain similarity of proteins, to generate the negative datasets. In addition, we will investigate more intelligent ways to adjust the degree of non-interacting proteins.

Table 11 Results of DNNs with AC, CT, LD and MCD feature descriptors on *S. cerevisiae*

Model	Dimension	ACC	PE	RE	SPE	MCC	F ₁	AUC
DNNs-AC	(210+210)	94.25%±0.22%	94.70%±0.49%	93.75%±0.26%	94.74%±0.56%	88.50%±0.45%	94.22%±0.19%	98.15%±0.08%
DNNs-CT	(343+343)	94.37%±0.24%	95.55%±0.75%	93.09%±0.81%	95.67%±0.65%	88.78%±0.48%	94.30%±0.23%	98.20%±0.21%
DNNs-LD	(630+630)	94.41%±0.14%	95.46%±0.41%	93.25%±0.44%	95.56%±0.44%	88.84%±0.28%	94.34%±0.15%	98.23%±0.06%
DNNs-MCD	(882+882)	94.25%±0.22%	94.70%±0.49%	93.75%±0.26%	94.74%±0.56%	88.50%±0.45%	94.22%±0.19%	98.15%±0.08%

●/○ indicates whether DNNs-AC is statistically (according to pairwise *t*-test at 95% significance level) superior/inferior to the other descriptors

Table 12 Results of our modal and of other state-of-the-art methods on *S. cerevisiae*

Method	Negative samples	ACC	PE	RE	SPE	MCC	FI	AUC
Our method	Sub	93.79%±0.43%	95.18%±0.41%	92.25%±0.78%	95.33%±0.45%	87.62%±0.83%	93.69%±0.38%	98.13%±0.17%
	NIP-SS	94.34%±0.38%	95.62%±0.75%	92.96%±0.40%	95.74%±0.75%	88.73%±0.77%	94.27%±0.34%	98.24%±0.11%
	NIP-RW	87.92%±0.24%	90.04%±1.69%	85.32%±1.90%	90.48%±2.20%	75.97%±0.55%	87.59%±0.35%	94.23%±0.12%
Du's work [27]	Sub	92.58%±0.38%	94.21%±0.45%	90.95%±0.41%	94.41%±0.45%	85.41%±0.76%	92.55%±0.39%	97.55%±0.16%
	NIP-SS	94.44%±0.35%	95.46%±0.38%	93.44%±0.45%	95.45%±0.41%	88.90%±0.68%	94.44%±0.37%	98.22%±0.20%
	NIP-RW	88.59%±0.32%	92.61%±0.41%	84.14%±0.43%	93.13%±0.35%	77.52%±0.59%	88.17%±0.34%	94.73%±0.18%
You's work [25]	Sub	89.15%±0.33%	90.00%±0.57%	88.10%±0.17%	90.21%±0.61%	78.33%±0.67%	89.04%±0.31%	94.78%±0.21%
	NIP-SS	94.42%±0.47%	96.71%±0.47%	91.96%±0.64%	96.87%±0.46%	88.94%±0.92%	94.28%±0.49%	98.46%±0.12%
	NIP-RW	86.03%±0.43%	89.19%±0.60%	82.00%±0.70%	90.06%±0.64%	72.30%±0.85%	85.44%±0.46%	93.33%±0.46%
Zhou's work [23]	Sub	88.76%±0.37%	89.44%±0.27%	87.89%±0.45%	89.62%±0.30%	77.53%±0.53%	88.66%±0.28%	94.69%±0.31%
	NIP-SS	92.10%±0.34%	93.48%±0.45%	90.51%±0.73%	93.68%±0.49%	84.24%±0.67%	91.97%±0.37%	97.29%±0.16%
	NIP-RW	82.64%±0.33%	83.98%±0.34%	80.67%±0.48%	84.61%±0.36%	65.34%±0.65%	82.30%±0.35%	90.00%±0.39%
Yang's work [22]	Sub	84.81%±0.37%	87.53%±0.14%	81.18%±0.84%	88.44%±0.18%	69.80%±0.71%	84.23%±0.47%	90.03%±0.31%
	NIP-SS	89.18%±0.35%	93.34%±0.33%	84.38%±0.53%	93.98%±0.31%	78.73%±0.69%	88.64%±0.38%	95.50%±0.20%
	NIP-RW	83.98%±0.48%	86.09%±0.67%	81.07%±0.80%	86.89%±0.76%	68.09%±0.97%	83.50%±0.51%	91.45%±0.27%
Guo's work [21]	Sub	87.88%±0.56%	88.16%±0.90%	87.53%±0.59%	88.24%±1.02%	75.77%±1.12%	87.84%±0.53%	93.69%±0.33%
	NIP-SS	90.00%±0.43%	90.45%±0.68%	89.45%±0.69%	90.55%±0.77%	80.01%±0.86%	89.94%±0.43%	95.02%±0.27%
	NIP-RW	82.43%±0.27%	83.48%±0.40%	80.87%±0.45%	83.99%±0.49%	64.89%±0.54%	82.15%±0.27%	89.04%±0.33%

Table 13 Prediction results on seven independent PPIs datasets, PPIs of *H. sapiens* are used as the training set

Species	Test pairs	Negative		
		NIP-SS	NIP-RW	Sub
<i>C. elegans</i>	4013 (interactions)	86.10%	78.11%	94.42%
<i>E. coli</i>	6984 (interactions)	85.34%	79.65%	92.68%
<i>H. sapiens</i>	1412 (interactions)	86.20%	85.03%	96.29%
<i>H. pylori</i>	1420 (interactions)	81.86%	79.15%	92.28%
<i>M. musculus</i>	313 (interactions)	85.64%	80.66%	96.10%
Mammalian	1937 (non-interactions)	15.69%	18.58%	4.67%
Mammalian-imbalanced	2250 (313 interactions, 1937 non-interactions)	23.45%	27.56%	17.75%

Acknowledgements

The authors would like to thank the anonymous reviewers for their critical reading and helpful comments and suggestions, which allowed us to improve the quality of this manuscript. This work will be as a oral presentations in 29th International Conference on Genome Informatics on December 3, 2018.

Funding

This work is supported by Natural Science Foundation of China (No. 61873214, 61872300, 61741217, 61871020, 61571163, 61532014), the National Key Research and Development Plan Task of China (Grant No. 2016YFC0901902), Natural Science Foundation of CQ CSTC (cstc2018jcyjAX0228 and cstc2016jcyjA0351), the Open Research Project of The Hubei Key Laboratory of Intelligent Geo-Information Processing (KLGIP-2017A05), and Chongqing Graduate Student Research Innovation Project (No. CYS18089). Publication costs are funded by Natural Science Foundation of China No. 61741217.

Availability of data and materials

The source codes and datasets of NIP-SS and NIP-RW are available at <http://mlda.swu.edu.cn/codes.php?name=NIP>.

About this supplement

This article has been published as part of *BMC Bioinformatics Volume 19 Supplement 19, 2018: Proceedings of the 29th International Conference on Genome Informatics (GIW 2018): bioinformatics*. The full contents of the supplement are available online at <https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-19-supplement-19>.

Authors' contributions

JW and MG initialized the project and solution, conceived the whole process and revised the manuscript. LZ performed the experiments, analyzed the results and drafted the manuscript. GY analyzed the results and revised the manuscript. All the authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹College of Computer and Information Sciences, Southwest University, Chongqing, China. ²School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China. ³Beijing Key Laboratory of Intelligent Processing for Building Big Data, Beijing, China.

Published: 31 December 2018

References

- Krogan NJ, Cagney G, Yu H, Zhong G, Guo X, Ignatchenko A, Li J, Pu S, Datta N, Tikuisis AP. Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. *Nature*. 2006;440(7084):637.
- Zhang QC, Petrey D, Deng L, Qiang L, Shi Y, Chan AT, Bisikirka B, Lefebvre C, Accili D, Hunter T. Structure-based prediction of protein-protein interactions on a genome-wide scale. *Nature*. 2012;490(7421):556–60.
- Huang H, Alvarez S, Nusinow DA. Data on the identification of protein interactors with the Evening Complex and PCH1 in *Arabidopsis* using tandem affinity purification and mass spectrometry (TAP-MS). *Data Brief*. 2016;8:56–60.
- Foltman M, Sanchez-Diaz A. Studying Protein-Protein Interactions in Budding Yeast Using Co-immunoprecipitation. *Yeast Cytokinesis: Methods and Protocols*. 2016;239–56.
- Chou PY, Fasman GD. Prediction of protein conformation. *Biochemistry*. 1974;13(2):222–45.
- Gavin A-C, Bösch M, Krause R, Grandi P, Marzioch M, Bauer A, Schultz J, Rick JM, Michon A-M, Cruciat C-M. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*. 2002;415(6868):141–7.
- Mehla J, Caufield JH, Uetz P. Mapping protein-protein interactions using yeast two-hybrid assays. *Cold Spring Harb Protoc*. 2015;2015(5):086157.
- Pazos F, Valencia A. In silico two-hybrid system for the selection of physically interacting protein pairs. *Proteins Struct Funct Bioinform*. 2002;47(2):219–27.
- You Z-H, Lei Y-K, Gui J, Huang D-S, Zhou X. Using manifold embedding for assessing and predicting protein interactions from high-throughput experimental data. *Bioinformatics*. 2010;26(21):2744–51.
- Xia J-F, Zhao X-M, Huang D-S. Predicting protein-protein interactions from protein sequences using meta predictor. *Amino Acids*. 2010;39(5):1595–9.
- Xia J-F, Han K, Huang D-S. Sequence-based prediction of protein-protein interactions by means of rotation forest and autocorrelation descriptor. *Protein Pept Lett*. 2010;17(1):137–45.
- Huang C, Morcos F, Kanaan SP, Wuchty S, Chen DZ, Izaguirre JA. Predicting protein-protein interactions from protein domains using a set cover approach. *IEEE/ACM Trans Comput Biol Bioinform*. 2007;4(1):78–87.
- Sheng Q, Lu C. Predicting protein-protein interaction based on protein secondary structure information using bayesian classifier. *J Inn Mong Univ Sci Technol*. 2010;1:021.
- Jansen R, Yu H, Greenbaum D, Kluger Y, Krogan NJ, Chung S, Emili A, Snyder M, Greenblatt JF, Gerstein M. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*. 2003;302(5644):449–53.
- You Z-H, Yin Z, Han K, Huang D-S, Zhou X. A semi-supervised learning approach to predict synthetic genetic interactions by combining functional and topological properties of functional gene network. *Bmc Bioinformatics*. 2010;11(1):343.
- Pazos F, Valencia A. Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein Eng*. 2001;14(9):609–14.
- Wang B, Chen P, Huang D-S, Li J-j, Lok T-M, Lyu MR. Predicting protein interaction sites from residue spatial sequence profile and evolution rate. *FEBS Lett*. 2006;580(2):380–4.

18. Bock JR, Gough DA. Predicting protein-protein interactions from primary structure. *Bioinformatics*. 2001;17(5):455–60.
19. Martin S, Roe D, Faulon J-L. Predicting protein-protein interactions using signature products. *Bioinformatics*. 2004;21(2):218–26.
20. Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, Li Y, Jiang H. Predicting protein-protein interactions based only on sequences information. *Proc Natl Acad Sci*. 2007;104(11):4337–41.
21. Guo Y, Yu L, Wen Z, Li M. Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *Nucleic Acids Res*. 2008;36(9):3025–30.
22. Yang L, Xia J-F, Gui J. Prediction of protein-protein interactions from protein sequence using local descriptors. *Protein Pept Lett*. 2010;17(9):1085–90.
23. Zhou YZ, Gao Y, Zheng YY. Prediction of protein-protein interactions using local description of amino acid sequence. *Adv Comput Sci Educ Appl*. 2011;202:254–62.
24. You Z-H, Zhu L, Zheng C-H, Yu H-J, Deng S-P, Ji Z. Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set. *BMC Bioinformatics*. 2014;15(15):9.
25. You Z-H, Chan KC, Hu P. Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest. *PLoS ONE*. 2015;10(5):0125811.
26. You Z-H, Li J, Gao X, He Z, Zhu L, Lei Y-K, Ji Z. Detecting protein-protein interactions with a novel matrix-based protein sequence representation and support vector machines. *BioMed Res Int*. 2015;2015:9.
27. Du X, Sun S, Hu C, Yao Y, Yan Y, Zhang Y. DeepPPI: Boosting Prediction of Protein-Protein Interactions with Deep Neural Networks. *J Chem Inf Model*. 2017;57(6):1499–510.
28. Wang J, Zhang L, Jia L, Ren Y, Yu G. Protein-protein interactions prediction using a novel local conjoint triad descriptor of amino acid sequences. *Int J Mol Sci*. 2017;18(11):2373.
29. Sprinzak E, Sattath S, Margalit H. How reliable are experimental protein-protein interaction data? *J Mol Biol*. 2003;327(5):919–23.
30. Ben-Hur A, Noble WS. Choosing negative examples for the prediction of protein-protein interactions. *BMC Bioinformatics*. 2006;7(1):2.
31. Salwinski L, Miller CS, Smith AJ, Pettit FK, Bowie JU, Eisenberg D. The database of interacting proteins: 2004 update. *Nucleic Acids Res*. 2004;32(suppl_1):449–51.
32. Angermueller C, Pärnamaa T, Parts L, Stegle O. Deep learning for computational biology. *Mol Syst Biol*. 2016;12(7):878.
33. Blohm P, Frishman G, Smailowski P, Goebels F, Wachinger B, Ruepp A, Frishman D. Negatome 2.0: a database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis. *Nucleic Acids Res*. 2013;42(D1):396–400.
34. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*. 2006;22(13):1658–9.
35. Consortium U. Update on activities at the Universal Protein Resource (UniProt) in 2013. *Nucleic Acids Res*. 2013;41:D43–D47.
36. Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci*. 1992;89(22):10915–9.
37. Chua HN, Sung W-K, Wong L. Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*. 2006;22(13):1623–30.
38. Wold S, Jonsson J, Sjöström M, Sandberg M, Rännar S. Dna and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least-squares projections to latent structures. *Anal Chim Acta*. 1993;277(2):239–53.
39. Tanford C. Contribution of hydrophobic interactions to the stability of the globular conformation of proteins. *J Am Chem Soc*. 1962;84(22):4240–7.
40. Hopp TP, Woods KR. Prediction of protein antigenic determinants from amino acid sequences. *Proc Natl Acad Sci*. 1981;78(6):3824.
41. Krigbaum W, Komoriya A. Local interactions as a structure determinant for protein molecules. II. *Biochimica et Biophysica Acta (BBA)-Protein Structure*. 1979;576(1):204–28.
42. Grantham R. Amino acid difference formula to help explain protein evolution. *Science*. 1974;185(4154):862–4.
43. Charton M, Charton BI. The structural dependence of amino acid hydrophobicity parameters. *J Theor Biol*. 1982;99(4):629–44.
44. Rose GD, Geselowitz AR, Lesser GJ, Lee RH, Zehfus MH. Hydrophobicity of amino acid residues in globular proteins. *Science*. 1985;229(4716):834–8.
45. Zhou P, Tian FF, Li B, Wu SR, Li ZL. Genetic algorithm-based virtual screening of combinative mode for peptide/protein. *Acta Chim Sin*. 2006;64(7):691–7.
46. Al Shalabi L, Shaaban Z, Kasasbeh B. Data mining: A preprocessing engine. *J Comput Sci*. 2006;2(9):735–9.
47. Hinton G, Deng L, Yu D, Dahl GE, Mohamed A-r, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Proc Mag*. 2012;29(6):82–97.
48. Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. *IEEE Trans Pattern Anal Mach Intell*. 2013;35(8):1798–828.
49. Wang P, Li W, Gao Z, Zhang J, Tang C, Ogunbona PO. Action recognition from depth maps using deep convolutional neural networks. *IEEE Trans Human-Machine Syst*. 2016;46(4):498–509.
50. Wang L, Zhang J, Liu P, Choo K-KR, Huang F. Spectral-spatial multi-feature-based deep learning for hyperspectral remote sensing image classification. *Soft Comput*. 2017;21(1):213–21.
51. Wang P, Li W, Gao Z, Tang C, Ogunbona PO. Depth pooling based large-scale 3-d action recognition with convolutional neural networks. *IEEE Trans Multimed*. 2018;20(5):1051–61.
52. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014. arXiv preprint arXiv:1406.1078.
53. Goldberg Y. Neural network methods for natural language processing. *Synth Lect Hum Lang Technol*. 2017;10(1):1–309.
54. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat Methods*. 2015;12(10):931.
55. Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res*. 2016;26(7):990–9.
56. Min S, Lee B, Yoon S. Deep learning in bioinformatics. *Brief Bioinform*. 2017;18(5):851–69.
57. Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. USA: Omnipress; 2010. p. 807–14.
58. Kingma D, Ba J. Adam: A method for stochastic optimization. *CoRR abs/1412.6980*. 2014.
59. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. *Comput Sci*. 2015;14(7):38–39.
60. Akbani R, Kwek S, Japkowicz N. Applying support vector machines to imbalanced datasets. In: *European Conference on Machine Learning*. Berlin: Springer Berlin Heidelberg; 2004. p. 39–50.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

