Wu *et al. BMC Bioinformatics* 2022, **23**(Suppl 4):560
https://doi.org/10.1186/s12859-022-04771-2

BMC Bioinformatics

RESEARCH Open Access

GCNCPR-ACPs: a novel graph convolution network method for ACPs prediction

Xiujin Wu[1], Wenhua Zeng[1*] and Fan Lin[1,2*]

From The 20th International Conference on Bioinformatics (InCoB 2021)
Kunming, China. 6-8 November 2021

*Correspondence:
whzeng@xmu.edu.cn;
iamafan@xmu.edu.cn

[1] School of Informatics, Xiamen University, Xiamen, Fujian, China
[2] Computational Health Informatics Program, Boston Children's Hospital, Boston, MA, USA

Abstract

Background: Anticancer peptide (ACP) inhibits and kills tumor cells. Research on ACP is of great significance for the development of new drugs, and the prediction of ACPs and non-ACPs is the new hotspot.

Results: We propose a new machine learning-based method named GCNCPR-ACPs (a Graph Convolutional Neural Network Method based on collapse pooling and residual network to predict the ACPs), which automatically and accurately predicts ACPs using residual graph convolution networks, differentiable graph pooling, and features extracted using peptide sequence information extraction. The GCNCPR-ACPs method can effectively capture different levels of node attributes for amino acid node representation learning, GCNCPR-ACPs uses node2vec and one-hot embedding methods to extract initial amino acid features for ACP prediction.

Conclusions: Experimental results of ten-fold cross-validation and independent validation based on different metrics showed that GCNCPR-ACPs significantly outperformed state-of-the-art methods. Specifically, the evaluation indicators of Matthews Correlation Coefficient (MCC) and AUC of our predicator were 69.5% and 90%, respectively, which were 4.3% and 2% higher than those of the other predictors, respectively, in ten-fold cross-validation. And in the independent test, the scores of MCC and SP were 69.6% and 93.9%, respectively, which were 37.6% and 5.5% higher than those of the other predictors, respectively. The overall results showed that the GCNCPR-ACPs method proposed in the current paper can effectively predict ACPs.

Keywords: Anticancer peptide, Graph convolution network, Graph collapse, Graph representation learning, Classification

Background

Cancer is a worldwide disease, and the number of people who die of cancer every year is very high [1, 2]. The major strategy for cancer treatment is traditional chemotherapy. Anticancer chemotherapeutic drugs can effectively treat cancer and kill the cancer cells but they can also kill the healthy cells and cause resistance in cancer cells [3]. Therefore,

more reasonable and effective therapeutic drugs are urgently needed. Anti-microbial peptides (AMPs) [4] are small molecular peptides produced by organisms and can kill certain cancer cells. Anticancer peptides (ACPs) are typically short peptides containing 10–50 amino acids [5, 6], are a subset of anti-microbial peptides with anticancer activity, and can effectively inhibit tumor growth and kill cancer cells by regulating gene expression [7] and mobilizing the immune system [8]. The ACPs can overcome the shortcomings of traditional cancer treatment methods and can kill cancer cells without harming normal cells, thus, becoming one of the most reliable anticancer therapies. Certain experimental methods have been proposed to determine whether a protein has anticancer activity. However, the main disadvantage of wet-lab experiments is that the process is complex, time-consuming, and costly. In contrast, with the development of machine learning and deep learning methods [9, 10], in silico prediction of the ACPs and non-ACPs has the advantages of being less costly, time-efficient, and highly accurate. A growing number of prediction methods have been proposed that guide the experimental screening of candidate ACPs.

In recent years, the application of machine learning and deep learning models to identify ACPs and non-ACPs has become a research hotspot in the field of bioinformatics and computational biology [11]. More and more ACPs have been discovered and validated experimentally [12]. Most of the previous computational methods use the existing databases, extract the features, and then classify the peptides into ACPs and non-ACPs automatically using the feature training model. For example, Vijayakumar et al. proposed a computational method using support-vector machine and protein relatedness measure feature vector [13]. Sequence-based feature extraction methods have been proposed, including amino acid composition, dipeptide composition, and binary pattern, to predict and discover new anticancer peptides [14, 15]. Chen et al. reported a sequence-based predictor called iACP, which was developed by optimizing the g-gap dipeptide components to predict the ACPs [16]. Shahid Akbar et al. proposed an intelligent model, 'iACP-GAEnsC', based on the evolutionary intelligent genetic algorithm, which uses three different discrete feature representation methods to predict the ACPs [17]. Manavalan et al. developed a machine-learning method called MLACP, which used support-vector machine and Random Forest (RF)-based tool to predict the ACPs using the amino acid sequence features [18]. Wei et al. proposed the ACP-FL method, which extracted and learned a set of informative features of the protein from a pool of support-vector machine-based models to identify the ACPs [19]. Another method called QSPred-FL [20] has also been proposed to automatically learn the most discriminative features from the existing feature descriptors in a supervised way to classify the ACPs and non-ACPs. Wei et al. further designed a bioinformatics tool for the ACP prediction called PEPred-Suite [21]. The PEPred-Suite extracted diverse sequence-based features, which could reliably predict different ACPs using RF models.

With the development of more and more ACP-predicting methods, their prediction accuracy has increased. The improved predictive ability of various computational methods will rapidly push forward their applications in cancer therapy. Graph neural network (GNN) is an advanced deep learning model that has been applied for various bioinformatics tasks [22–24], such as link prediction [11], node classification, and community detection. More and more GNN methods [25] are proposed, and their application to

the ACP data can be considered. Most of the ACP data consisted of sequence-based features, and it was then integrated with the other features as the classifier inputs to build the predictive model. However, most of the traditional classification models for ACP prediction mainly regard ACP data as ordered sequence data, ignoring the ACP structure and the relationship between amino acids. If the ACP data was regarded as a kind of structured data, likely graph data, the amino acids would be regarded as nodes and the relationship between amino acids as edges. Then, we could use graph-based methods to deal with ACP data. Moreover, the physicochemical information will be integrated to describe the node attributes in the graph, making the ACPs a topological map composed of amino acid sequences.

In the current study, a graph convolution network (GCN) method based on graph collapse pooling and residual network for predicting ACPs (GCNCPR-ACPs) was used to deal with the graph-based data of ACPs. The graph convolutional neural network is used to extract the graph structure features of amino acids and calculate the graph collapse pooling operator. The graph collapse pooling module is used to aggregate multiple nodes into a large node. After several layers of collapse, the peptide chain composed of amino acids graph is finally collapsed into a large node, and the feature of the collapsed large node is the feature of the whole ACP line. The residual network is used to solve the problem of gradient disappearance caused by the deepening of layers. The main contributions of the current study can be summarized as follows:

1. We proposed a novel GCN-based framework named GCNCPR-ACPs for ACP prediction. To the best of our knowledge, this is the first attempt to adapt graph collapse pooling for ACP prediction.
2. To effectively capture different levels of node attributes for amino acid node representation learning, this is the first attempt to use 4 kinds of node attributes for ACP prediction, containing 2 physicochemical properties of amino acids and 2 others extracted by one-hot embedding and node2vec methods separately.
3. The GCNCPR-ACPs method predicts the ACPs based on multiple properties of the nodes, the diverse characteristics of the ACP graph data, and the novel GCNCPR model.
4. Experiments have been conducted extensively to evaluate the performance of the graph convolutional neural network method based on collapse pooling and residual network.

Results and discussion

In this section, we introduced the evaluation metric used in our experiments; then we presented the evaluation of the performance of the proposed GCNCPR-ACPs model using these evaluation indicators, and discussed the results. Finally, we introduced the experimental setting parameter to verify the effectiveness of the model.

Evaluation metric

For performance evaluation, several machine learning metrics are widely used in prediction methods. They were used to verify the effectiveness of our model, including

sensitivity (SE), specificity (SP), accuracy (ACC), Matthew's correlation coefficient (MCC), and area under the curve (AUC). The formulas of the five metrics used are as follows:

$$\begin{cases} Sensitive = recall = \frac{TP}{TN+FN} = \frac{TP}{P} \times 100\% \\ Specificity = \frac{TN}{TN+FP} = \frac{TN}{N} \times 100\% \\ Precision = \frac{TP}{TP+FP} \times 100\% \\ Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{TP+TN}{P+N} \times 100\% \\ MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FN)(TN+FP)}} \times 100\% \\ F1 = \frac{2 \times recall \times precision}{recall+precision} \end{cases} \tag{1}$$

True positive (TP) indicates the number of true ACP samples that are predicted correctly. False positive (FP) indicates the number of ACP samples with false prediction, that is, the non-ACP samples that are classified as ACP samples by the classifier. True negative (TN) indicates the number of non-ACP samples that are predicted correctly. False negative (FN) indicates the number of non-ACP samples with false prediction. AUC measures the overall performance of the predictor. The higher the AUC, the better is the performance of the model.

Results of ten-fold cross-validation

To validate the predictive performance of the proposed GCNCPR-ACPs, we compared its performance with that of several existing predictors, including iACP [16], ACPred-FL [19], PEPred-Suite [21], ACPred-Fuse [26], AntiCP_ACC [13], AntiCP_DC [13], and Hajisharifi's [27]. The cross-validation results are presented in Table 1. It was observed that GCNCPR-ACPs outperformed, since the scores of all of its evaluation indicators were the highest, especially, the Matthews Correlation Coefficient (MCC) and specificity (SP), which were 88% and 100%, respectively.

Results of independent test

To validate the robustness of the proposed GCNCPR-ACPs, we compared its performance with that of several existing predictors. In the independent test, the scores of MCC and SP of our model were 69.6% and 93.9%, respectively, and were 37.6% and 5.5%

Table 1 Cross-validation results of the GCNCPR-ACPs and other methods

Methods	SE	SP	ACC	MCC	AUC
iACP	57.2	84.0	70.6	42.8	80.9
ACPred-FL	71.6	84.4	78.0	56.5	84.6
PEPred-Suite	72.8	<u>88.0</u>	80.4	61.5	86.0
ACPred-Fuse	<u>77.2</u>	87.6	<u>82.4</u>	<u>65.2</u>	<u>88.2</u>
AntiCP_ACC	66.8	78.4	72.6	45.5	82.4
AntiCP_DC	71.6	77.6	74.6	49.3	82.5
Hajisharifi's	67.2	83.6	75.4	51.5	83.1
GCNCPR-ACPs	**81.5**	**88.1**	**84.6**	**69.5**	**90.0**

The highest scores are marked in bold and the second-highest are underlined

Table 2 Independent test results of the proposed predictor and the existing predictors

Methods	SE	SP	ACC	MCC	AUC
iACP	54.9	88.8	87.7	22.6	76.1
ACPred-FL	69.5	85.8	85.3	25.9	85.1
PEPred-Suite	68.3	90.6	**89.9**	32.0	86.1
ACPred-Fuse	72	89.5	89	32.0	**86.8**
AntiCP_ACC	68.3	88.5	87.9	28.8	85.3
AntiCP_DC	68.3	82.6	82.2	22.3	83.0
Hajisharifi's	69.5	88.4	87.9	29.2	85.5
GCNCPR-ACPs	**74.4**	**93.9**	84.1	**69.6**	84.1

The highest scores are marked in bold and the second-highest are underlined

higher than that of the other predictors, respectively. The independent test results are presented in Table 2.

Parameter analysis

Several important parameters influence the performance of our model, such as the learning rate, the number of layers of the GCN, and the assign ratio. In the current section, we present the results of the sensitivity analysis of these parameters. In our model, the training epoch was set to 1000. The hidden dimension and the output dimension were 64.

We then evaluated our model by choosing the learning rate from 0.1, 0.01, and 0.001. Figure 1A, B show that as the learning rate varies, the performance gradually increases initially and then decreases, where a learning rate of 0.01 gives the best performance. As shown in Fig. 1C, D, we observed that our model was slightly influenced by the number of layers. After increasing the number of layers from 5 to 10 with a step value of 1, we observed that our model was relatively robust since ACC and F1 were quite stable. The assign ratio was chosen from 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. As shown in Fig. 1E, F, when the assign ratio was larger, the performance of independent test results was better.

Ablation experiments

We compared our experiments with ablation experiments. The training dataset was divided using a ratio of 9:1, that is, into 450 training samples and 50 verification samples for the ablation experiment. The ablation experiment results are presented in Table 3. We observed that our experiment was mainly composed of three modules: n-layer stacking graph convolution neural network G(X), graph collapse pooling module D(X), and residual network R(X). We observed that the results of our model were the best, the results of G(X) + D(X) were second-to-best, and the results of G(X) were the worst.

Training ratio comparison experiments

For comparison, the setting of training data set is the same as that of Wei's articles, and they all focus on anticancer peptides prediction [28, 29]. The training data was balanced, i.e., there were 250 positive samples and 250 negative samples. We also tried different ratios of positives-to-negatives (e.g., 1:1, 1:5, and 1:10) to further test the performance

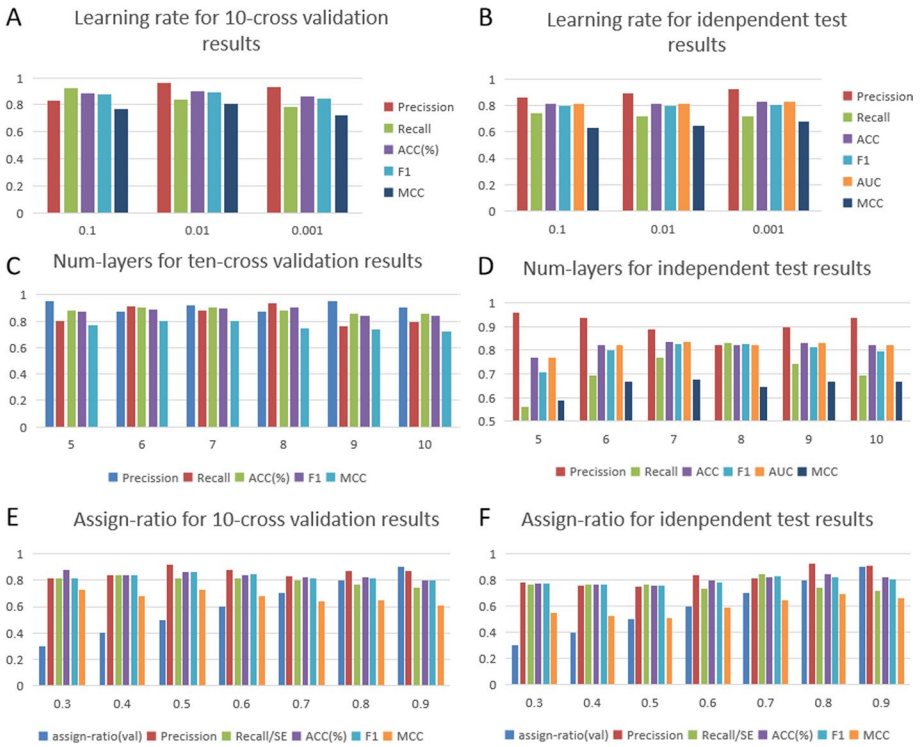

Fig. 1 The comparison of the performance of our proposed GCNCPR-ACPs with other state-of-the-art predictors. **A** The effects of the learning rate on the performance of the tenfold cross-validation results of the proposed GCNCPR-ACPs and the existing prediction models. **B** The effects of the learning rate on the performance of the independent test results of the proposed GCNCPR-ACPs and the existing prediction models. **C** The effects of the number of layers on the performance of the tenfold cross-validation results of the proposed GCNCPR-ACPs and the existing prediction models. **D** The effects of the number of layers on the performance of the independent test results of the proposed GCNCPR-ACPs and the existing prediction models. **E** The effects of the assign ratio on the performance of the tenfold cross-validation results of the proposed GCNCPR-ACPs and the existing prediction models. **F** The effects of the assign ratio on the performance of the independent test results of the proposed GCNCPR-ACPs and existing prediction models

Table 3 Results of the ablation experiments

Model	Precision	Recall	ACC	F1	MCC
G(X)	0.76	0.75	0.76	0.76	0.45
G(X) + R(X)	0.83	0.83	0.84	0.83	0.67
G(X) + D(X)	0.83	0.74	0.84	0.78	0.62
G(X) + D(X) + R(X)	0.84	0.86	0.88	0.85	0.70

Table 4 Results of the training ratio comparison experiments

Ratio	Precision	Recall	ACC	F1	MCC
1:1	0.84	0.86	0.88	0.85	0.70
1:5	0.61	0.96	0.70	0.75	0.49
1:10	0.47	1.0	0.54	0.63	0.38

of the proposed model. And the training ratio comparison experiments results are presented in Table 4. We observed that the training ratio of 1:1 worked best, the training ratio of 1:5 worked second-to-best, and the training ratio of 1:10 worked the worst. As the proportion of positive and negative training samples changes from 1:1 to 1:10, the experimental results show that with the ratio between positive and negative samples decreases, the recall increases, but all of the other values decrease.

Conclusion

Here, we proposed a new prediction model called GCNCPR-ACPs. It is a powerful bioinformatics tool to predict anticancer peptides using GCN and graph collapse pooling and residual network model. The advantage of GCNCPR-ACPs is that it can effectively construct the anticancer peptide map. It can extract useful features from graph data, including node attributes, line attributes, icon labels, node labels, and adjacency matrix. GCNCPR-ACPs model is novel and it mainly includes the following modules: graph differentiable pool module D(X), stacked graph convolution neural network module G(X), and residual network module R(X). The experimental results of ten-fold cross-validation and independent test show that the proposed predictor can more effectively classify the ACPs and non-ACPs. The effective predicting ability of the model will accelerate its application in cancer treatment.

Methods

In the current section, we will introduce the overall framework of our model. The steps of the model are shown in Fig. 2. Step 1: the amino acid sequences of ACPs are collected from three protein datasets to form our training and test datasets. Step 2: the ACP

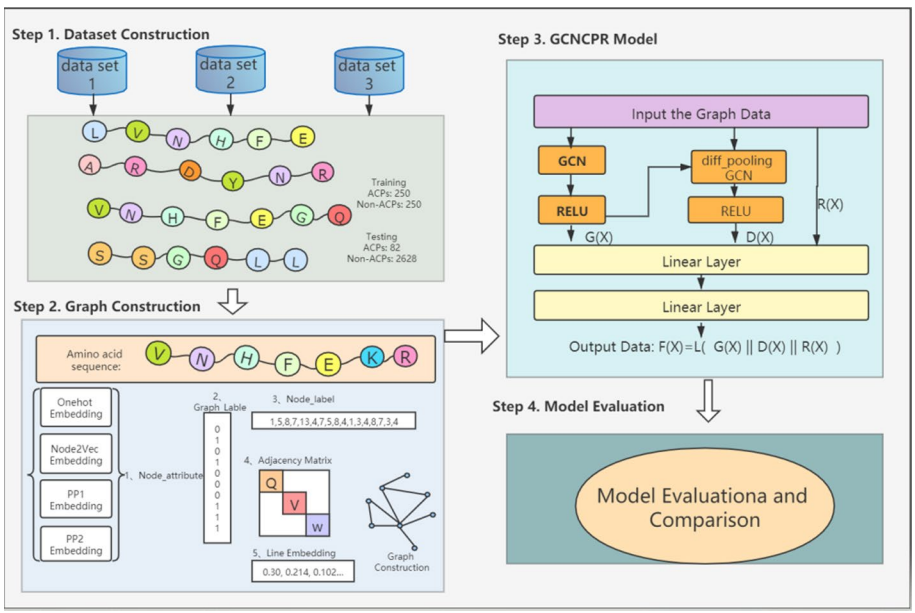

Fig. 2 The overview of the GCNCPR-ACPs predictor. Step 1, data construction: the ACP datasets are prepared to obtain the training and test datasets. Step 2, graph construction: the ACPs chains are used to construct the graphs using the amino acids as nodes. Step 3, GCNCPR model: the graph data is used as input for the GCNCPR model and to classify the ACP chain. Step 4, model evaluation: the classification results of our model are evaluated and compared with those of the other models

chains will be used to construct the graphs with the amino acids as nodes. The graph data properties contain the graph labels, node labels, adjacency matrix, node attributes, and line embedding data. Step 3: the GCNCPR model will be introduced in detail. Step 4: The training and test results of our model will be discussed and compared with those of other models.

Materials

The peptides with anticancer activity are called anticancer peptides (ACPs), and they are regarded as positive samples. On the contrary, the peptide samples without anticancer activity are called non-anticancer peptides (non-ACPs) and are considered negative peptides. In the current chapter, the training and independent test datasets of ACPs are introduced.

In the current study, we used the datasets used by Wei et al. [19]. There were 332 ACP samples and 2878 non-ACP samples in the dataset. In the report by Wei et al. [19], the training datasets contained 250 ACPs and 250 non-ACPs. The rest of the dataset included the remaining 82 ACP samples (positive samples) and 2628 non-ACP samples (negative samples).

Graph construction

The ACPs are composed of amino acid sequences. According to graph neural network theory, the amino acids are regarded as vertices (V) and the links between amino acids as edges (E). The ACP protein chain, which is composed of amino acid sequences, is used to construct the ACP graph neural network (G). The ACP graph network (G) possesses the graph data properties, such as node attributes, line attributes, graph labels, node labels, and the adjacency matrix A. $A_{n\times n} \in [0, 1]$. The number "1" denotes that the two amino acids are connected and "0" denotes that there is no edge between them. The graph labels are the labels of the ACP chains that are composed of several amino acids. The graph label value of "1" denotes non-ACPs (negative samples), and "0" denotes ACPs (positive samples). The line attribute, as proposed by Wei et al. [19], represents the characteristic of one ACP chain. The amino acid nodes in the graph are represented by 20 English letters. The amino acid nodes are divided into 20 categories, and each amino acid node has a category identifier from 1 to 20, called node labels. The node attributes are the features of amino acid nodes in the graph, obtained using four embedding methods—one-hot embedding method, node2vec embedding method, and two kinds of physicochemical property-embedding methods.

One-hot embedding

One-hot embedding is an effective encoding method expressed using binary vectors. Only one bit is valid at any time, and other positions are set to 0. The primary structural information of ACP protein is mainly composed of 20 common amino acids. Each of these 20 amino acids is represented by a single English letter (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, and Y). Therefore, each amino acid node in the graph is represented as a 20-dimensional feature vector by one-hot embedding.

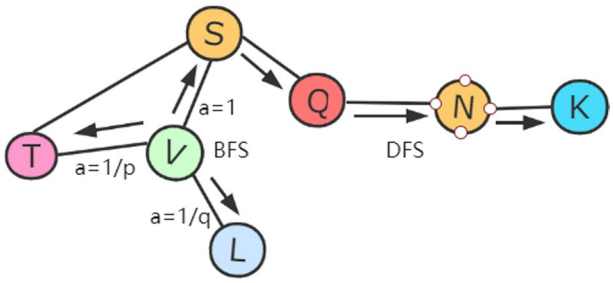

Fig. 3 Node2vec embedding method. Shows the BFS and DFS search strategies starting from node V. In the current study, we used the DFS sampling method to sample the amino acid nodes

Table 5 The categorization of the standard amino acid nodes based on ten physicochemical properties

Id	Physicochemical properties	Amino acids
1	Aromatic	F, Y, W, H
2	Negative	D, E
3	Positive	K, H, R
4	Polar	N, Q, S, D, E, C, T, K, R, H, Y, W
5	Hydrophobic	A, G, C, T, I, V, L, K, H, F, Y, W, M
6	Aliphatic	I, V, L
7	Tiny	A, S, G, C
8	Charged	K, H, R, D, E
9	Small	P, N, D, T, C, A, G, S, V
10	Proline	P

Node2vec embedding

Node2vec embedding considers the distance between two nodes. We used the node-2vec embedding method to encode amino acid nodes. Node2vec uses random walk to get the nearest neighbor information of the vertices. The Node2vec embedding method mainly uses random walk to sample the node sequence. When it comes to random walk sampling, there are two main kinds of graph walking—depth-first sampling (DFS) and breadth-first sampling (BFS), shown in Fig. 3.

Physicochemical property embedding

Although the above embedding methods can consider the characteristics of the nodes in the graph and evaluate the distance between two nodes, the amino acid nodes have their own physical and chemical sense. To capture the physical and chemical sense of the amino acid node, Dou et al. [30] divided the 20 amino acids into 10 groups according to the physicochemical properties of these amino acids. The amino acids often have more than one property. As shown in Table 5, the ten properties are binary coded to form a 10-dimensional embedding feature vector.

The second embedding feature method based on the physicochemical properties of amino acids is shown below. According to the Composition, Transition, and Distribution (CTD) [31] of the amino acid attributes, the standard amino acids can be

Table 6 The categorization of the standard amino acid nodes based on the seven physicochemical properties

Physicochemical properties	Group 1	Group 2	Group 3
Hydrophobicity	A, C, F, G, H, I, L, M, N, P, Q, S, T, V, W, Y	D, E	K, R
Normalized	C, F, I, L, M, V, W	A, G, H, P, S, T, Y	D, E, K, N, Q, R
Polarity	A, C, D, G, P, S, T	E, I, L, N, Q, V	F, H, K, M, R, W, Y
Polarizibility	C, F, I, L, M, V, W, Y	A, G, P, S, T	D, E, H, K, N, Q, R
Charge	A, D, G, S, T	C, E, I, L, N, P, Q, V	F, H, K, M, R, W, Y
Secondary structures	D, G, N, P, S	A, E, H, K, L, M, Q, R	C, F, I, T, V, W, Y
Solvent accessibility	A, C, F, G, I, L, V, W	H, M, P, S, T, Y	D, E, K, N, R, Q

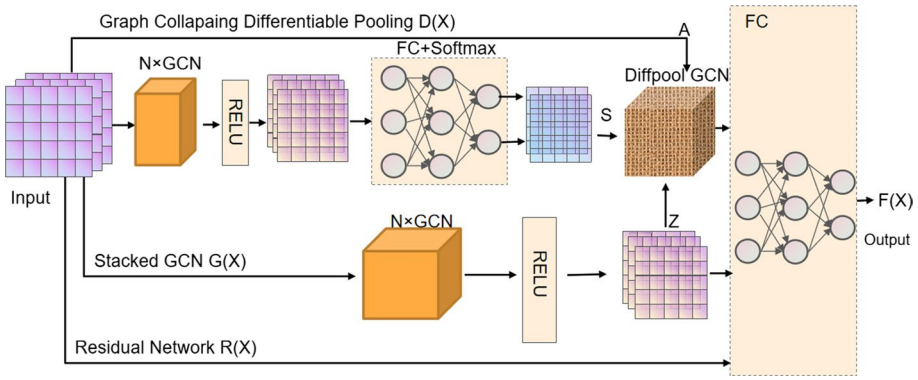

Fig. 4 The GCNCPR module. The GCNCPR model mainly includes the following modules—the graph collapse differentiable pooling module D(X), the stacked graph convolution neural network module G(X), and the residual network module R(X)

categorized using seven physicochemical properties as shown in Table 6. Each physicochemical property has three groups. Therefore, a total of 21 embedding features are used to characterize each amino acid node.

GCNCPR model

In the current section, we introduced a flexible model using an n-layered GCN for supervised learning of a graph (Fig. 4). Firstly, the graph data is input into the GCNCPR module. The stacked graph convolution neural network module G(X) is used to extract the features of ACPs, the graph collapse differentiable pooling module D(X) is used to extract the ACP chain features, and the residual network R(X) is used to prevent the gradient disappearing problem. We combined these ACP features and input them into the FC (full connection) module for dimension reduction to get the final output data. Then, the ACP chains were classified into ACPs and non-ACPs. We then concatenated the learned representations as the ACP chain features as below:

$$F(X) = L(G(X) \| D(X) \| R(X)) \tag{2}$$

where || denotes the operation of vector concatenation, such as mean or addition, and L(.) represents the function that passes through two linear fully connected layers.

Stacked graph convolution network G(X)

The stacked graph convolution network is a multi-layered stack of GCN, which is used to learn the representation of amino acid nodes in the ACP graph. The theoretical formula of a single layer of GCN is as follows:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{3}$$

where $H^{(l)} \in R^{n\times d}$, n is the number of nodes in the graph, and each node is represented by a d-dimensional feature vector called node attribute. The input feature of layer l is $H^{(l)}$. A is an adjacency matrix. $\tilde{A}$ is the adjacency matrix with self-connections,$\tilde{A} = A + I_N$.$\tilde{D}$ is the degree matrix, $\tilde{D}_{ij} = \sum_j \tilde{A}_{ij}$.$W^{(l)}$ is the trainable parameter, and σ is the corresponding activation function, such as ReLU$(\cdot)$ or $max(0, \cdot)$. $H^{(l+1)}$ is the output feature data of the $(l+1)$th layer graph. Equation (2) is the final form of GCN.

Graph collapsing differentiable pooling module D(X)

Differential pooling is an algorithm that combines the graph collapse process with GCN for graph representation learning [32]. Firstly, the graph data with amino acid node features $H^{(l)}$ and adjacency matrix $A^{(l)}$ of the nodes in layer l were input into the differential pooling GCN module. According to the GCN formula above, after three layers of GCN, we get the node feature expression $Z^{(l)}$. The softmax is used to make a full connection, the connection structure of the lower layer supernodes is obtained, and the matrix allocator $S^{(l)}$ is learned, whose value represents the probability that the nodes are assigned to any cluster. The matrix allocator S is also called the graph collapse operator. The closer the probability value, the more likely are the nodes to be assigned to the same cluster.

$$Z^{(l)} = H^{(l+1)} = G_{l,embed}\left(A^{(l)}, H^{(l)}\right) \tag{4}$$

$$S^{(l)} = softmax\left(G_{l,pool}\left(A^{(l)}, H^{(l)}\right)\right) \tag{5}$$

$G_{l,embed}$ and $G_{l,pool}$ are two independent GCN layers. Their inputs are the same, which are the amino acid node features $H^{(l)}$ and adjacency matrix $A^{(l)}$ of the nodes. However, their parameters and learning purposes are different. For the cluster allocation matrix S in the last layer, we need to directly fix it into a matrix, which is filled by "1", because we need to collapse the graph into a super large node, to obtain the global representation of the graph.

$$Z^{(l+1)} = {S^{(l)}}^T Z^{(l)} \tag{6}$$

$$A^{(l+1)} = {S^{(l)}}^T A^{(l)} S^{(l)} \tag{7}$$

where $A^{(l)} \in R^{n^{(l)}\times n^{(l)}}$, $S^{(l)} \in R^{n^{(l)}\times n^{(l+1)}}$, $n^{(l)}$ represents the number of nodes in layer l, $A^{(l+1)}$ and $Z^{(l+1)}$ denotes the number of nodes (clusters) in layer $(l+1)$.

Residual network module R(X)

It is known that increasing the depth of the network improves the performance of the network. The performance of a shallow neural network is often poor than that of a deep

neural network, but if we simply increase the depth, it will lead to gradient dispersion or gradient explosion. Moreover, with the increase in network layers, the accuracy of the training set does not increase further or even decreases, resulting in degradation. To solve this problem, we use the residual network into our model.

Loss function

Cross entropy loss function was used to optimize the model training loss. In the case of two classifications, the final prediction results of the model are only two cases, either 0 or 1. For each category, the probability of our prediction is p and $1-p$. The specific formula used was as follows:

$$Loss = -\frac{1}{N}\sum_{i} -[y_i log(p_i) + (1-y_i)\cdot log(1-p_i)] \tag{8}$$

where, y_i represents the real label of the ith sample. And p_i represents the probability of the ith sample which is predicted to be a positive class. N represents the number of all of the ACP samples.

Abbreviations

ACP	Anticancer peptide
GCN	Graph convolution network
GCNCPR-ACPs	The graph convolutional neural network method based on collapse pooling and residual network (GCNCPR) for ACPs prediction
CTD	Composition, transition, and distribution

Acknowledgements

The authors thank the reviewers for their suggestions that helped improve the manuscript and MJEditor (www.mjeditor.com) for its linguistic assistance during the preparation of this manuscript. The authors also thanks to Mr Wei Leyi and Professor Min Wu for their suggestions on the manuscript.

About this supplement

This article has been published as part of BMC Bioinformatics Volume 23 Supplement 4, 2022: The 20th International Conference on Bioinformatics (InCoB 2021). The full contents of the supplement are available online at https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-23-supplement-4.

Author contributions

FL and WZ initialized the research project and designed the experiments; XW designed software, performed the experiments, and wrote the paper; all authors reviewed the manuscript. All authors read and approved the final manuscript.

Funding

Publication costs are funded by the Natural Science Foundation of Fujian Province of China (Grant No. 2020J01435, No. 2019J01846, No. 2021J011169). The funders had no role in the design of the study, collection, analysis or interpretation of the data, the writing of the manuscript or the decision to submit the manuscript for publication.

Availability of data and materials

ACP500 and ACP164 datasets are available at https://github.com/hengenggg/gcncpr.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 27 May 2022 Accepted: 31 May 2022
Published: 23 December 2022

References

1. Ferlay J, Shin HR, Bray F, Forman D, Mathers C, Parkin DM. Estimates of worldwide burden of cancer in 2008: GLOBOCAN 2008. Int J Cancer. 2010;127(12):2893–917.
2. Jemal A, Siegel MR, Ward E, Hao Y, Xu J, Murray MT, Thun MJ. Cancer statistics. CA Cancer J Clin. 2008;58(2):71–96.
3. Caitriona H, Sandra VS, Longley DB, Johnston PG. Cancer drug resistance: an evolving paradigm. Nat Rev Cancer. 2013;13(10):714–26.
4. Xiao X, Wang P, Lin WZ, Jia JH, Chou KC. iAMP-2L: a two-level multi-label classifier for identifying antimicrobial peptides and their functional types. Anal Biochem. 2013;436(2):168–77.
5. Barras D, Widmann C. Promises of apoptosis-inducing peptides in cancer therapeutics. Curr Pharm Biotechnol. 2011;12:1153.
6. J Boohaker R, W Lee M, Vishnubhotla P, LM Perez J, R Khaled A. The use of therapeutic peptides to target and to kill cancer cells. Curr Med Chem. 2012;19(22):3794–804.
7. Shi SL, Wang YY, Liang Y, Li QF. Effects of tachyplesin and n-sodium butyrate on proliferation and gene expression of human gastric adenocarcinoma cell line BGC-823. World J Gastroenterol. 2006;12(11):1694–8.
8. Chen J, Xu XM, Underhill CB, Yang S, Wang L, Chen Y, Hong S, Creswell K, Zhang L. Tachyplesin activates the classic complement pathway to kill tumor cells. Cancer Res. 2005;65(11):4614.
9. Chen M, Li Y, Zhou X. CoNet: co-occurrence neural networks for recommendation. Future Gener Comput Syst. 2021;124:308–14.
10. Chen M, Zhou X. DeepRank: learning to rank with neural networks for recommendation. Knowl Based Syst. 2020;209:106478.
11. Ata SK, Wu M, Fang Y, Le OY, Li XL. Recent advances in network-based methods for disease gene prediction. Brief Bioinform. 2021;22(4):bbaa303.
12. Tyagi A, Tuknait A, Anand P, Gupta S, Sharma M, Mathur D, Joshi A, Singh S, Gautam A, Raghava GPS. CancerPPD: a database of anticancer peptides and proteins. Nucl Acids Res. 2015;43(Database issue):837–43.
13. Vijayakumar S, Ptv L. ACPP: a web server for prediction and design of anti-cancer peptides. Int J Peptide Res Therap. 2015;21(1):99–106.
14. Tyagi A, Kapoor P, Kumar R, Chaudhary K, Gautam A, Raghava GP. In silico models for designing and discovering novel anticancer peptides. Sci Rep. 2013;3(10):2984.
15. Zhang J, Ju Y, Lu H, Xuan P, Zou Q. Accurate identification of cancerlectins through hybrid machine learning technology. Int J Genomics. 2016;2016(4):1–11.
16. Chen W, Hui D, Feng P, Hao L, Kuo-Chen C. iACP: a sequence-based tool for identifying anticancer peptides. Oncotarget. 2016;7(13):16895–909.
17. Akbar S, Hayat M, Iqbal M, Jan MA. iACP-GAEnsC: evolutionary genetic algorithm based ensemble classification of anticancer peptides by utilizing hybrid feature space. Artif Intell Med. 2017;79:62–70.
18. Manavalan B, Basith S, Shin TH, Sun C, Kim MO, Lee G. MLACP: machine-learning-based prediction of anticancer peptides. Oncotarget. 2017;8(44):77121.
19. Wei L, Zhou C, Chen H, Song J, Su R. ACPred-FL: a sequence-based predictor based on effective feature representation to improve the prediction of anti-cancer peptides. Bioinformatics. 2018;34(23):4007–16.
20. Wei L, Hu J, Li F, Song J, Su R, Zou Q. Comparative analysis and prediction of quorum-sensing peptides using feature representation learning and machine learning algorithms. Brief Bioinform. 2018;21:106–19.
21. Wei L, Zhou C, Su R, Zou Q, Hancock J. PEPred-Suite: improved and robust prediction of therapeutic peptides using adaptive feature representation learning. Bioinformatics. 2019;35:4272–80.
22. Sun M, Zhao S, Coryandar G, Olivier E, Zhou J, Wang F. Graph convolutional networks for computational drug development and discovery. Brief Bioinform. 2019;21:919–35.
23. Cai R, Chen X, Fang Y, Wu M, Hao Y. Dual-dropout graph convolutional network for predicting synthetic lethality in human cancers. Bioinformatics. 2020;36:4458–65.
24. Long Y, Wu M, Keong KC, Luo J, Li X. Predicting human microbe–drug associations via graph convolutional network with conditional random field. Bioinformatics. 2020;36(19):4918–27.
25. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst. 2020;32:4–24.
26. Rao B, Zhou C, Zhang G, Su R, Wei L. ACPred-Fuse: fusing multi-view information improves the prediction of anticancer peptides. Brief Bioinform. 2020;21(5):1846–55.
27. Hajisharifi Z, Piryaiee M, MohammadBeigi M, Mohabatkar H. Predicting anticancer peptides with Chou's pseudo amino acid composition and investigating their mutagenicity via Ames test. J Theor Biol. 2014;341:34–40.
28. Wei L, Hu J, Li F, Song J, Su R, Zou Q. Comparative analysis and prediction of quorum-sensing peptides using feature representation learning and machine learning algorithms. Brief Bioinform. 2020;21(1):106–19.
29. Su R, Hu J, Zou Q, Balachandran M, Wei L. Empirical comparison and analysis of web-based cell-penetrating peptide prediction tools. Brief Bioinform. 2020;21(2):408–20.
30. Dou Y, Yao B, Zhang C. PhosphoSVM: prediction of phosphorylation sites by integrating various protein sequence attributes with a support vector machine. Amino Acids. 2014;46(6):1459–69.
31. Govindan N. Composition, transition and distribution (CTD): a dynamic feature for predictions based on hierarchical structure of cellular sorting. In: India Conference. 2012.
32. Ying Z, You J, Morris C, Ren X, Hamilton W, Leskovec J. Hierarchical graph representation learning with differentiable pooling. Neural Inf Process Syst. 2018;31:4805–15.

Publisher's Note