scientific reports

Check for updates

OPEN

Root-associated protein prediction using a protein large language model and hypergraph convolutional networks

Lei Chen[1]✉, Xingyu Xun[1] & Bo Zhou[2]

Plant root-associated proteins promote plant growth and enhance stress tolerance. They participate in signaling and plant growth regulation. It is clear that they play key roles in plant growth, development and environmental adaptation. At present, the root-associated proteins have not been fully discovered. It is essential to identify latent root-associated proteins. Traditional methods (proteomic analysis, transcriptome and expression analysis) for determining root-associated proteins are highly relied on the data generated by biochemical experiments, which are always expensive and time-consuming. On the other hand, the current computational models show weak ability, providing great spaces for improvement. In this study, we propose a new computational model, Hypergraph-Root, for predicting root-associated proteins. The model employed several feature types to represent proteins, which were derived from proteins BLOSUM62 and position-specific scoring matrices as well as by a protein language model. These features were improved by hypergraph convolutional network and multi-head attention. The final predicted result was yielded by a fully connected layer. The model yielded high performance with AUC about 0.9 on training and independent datasets. It had evident advantages compared with existing models. Some additional tests were conducted to prove the rationality of the model's structure.

Keywords Protein classification, ProtT5, BLOSUM62 matrix, Position-specific scoring matrix, Hypergraph, Deep learning

The plant root system is the main organ for water and mineral uptake, which is crucial for plant growth and development[1]. Root-associated proteins play multiple roles in this process, which not only promote root growth and development, enhance plant stress tolerance, but also participate in the regulation of signaling and growth-regulating mechanisms in the plant and interact with soil microbes[2]. In agricultural production, root-associated proteins have an important impact on the growing environment and yield quality of crops through their direct involvement in plant growth and development, as well as their indirect influence on the soil microbial community, which is a key factor in improving crop resilience and productivity[3].

Among the traditional biological approaches, proteomic analysis, and transcriptome and expression analysis are the two main techniques for identifying root-associated proteins[4]. Proteomics analysis mainly relies on mass spectrometry to identify and compare protein expression differences in different samples, while transcriptome sequencing combined with real-time quantitative PCR (qRT-PCR) is used to probe the expression patterns of specific genes under different environmental conditions. Of course, these two approaches can yield reliable references for studying the functions and regulatory mechanisms of root-associated proteins. Although both techniques are valuable in root-associated protein research at the biomolecular level, they also have some drawbacks, such as high cost, high complexity of data analysis, technical limitations, sample handling limitations, and problems with reproducibility and accuracy of results. Therefore, it is still necessary to develop accurate and reliable computational methods to predict root-associated proteins.

In recent years, machine learning methods have been widely applied to tackle protein-related problems. They can deeply analyze known data and mine hidden associations, thereby learning special patterns for making predictions. Generally, the execution of machine learning methods must be supported by a large amount of data. For the prediction of root-associated proteins, the public RGPDB database[5] collects many root-associated

[1]College of Information Engineering, Shanghai Maritime University, Shanghai, China. [2]School of Basic Medical Sciences, Shanghai University of Medicine and Health Sciences, Shanghai 201318, China. ✉email: lchen@shmtu.edu.cn

genes, providing a strong support for building machine learning-based models. It is known that a protein's final localization and function are directly determined by intrinsic signals encoded in its amino acid sequence, such as signal peptides and transmembrane domains[6,7]. Crucially, root proteomic studies consistently demonstrate that root tissues specifically enrich proteins bearing these sequence features (e.g., plasma membrane-localized transporters and receptor kinases), which are directly responsible for root-specific functions, like nutrient uptake and stress response[8,9]. The special sequence patterns that are associated with root-related biological processes are waiting for exploration. On the other hand, protein sequences are always the first-hand materials to investigate protein-related problems because they are easily to obtain. The models based on protein sequence only always have wide applications. Thus, it is feasible and necessary to identify root-associated proteins only using their sequence information. To date, two machine learning-based models have been proposed, which all adopted protein sequence information. Kumar et al. first designed a machine learning-based model, named SVM-Root[10], for the prediction of root-associated proteins. This model adopted five feature types derived from protein sequences and employed support vector machine (SVM) as the prediction engine. However, its performance was not high. The accuracies on training and test datasets were all lower than 0.75. Later, the second model, named Graph-Root[11], employed more protein features, such as network features, domain features, as well as deep learning algorithms, including graph convolutional network (GCN) and multi-head attention. This model was superior to SVM-Root. However, it was still not efficient enough. Its accuracies on training and test datasets were between 0.75 and 0.80. Evidently, there still exist great spaces for improvement. Existing two models have evident limitations. The first model (SVM-Root) adopted traditional machine learning algorithms, which cannot fully mine the associations between features and root-associated proteins. The second model (Graph-Root) improved the SVM-Root by employing more information of proteins and deep learning algorithms (GCN and multi-head attention). The GCN can capture the relationships between amino acids in one protein sequence and refine protein features at amino acid level. However, GCN can only capture binary relationships between amino acids. The complex relationships beyond such relationships cannot be captured by GCN as GCN uses a general graph as an input, which does not contain the complex relationships. The hypergraph is a generalized version of graph, which can contain more complex relationships between nodes as more than two nodes can comprise one hyperedge. The employment of hypergraph to build the model for identifying root-associated proteins can help the model to make use of the complex relationships between amino acids, thereby enhancing model's performance. On the other hand, the protein language model (pLM) provides new protein representations, which may also be useful to identify root-associated proteins.

In this study, a new computational model was designed to predict root-associated proteins, which was called Hypergraph-Root. This model adopted two general feature types derived from protein sequences, say BLOSUM62 and Position-Specific Scoring Matrix (PSSM) features. It also employed the features yielded by a pLM, ProtT5, which contain high-level information hidden in protein sequences. In addition, the hypergraph was employed for each protein to represent complex relationships between amino acids in this protein sequence and the hypergraph convolutional network (HGCN)[12] was applied to above features and hypergraphs to yield high-order features. After the high-order features were processed by a multi-head attention, the fully connected layer (FCL) was designed to make predictions. The cross-validation on training dataset and the independent test shown that the accuracies were higher than 0.83, which exceeded the accuracies generated by SVM-ROOT and Graph-ROOT. Furthermore, we also conducted some tests to prove the reasonability of the model's structure.

Materials and methods

Dataset

The root-associated proteins were obtained from one previous study[11]. These proteins were original extracted from the RGPDB database (http://sysbio.unl.edu/RGPDB/, assessed on 20 March 2023)[5], a public database collecting more than 1200 candidates of root-associated genes and their corresponding promoter sequences, including 592, 363, and 400 genes for maize, sorghum, and soybean, respectively. These genes were identified by analyzing multiple types of omics datasets for maize, soybean, and sorghum, including tissue transcriptomic and proteomic data. After mapping above genes to STRING database (https://cn.string-db.org/, version 11.5)[13] and Ensembl Genomes (https://www.ensemblgenomes.org, accessed on 10 April 2023)[14], 1259 root-associated proteins were obtained. These proteins were termed as positive samples. The purpose of this study was to design a computational method for identifying root-associated proteins. To this end, we further employed negative samples, which were also retrieved from the previous study[11], including 41,538 non-root-associated proteins. These proteins were downloaded from UniProt (https://www.uniprot.org/, Release 2023_01)[15]. All above proteins constituted the initial dataset of this study.

To further construct a well-defined dataset, all proteins were processed by the following two steps: (1) Proteins with sequence length larger than 1000 were removed; (2) Homologous proteins were also excluded using CD-HIT (with cutoff 0.4)[16]. As a result, 525 root-associated proteins and 9260 non-root-associated proteins were retained. Above root-associated proteins were randomly divided into two sets, denoted by S_{tr}^P and S_{te}^P. The first set S_{tr}^P contained 90% root-associated proteins and the remaining 10% root-associated proteins constituted the second set S_{te}^P. The same operation was performed on non-root-associated proteins, yielding two sets, denoted as S_{tr}^N and S_{te}^N. Generally, proteins in S_{tr}^P and S_{tr}^N can be combined to train the model. However, proteins in S_{tr}^N was much more than those in S_{tr}^P. The model trained on such imbalanced dataset may produce bias. Thus, we randomly selected non-root-associated proteins from S_{tr}^N, which were as many as root-associated proteins in S_{tr}^P. Their combination constituted one training dataset. As the selection of proteins in S_{tr}^N may influence model's performance, above procedures were executed 50 times, yielding 50 training datasets, denoted as $S_{tr}^1, S_{tr}^2, \ldots, S_{tr}^{50}$. Furthermore, we constructed two test datasets. The first test dataset, denoted by S_{te}^1, contained all proteins in S_{te}^P and S_{te}^N, that is $S_{te}^1 = S_{te}^P \cup S_{te}^N$. Clearly, this test dataset was imbalanced as non-root-associated proteins were much more than root-associated proteins. Thus, we called this test dataset as

imbalanced test dataset. In addition, we also constructed a balanced test dataset, denoted by S_{te}^2. This test dataset contained all proteins in S_{te}^P and randomly selected non-root-associated proteins in S_{te}^N, which were as many as proteins in S_{te}^P. The model built on the training datasets will be applied to the test datasets for evaluating its generalization ability.

Original protein feature extraction

Traditionally, the accuracy of samples' features can directly influence the models' performance. In this study, we first extracted general features from proteins, which were then processed by some advanced computational methods. Three feature types were extracted from protein sequences, indicating the essential properties of proteins at amino acid level. They were described as below.

Protein language model feature

Large language models (LLMs) have achieved remarkable success in processing massive amounts of unlabeled natural language data and learning linguistic embeddings[17]. Utilizing deep learning techniques, these models are able to accurately capture the nuances and complex structures of language, and thus have demonstrated superior performance in several areas of natural language processing (NLP). Inspired by this, the pLMs were designed for protein sequence analysis, which treat protein sequences as a "language" and employs NLP techniques to recognize parse patterns as well as connections in the sequences. Trained on large-scale protein sequences in some databases, such as UniProt[15], pLMs can efficiently capture potential structural and functional features in sequences. The protein embeddings generated by pLMs are valuable in the protein-related researches.

In this study, we employed one newly proposed pLM, named ProtT5[18], to generate protein embeddings. ProtT5 is a 24-layer transformer-based language model that was initially pre-trained on a comprehensive protein dataset from the Big Fantastic Database (BFD)[19,20], and subsequently fine-tuned using the UniRef 50 dataset[21]. In detail, ProtT5 consists of one encoder and one decoder, where the encoder is responsible for converting the primary sequence of a protein into a numeric vector, while the decoder reconstructs the target sequence based on the embeddings yielded by the encoder.

This study directly adopted the pre-trained ProtT5, which was downloaded at https://github.com/agemagician/ProtTrans. The root-associated and non-root-associated proteins were fed into ProtT5. The output of its encoder was picked up as the features of one input protein, which was a $L \times 1024$ embedding matrix, where L represents the length of the protein sequence. It can be seen that each row was the representation of the corresponding amino acid in the sequence. For easy descriptions, this original protein feature was called ProtT5 feature.

BLOSUM62 feature

The BLOSUM62 matrix[22] is a scoring matrix for protein sequence comparison based on the frequency of amino acid substitutions observed in conserved sequence blocks. It is suitable for protein alignment at various evolutionary distances. When two proteins were aligned, the amino acid sequences within each cluster or block were at least 62% identical. It has been widely used to construct various computational models for tackling protein-related problems[23–25]. Compared with other protein scoring matrices, BLOSUM62 matrix has higher sensitivity to the sequences with long evolutionary distances and can detect homologous sequences with weak similarity[26]. Based on this matrix, each protein sequence can be encoded into a $L \times 20$ feature matrix, where L is the length of the protein sequence and each row contains the statistical likelihood between one amino acid and all 20 amino acids. This protein feature type was called BLOSUM62 feature.

PSSM feature

Protein evolutionary information is usually useful in tackling protein-related problems. PSSM[27] is a commonly used type of evolutionary information. In this study, we adopted PSI-BLAST[28] using Swiss-Prot database[29] to generate the PSSM matrix for each root-associated and non-root-associated protein, which was executed with e-value of 0.001, three iterations, and other default parameters. For a protein with sequence length L, its PSSM matrix contains L rows and 20 columns, that is, each amino acid in the sequence is represented by 20 features. This feature type was termed as PSSM feature.

Protein representation

As mentioned above, each protein can be represented by ProtT5, BLOSUM62, and PSSM features. Their detailed information is listed in Table 1. After combining them together, we obtained an $L \times d$ feature matrix for each protein, where $d = 1064$ $(1024 + 20 + 20)$ in this study. For the following formulation, this matrix is denoted by X, which will be refined in subsequent procedures.

Feature type	Dimension[a]
ProtT5 feature	$L \times 1024$
BLOSUM62 feature	$L \times 20$
PSSM feature	$L \times 20$

Table 1. Information of three protein feature types. [a]L in this column stands for the length of protein sequences.

Protein feature improved by HGCN

In recent years, most proposed prediction models contain a feature improving procedure to yield informative features, which are helpful for the following prediction procedure. This study adopted HGCN to improve the original protein features.

Protein contact map prediction

In "Original protein feature extraction" section, each protein is assigned a feature matrix, where each row represents one amino acid in the sequence. To refine this feature matrix, we need to measure the associations between any two amino acids in the sequence so that a hypergraph can be constructed. In view of this, SPOT-Contact-LM[30] was employed, which is a neural network-based contact map prediction method. It processes the one-dimensional sequence features with one-hot encoding using the ESM-1b attention map and generates a contact map via ResNet network. For a protein sequence of length L, a contact probability matrix $C \in R^{L\times L}$ can be generated, where C_{ij} denotes the contact probability between the i-th and j-th amino acids. The contact probability matrix indicates the associations between any two amino acids in the sequence, revealing the structural characteristics of proteins.

Hypergraph construction

Hypergraphs are an extended form of graphs, which allow hyperedges to connect any number of vertices. In this way, hypergraphs can represent higher-order relationships between nodes. A hypergraph is defined as $G = (V, E, W)$, where V is the set of vertices, denoted as $V = \{v_1, v_2, v_3, \ldots, v_n\}$; E is the set of hyperedges, denoted as $E = \{e_1, e_2, e_3, \ldots, e_m\}$; each hyperedge is assigned a weight collected in a diagonal matrix W, denoted as $W = \{w_1, w_2, w_3, \ldots, w_m\}$. Generally, the hypergraph can be represented by a $|V| \times |E|$ correlation matrix H, defined as

$$H(v, \mathrm{e}) = \begin{cases} 1, & v \in \mathrm{e} \\ 0, & v \notin \mathrm{e} \end{cases} . \tag{1}$$

To capture the high-order relationships between amino acids in one protein sequence, a hypergraph was constructed based on the contact probability matrix yielded by SPOT-Contact-LM. In this hypergraph, amino acids in a given protein sequence were defined as vertices. The hyperedges were determined by the K-Nearest Neighbors (KNN) algorithm, which is a popular method to construct hypergraphs[31,32]. In detail, for each amino acid, its K nearest neighbors were determined based on the Euclidean distances between it and other amino acids, where each amino acid was represented by the corresponding row in the contact probability matrix. Then, this amino acid and its K nearest neighbors constituted a hyperedge. Under this operation, the number of hyperedges was equal to the number of vertices (amino acids). Accordingly, the correlation matrix H was a square matrix. As for the weights of hyperedges, they were set to one. The obtained hypergraph was denoted by *HG*.

HGCN

In recent years, GCN has successful applications in several fields. It can capture the pairwise relations in a graph and combine this information with the input features of vertices. For hypergraphs, the newly proposed HGCN[12] can encode high-order relations in them. As mentioned in "Hypergraph construction" section, a hypergraph can be represented by a correlation matrix H and weight W of hyperedges. Based on them, a hyperedge convolution layer of HGCN is defined as

$$X^{(l+1)} = \sigma\left(D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2} X^{(l)} W^{(l)}\right) \tag{2}$$

where $X^{(0)} = X$ (X is the input feature matrix of all vertices, see "Original protein feature extraction" section), $X^{(l)}$ is the output feature matrix at the l-th layer, $W^{(l)}$ is the learnable filter matrix at the l-th layer, σ represents the nonlinear activation function (it was set to LeakyReLU function in this study). D_e denotes the diagonal matrices of hyperedge degrees. The degree of an hyperedge e is defined as $d(e) = \sum_{v\in V} H(v, e)$. D_v stands for the diagonal matrices of vertex degrees. The degree of a vertex v can be computed by $d(v) = \sum_{e\in E} w(e) H(v, e)$.

In this study, we improved the original feature matrix X of a protein by HGCN. In detail, the original feature matrix X and the constructed hypergraph *HG* were fed into HGCN. The output feature matrix was denoted by $F \in R^{L\times f}$, where f denotes the output dimension corresponding to each amino acid.

Multi-head attention

To further highlight important information in $F \in R^{L\times f}$ and tackle the problem of different sizes of F for different proteins, we employed multi-head attention[33] to process F. The attention matrix $M \in R^{r\times L}$ can be calculated by

$$M = SoftMax\left(M_1 \tanh\left(M_2 F^T\right)\right) \tag{3}$$

where $M_1 \in R^{r\times k}$ and $M_2 \in R^{k\times f}$ represent the two attention weight matrices. Subsequently, the learned attention matrix $M \in R^{r\times L}$ is multiplied with F to generate the final feature matrix of one protein. Given that the FCL was selected for prediction, the feature matrix was flattened into a feature vector $Y \in R^{rf}$ with a unified length, that is

$$Y = Flatten(MF). \tag{4}$$

This feature vector contains key information in the protein sequence, which is helpful for the following prediction task.

Prediction and loss function

This study adopted FCL as the prediction function, which contained two layers. The weight matrices of these layers are denoted by $M_3 \in R^{m\times(rf)}$ and $M_4 \in R^m$, respectively. The Sigmoid function is used to calculate the probability P to determine whether the input protein is root-associated or not, that is,

$$P = Sigmoid\left(M_4 M_3 Y^T\right). \tag{5}$$

The probability P is between 0 and 1. If it is higher than the predefined threshold 0.5, the input protein is predicted to be root-associated; otherwise, it is predicted to be non-root-associated.

Based on the predictions, the loss function is used to estimate the quality of prediction. Here, we adopted the widely used loss function of binary cross-entropy, which is defined as

$$L = -\sum \left(ylogp\left(x\right) + \left(1-y\right)\log\left(1-p\left(x\right)\right)\right), \tag{6}$$

where $p\left(x\right)$ is the outcome of the model and y stands for the true label. According to the result of loss function, Adam optimizer[34] was employed to optimize the parameters in this model, including $W^{(l)}$ $(l = 1, 2)$ in HGCN, M_1 and M_2 in multi-head attention, and M_3 and M_4 in FCL.

Model evaluation

In "Dataset" section, 50 training datasets and two test datasets were constructed. On each training dataset, the model was built and evaluated by five-fold cross-validation[35–39]. The average performance was calculated to assess model's performance. Furthermore, the models built on 50 training datasets were applied to the test datasets. Also, the average performance was picked up to estimate the generalization ability of the model.

As a binary classification problem, several metrics have been proposed to assess models' performance. This study selected sensitivity, specificity, accuracy, precision, F-score, Matthews correlation coefficient (MCC), and AUC[40–45]. Before calculating these metrics, it is necessary to determine the four key numbers: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Then, above metrics, except AUC, can be computed by

$$Sensitivity = \frac{TP}{TP+FN} \tag{7}$$

$$Specificity = \frac{TN}{TN+FP} \tag{8}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{9}$$

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

$$F-score = \frac{2\times TP}{2\times TP+FP+FN} \tag{11}$$

$$MCC = \frac{TP\times TN - FP\times FN}{\sqrt{\left(TP+FP\right)\left(TP+FN\right)\left(TN+FP\right)\left(TN+FN\right)}}. \tag{12}$$

Among these metrics, sensitivity, specificity, accuracy, precision, F-score are all between 0 and 1, whereas MCC is between – 1 and 1. The high values suggest the high performance. AUC is quite different from above metrics, which can evaluate model performance under a set of thresholds for the probability of predicting positive samples. A group of sensitivity and 1-specificity were obtained by setting various thresholds. Then, a curve with sensitivity as Y-axis and 1-specificity as X-axis is plotted in a coordinate system, which is generally called the receiver operating characteristic curve (ROC). AUC is defined as the area under this curve. Generally, the higher the AUC, the higher the performance of the model.

Among above metrics, sensitivity, specificity, and precision only evaluate models' performance from a special aspect, whereas accuracy, F-score, MCC, and AUC can give an overall evaluation. Thus, we mainly used overall metrics when comparing the performance of different models.

Outline of the Hypergraph-Root

In this study, a computational model was designed for the prediction of root-associated proteins. The entire construction procedures are illustrated in Fig. 1. Three feature types were extracted from each protein sequence, including PSSM, ProtT5, and BLOSUM62 features. At the same time, a contact probability matrix was built from each protein sequence through SPOT-Contact-LM, which was further used to construct a hypergraph graph. Tree feature types and the hypergraph graph were fed into HGCN to yield high-order features. After the high-order features processed by multi-head attention and flattening, they were subject to the FCL to make predictions. For easy descriptions, the constructed model was called Hypergraph-Root.

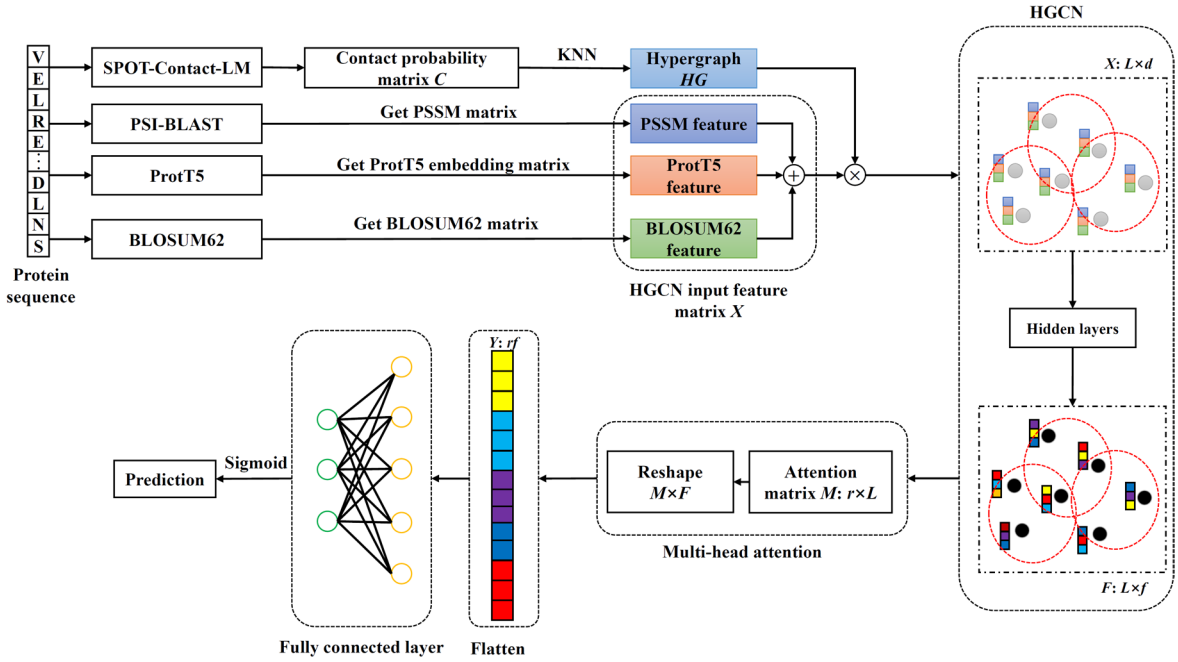


Fig. 1. Construction procedures of Hypergraph-Root. Three protein feature types are derived from sequences. These features are improved by a hypergraph convolutional network and then deeply optimized by a multi-head attention. The refined features are fed into a fully connected layer to make predictions.

Module	Hyperparameter	Value
Hypergraph construction	K	10
HGCN	Number of layers	2
	Size of layers	256 (first layer) 64 (second layer)
Multi-head attention	Number of attention heads (k, r)	64
Fully connected layer	Number of neurons (m)	1024

Table 2. The settings of hyperparameters in Hypergraph-Root.

Results and Discussion

Hyperparameter adjustment

The proposed model Hypergraph-Root contained several modules, including original feature extraction, hypergraph construction, HGCN, multi-head attention, and FCL. The hyperparameters in some modules should be tuned for improving the performance of Hypergraph-Root. We used a two-step strategy to tune main hyperparameters.

In the first step, we mainly focused on the parameter K when constructing the hypergraph HG, which determined the number of vertices in each hyperedge, and the number of attention heads (k and r) in multi-head attention. For K, we attempted several values, including 5, 10, 15, …, 30, 35. For k and r, they were set to the same value in {32, 64, 128}. We used grid research to build models and evaluated them using five-fold cross-validation on 50 training datasets. The average values on seven metrics are listed in Supplementary Table S1. It can be found that when $K=10$ and $k=r=64$, the model yielded the best performance. Although its sensitivity and specificity were not highest, they only assessed model's performance on one aspect. The overall metrics (e.g. accuracy, F-score, MCC, and AUC) of this model were consistently the highest. Thus, we determined these hyperparameters as above values.

After determining above hyperparameters, we tuned the sizes of layers in HGCN and the number of neurons (m) in the first layer of FLC. First, the number of layers in HGCN was set two, similar to the general setting of GCN. The sizes of two layers were set to various values in {64, 128, 256, 512}. The number of neurons in the first layer of FLC was set to 256, 512 and 1024. The models under different settings of above hyperparameters were also evaluated by five-fold cross-validation on 50 training datasets. The average performance is provided in Supplementary Table S2. It can be observed that when the sizes of two layers in HGCN were set to 256 (first layer) and 64 (second layer), and the number of neurons in the first layer of FLC was set to 1024, the model consistently yielded the highest values on all seven metrics. Thus, above values were set to these hyperparameters.

With above argument, we determined the settings of main hyperparameters, which are listed in Table 2.

Measurement	Training dataset	Imbalanced test dataset	Balanced test dataset
Accuracy	0.8372 ± 0.0086	0.8245 ± 0.0104	0.8482 ± 0.0052
Precision	0.8316 ± 0.0099	0.2193 ± 0.0109	0.8035 ± 0.0129
Sensitivity	0.8475 ± 0.0117	0.8947 ± 0.0316	0.9227 ± 0.0195
Specificity	0.8270 ± 0.0116	0.8207 ± 0.0115	0.7737 ± 0.0225
F-score	0.8389 ± 0.0087	0.3521 ± 0.0149	0.8588 ± 0.0048
MCC	0.6755 ± 0.0173	0.3894 ± 0.0176	0.7050 ± 0.0103
AUC	0.8988 ± 0.0088	0.9254 ± 0.0096	0.9232 ± 0.0091

Table 3. Performance of Hypergraph-Root on 50 training datasets under five-fold cross-validation and two test datasets.

Feature type	Accuracy	Precision	Sensitivity	Specificity	F-score	MCC	AUC
PSSM feature	0.6626 ± 0.0179	0.6519 ± 0.0174	0.7027 ± 0.0211	0.6224 ± 0.0237	0.6755 ± 0.0171	0.3271 ± 0.0358	0.7066 ± 0.0182**
BLOSUM62 feature	0.7232 ± 0.0100	0.7243 ± 0.0118	0.7233 ± 0.0111	0.7229 ± 0.0152	0.7231 ± 0.0095	0.4470 ± 0.0198	0.7611 ± 0.0098**
ProtT5 feature	0.8284 ± 0.0100	0.8271 ± 0.0107	0.8321 ± 0.0137	0.8248 ± 0.0117	0.8291 ± 0.0103	0.6576 ± 0.0201	0.8941 ± 0.0094**
PSSM and BLOSUM62 features	0.7479 ± 0.0132	0.7372 ± 0.0147	0.7725 ± 0.0134	0.7230 ± 0.0185	0.7540 ± 0.0123	0.4969 ± 0.0264	0.7927 ± 0.0144**
PSSM and ProtT5 features	0.8343 ± 0.0094	0.8298 ± 0.0111	0.8429 ± 0.0113	0.8256 ± 0.0131	0.8357 ± 0.0092	0.6696 ± 0,0187	0.8973 ± 0.0084
ProtT5 and BLOSUM62 features	0.8307 ± 0.0115	0.8275 ± 0.0147	0.8372 ± 0.0108	0.8241 ± 0.0171	0.8319 ± 0.0109	0.6621 ± 0.0231	0.8945 ± 0.0102**

Table 4. Results of ablation tests on features. "**" in the last column indicates that the *p*-value between the AUC of Hypergraph-Root and AUC in this column is less than 0.01.

Performance of Hypergraph-Root on the training and test datasets

The Hypergraph-Root was constructed using the hyperparameter settings listed in Table 2. Its performance was evaluated by five-fold cross-validation on 50 training datasets. Each training dataset contained same positive samples and randomly selected negative samples. The predicted results were counted as metrics mentioned in "Model evaluation" section. The average and standard deviation values were calculated for each metric, which is listed in Table 3. The accuracy, precision, sensitivity, specificity, F-score, MCC, and AUC are 0.8372, 0.8316, 0.8475, 0.8270, 0.8389, 0.6755, and 0.8988, respectively. Evidently, all metrics except MCC exceeded 0.8, whereas MCC was higher than 0.65. All these results suggested the high performance of Hypergraph-Root. Furthermore, the standard deviation values were low, suggesting the stability of Hypergraph-Root.

Two test datasets (imbalanced and balanced test datasets S_{te}^1 and S_{te}^2) were fed into Hypergraph-Root built on 50 training datasets. The average performance was calculated, which is listed in Table 3. On the imbalanced test dataset S_{te}^1, the average accuracy, sensitivity, specificity, AUC were quite high (> 0.82) and they were similar to or even higher than those on training datasets. These results implied that Hypergraph-Root had a strong generalization ability. The average precision, F-score, and MCC were low (< 0.4) and they were evidently lower than those on training datasets. However, this comparison was not fair. In S_{te}^1, the negative samples were 17.8 times as many as positive samples, whereas negative samples were as many as positive samples in training datasets. Thus, the metrics were obtained under quite different sample distributions. Simple comparisons cannot yield reliable results, especial for precision, F-score, and MCC, which are quite sensitive to the data imbalanced problem. According to sensitivity (0.8947) and specificity (0.8207), meaning the prediction accuracy on positive and negative samples, respectively, Hypergraph-Root can correctly predict most positive and negative samples, confirming its strong generalization ability.

On the balanced test dataset S_{te}^2, the average accuracy, F-score, MCC, AUC were slightly higher than those on the training datasets. The average sensitivity was evidently higher than that on the training datasets and the average precision and specificity were slightly lower than those on the training datasets. Accordingly, the overall performance on the balanced test dataset and training datasets was quite similar, further proving the strong generalization ability of Hypergraph-Root.

Ablation tests

The Hypergraph-Root was constructed by employing three feature types, which were processed by several modules. Here, we proved that the employment of these feature types and module design were reasonable.

Three feature types were extracted to represent proteins, including BLOSUM62, PSSM, and ProtT5 features. There were six different combinations of feature types except the combination of all three feature types. The models using above six feature combinations were built on 50 training datasets and evaluated by five-fold cross-validation. The results are listed in Table 4. By comparing the metrices in Table 3, Hypergraph-Root provided the highest performance on all metrics. We further performed the paired student's t-test on AUC values yielded by Hypergraph-Root and above models, obtaining the *p*-values. The significance level is marked in Table 4, where "**" and "*" indicate the *p*-values less than 0.01 and between 0.01 and 0.05, respectively. It can be found that five models yielded significant lower AUC values than Hypergraph-Root, suggesting the superiority of Hypergraph-Root. As the six models lacked at least one feature type, it was proved that all feature types can bring positive

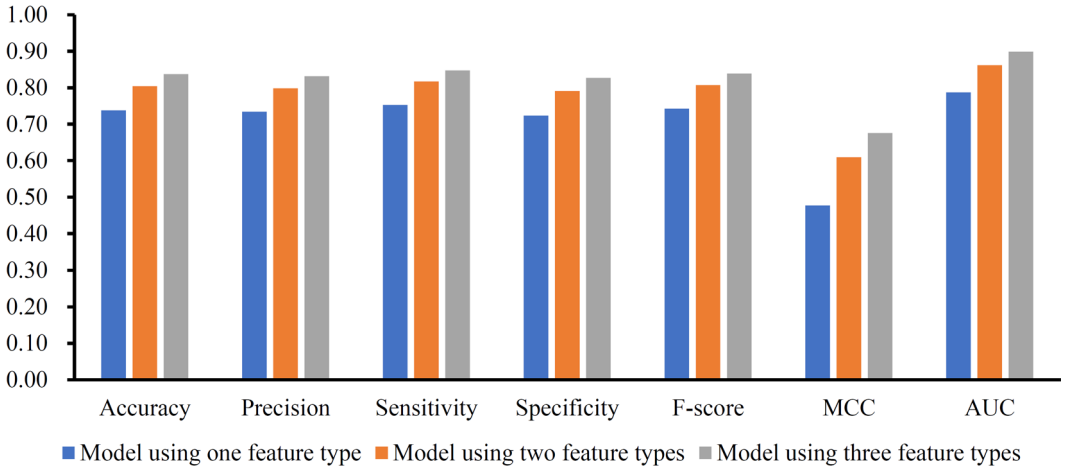


Fig. 2. Bar chart to show the performance of models using one, two, or three feature types. Models using more feature types yield higher performance.

Module	Accuracy	Precision	Sensitivity	Specificity	F-score	MCC	AUC
Hypergraph-Root (no HGCN)	0.8355 ± 0.0092	0.8283 ± 0.0117	0.8483 ± 0.0094	0.8228 ± 0.0137	0.8377 ± 0.0088	0.6719 ± 0.0184	0.9017 ± 0.0066*
Hypergraph-Root (GCN)	0.8343 ± 0.0074	0.8297 ± 0.0084	0.8431 ± 0.0116	0.8255 ± 0.0102	0.8358 ± 0.0078	0.6695 ± 0.0149	0.8971 ± 0.0086

Table 5. Results of ablation tests on model architectures. "*" in the last column indicates that the *p*-value between the AUC of Hypergraph-Root and AUC in this column is between 0.01 and 0.05.

contributions to Hypergraph-Root. To further confirm this conclusion, we picked up the metrics yielded by models using one or two feature types and calculate the average value for each metric, which is illustrated in Fig. 2. It can be observed that on each metric, its value followed an increasing trend when more feature types were added, indicating more features can bring higher performance. This result was reasonable because more features can give more complete representations on proteins, thereby improving model's performance.

With above arguments, all three feature types provided positive contributions to Hypergraph-Root. However, their contributions were not same. According to the performance of models using one feature type (first three rows in Table 4), the model using ProtT5 feature yielded the highest performance, followed by the models using BLOSUM62 and PSSM features. Thus, ProtT5 feature gave the highest contributions to Hypergraph-Root, followed by BLOSUM62 and PSSM features. The result that BLOSUM62 feature was more important than PSSM feature in predicting root-associated proteins was same as that in the previous study[11]. As for ProtT5 feature, it was yielded by a pLM, which deeply integrates lots of information of protein sequences and their associations. Its information is more abundant than BLOSUM62 and PSSM features, inducing higher performance of the model using this feature type.

Among several modules of Hypergraph-Root, HGCN may play an essential role. To verify this, we constructed two models. The first model directly removed HGCN. In this case, the BLOSUM62, PSSM, and ProtT5 features were directly fed into the multi-head attention. This model is called Hypergraph-Root (no HGCN). The second model was obtained by replacing HGCN with GCN, which was called Hypergraph-Root (GCN). Both models were built on 50 training datasets and evaluated by five-fold cross validation. The evaluation results are presented in the Table 5. The significance level on AUC of Hypergraph-Root is also marked in this table, which was obtained by the paired student's t-test. It was clear that Hypergraph-Root provided the best performance on five metrics and ranks second on two metrics (sensitivity and AUC) by comparing the performance of Hypergraph-Root (Table 3). This suggests that the use of the HGCN can improve model performance, suggesting its positive contribution to Hypergraph-Root.

Comparison with models using traditional machine learning algorithms

In this study, some deep learning algorithms, such as HGCN and multi-head attention, were employed to construct Hypergraph-Root. To validate that they were helpful to accurately predict root-associated proteins, some traditional machine learning algorithms were adopted to construct models, which were further compared with Hypergraph-Root.

Three feature types: ProtT5, BLOSUM62, and PSSM features, were used in Hypergraph-Root. They were also used to construct traditional machine learning based models. Due to the different sizes of feature matrices for proteins of different lengths, they were processed as follows. For BLOSUM62 and PSSM features, Bigram method[46] was adopted to convert each feature type into a 20×20 feature matrix, which was further flattened into a 400-dimensional feature vector. As for ProtT5 features, the average operation was adopted to yield a 1024-dimensional feature vector. Finally, each protein was represented by an 1824-dimensional feature vector.

Then, four traditional machine learning algorithms: multilayer perceptron (MLP), decision tree (DT)[47], SVM[48] and random forest (RF)[49] were used to construct prediction models based on above feature representation. These algorithms have wide applications in tackling various problems in bioinformatics[36,37,50–53]. For convenience, the corresponding packages in scikit-learn[54] were directly employed to implement above four algorithms. They were executed with their default parameters. For each feature type combination, four models were built based on above four algorithms. All models were trained on 50 training datasets and evaluated by five-fold cross-validation. Their average performance is listed in Table 6. It can be found that Hypergraph-Root yielded the highest performance on all metrics except AUC by comparing the metrics of Hypergraph-Root (Table 3). Its AUC (0.8988) was slightly lower than the highest AUC, which was 0.9032. The significance level on AUC of Hypergraph-Root comparing with AUC values in Table 6 is also marked in this table. Evidently, Hypergraph-Root generally outperformed traditional machine learning based models, implying using deep learning techniques can indeed improve the performance of the model. Furthermore, by observing the models using PSSM, BLOSUM62, and ProtT5 features, we can find that models using ProtT5 features generally generated the best performance, whereas models using BLOSUM62 features were better than those using PSSM features. These results further confirmed the different importance of three feature types in predicting root-associated proteins, that is, ProtT5 feature was the most important, followed by BLOSUM62 and PSSM features.

Comparison with previous models

To date, two models (SVM-Root[10] and Graph-Root[11]) have been proposed to predict root-associated proteins. Here, they were compared with Hypergraph-Root to show its superiority. The five-fold cross-validation results of three models on training datasets are shown in Fig. 3. The MCC of SVM-Root was not reported in Kumar Meher et al.'s study. It was inferred by reconstructing confusion matrix based on sensitivity and specificity. It can be observed that Hypergraph-Root generated much better performance than SVM-Root and Graph-Root. Furthermore, the paired student t-test was performed on AUC values yielded by Hypergraph-Root and above two models, resulting in the p-values of 3.699×10^{-46} and 3.928×10^{-49}. It was suggested the significant superiority of Hypergraph-Root on the training datasets. Furthermore, the independent test results are shown in Fig. 4. As the SVM-Root and Graph-Root were both tested on an imbalanced test dataset, we also listed the metrics of

Feature	Classification algorithm	Accuracy	Precision	Sensitivity	Specificity	F-score	MCC	AUC
PSSM	MLP	0.6725 ± 0.0170	0.6757 ± 0.0179	0.6664 ± 0.0235	0.6794 ± 0.0209	0.6694 ± 0.0184	0.3464 ± 0.0344	0.7382 ± 0.0178**
	DT	0.5835 ± 0.0149	0.5847 ± 0.0152	0.5811 ± 0.0198	0.5867 ± 0.0208	0.5812 ± 0.0159	0.1680 ± 0.0291	0.5839 ± 0.0145**
	SVM	0.6591 ± 0.0139	0.6716 ± 0.0159	0.6318 ± 0.0306	0.6895 ± 0.0262	0.6479 ± 0.0185	0.3230 ± 0.0270	0.7218 ± 0.0141**
	RF	0.6578 ± 0.0161	0.6640 ± 0.0163	0.6434 ± 0.0204	0.6744 ± 0.0177	0.6516 ± 0.0175	0.3180 ± 0.0316	0.7200 ± 0.0154**
BLOSUM62	MLP	0.7032 ± 0.0156	0.7105 ± 0.0173	0.6877 ± 0.0190	0.7194 ± 0.0200	0.6976 ± 0.0162	0.4076 ± 0.0316	0.7627 ± 0.0149**
	DT	0.6045 ± 0.0189	0.6056 ± 0.0201	0.6014 ± 0.0237	0.6083 ± 0.0284	0.6020 ± 0.0192	0.2097 ± 0.0382	0.6048 ± 0.0191**
	SVM	0.7186 ± 0.0105	0.7353 ± 0.0140	0.6857 ± 0.0146	0.7526 ± 0.0174	0.7082 ± 0.0110	0.4396 ± 0.0216	0.7886 ± 0.0101**
	RF	0.6960 ± 0.0109	0.7101 ± 0.0137	0.6654 ± 0.0146	0.7281 ± 0.0178	0.6853 ± 0.0111	0.3945 ± 0.0220	0.7644 ± 0.0095**
ProtT5	MLP	0.8206 ± 0.0113	0.8219 ± 0.0138	0.8196 ± 0.0126	0.8217 ± 0.0160	0.8199 ± 0.0110	0.6418 ± 0.0226	0.8993 ± 0.0089
	DT	0.6909 ± 0.0153	0.6932 ± 0.0161	0.6872 ± 0.0267	0.6956 ± 0.0219	0.6886 ± 0.0178	0.3832 ± 0.0313	0.6914 ± 0.0157**
	SVM	0.8227 ± 0.0111	0.8182 ± 0.0132	0.8308 ± 0.0143	0.8150 ± 0.0161	0.8236 ± 0.0110	0.6460 ± 0.0221	0.9003 ± 0.0085
	RF	0.8044 ± 0.0123	0.8085 ± 0.0150	0.7992 ± 0.0128	0.8108 ± 0.0177	0.8026 ± 0.0117	0.6102 ± 0.0245	0.8850 ± 0.0102**
PSSM + BLOSUM62	MLP	0.7405 ± 0.0153	0.7484 ± 0.0177	0.7272 ± 0.0160	0.7544 ± 0.0204	0.7363 ± 0.0152	0.4821 ± 0.0309	0.8107 ± 0.0151**
	DT	0.6093 ± 0.0161	0.6104 ± 0.0155	0.6076 ± 0.0240	0.6117 ± 0.0205	0.6075 ± 0.0180	0.2196 ± 0.0321	0.6097 ± 0.0161**
	SVM	0.7345 ± 0.0102	0.7472 ± 0.0136	0.7116 ± 0.0166	0.7590 ± 0.0186	0.7274 ± 0.0108	0.4709 ± 0.0206	0.8082 ± 0.0096**
	RF	0.7015 ± 0.0137	0.7084 ± 0.0148	0.6877 ± 0.0189	0.7170 ± 0.0188	0.6962 ± 0.0146	0.4050 ± 0.0268	0.7751 ± 0.0139**
PSSM + ProtT5	MLP	0.8221 ± 0.0120	0.8250 ± 0.0144	0.8186 ± 0.0127	0.8255 ± 0.0169	0.8210 ± 0.0116	0.6444 ± 0.0240	0.9032 ± 0.0110*
	DT	0.6879 ± 0.0149	0.6894 ± 0.0170	0.6867 ± 0.0192	0.6899 ± 0.0229	0.6868 ± 0.0149	0.3769 ± 0.0306	0.6883 ± 0.0153**
	SVM	0.8194 ± 0.0110	0.8166 ± 0.0130	0.8257 ± 0.0152	0.8149 ± 0.0153	0.8199 ± 0.0112	0.6402 ± 0.0220	0.8991 ± 0.0085
	RF	0.8028 ± 0.0107	0.8076 ± 0.0126	0.7965 ± 0.0131	0.8102 ± 0.0143	0.8009 ± 0.0109	0.6067 ± 0.0213	0.8834 ± 0.0079**
ProtT5 + BLOSUM62	MLP	0.8186 ± 0.0135	0.8187 ± 0.0143	0.8197 ± 0.0159	0.8182 ± 0.0154	0.8181 ± 0.0137	0.6383 ± 0.0269	0.8991 ± 0.0101
	DT	0.6889 ± 0.0168	0.6905 ± 0.0161	0.6859 ± 0.0247	0.6923 ± 0.0188	0.6870 ± 0.0184	0.3786 ± 0.0334	0.6891 ± 0.0167**
	SVM	0.8198 ± 0.0114	0.8172 ± 0.0141	0.8249 ± 0.0115	0.8156 ± 0.0169	0.8200 ± 0.0109	0.6404 ± 0.0231	0.8969 ± 0.0086
	RF	0.8011 ± 0.0109	0.8105 ± 0.0130	0.7878 ± 0.0143	0.8154 ± 0.0141	0.7977 ± 0.0114	0.6037 ± 0.0224	0.8827 ± 0.0096**
ProtT5 + BLOSUM62 + PSSM	MLP	0.8192 ± 0.0105	0.8201 ± 0.0123	0.8187 ± 0.0132	0.8202 ± 0.0146	0.8184 ± 0.0106	0.6391 ± 0.0211	0.8989 ± 0.0088
	DT	0.6849 ± 0.0168	0.6857 ± 0.0175	0.6836 ± 0.0222	0.6864 ± 0.0211	0.6834 ± 0.0177	0.3703 ± 0.0337	0.6850 ± 0.0170**
	SVM	0.8193 ± 0.0114	0.8189 ± 0.0145	0.8215 ± 0.0168	0.8180 ± 0.0173	0.8191 ± 0.0115	0.6396 ± 0.0230	0.8986 ± 0.0074
	RF	0.7989 ± 0.0095	0.8093 ± 0.0110	0.7834 ± 0.0155	0.8156 ± 0.0125	0.7948 ± 0.0106	0.5994 ± 0.0187	0.8810 ± 0.0073**

Table 6. Comparison with different traditional machine learning based models. "**" and "*" in the last column indicate that the p-value between the AUC of Hypergraph-Root and AUC in this column is less than 0.01 and between 0.01 and 0.05 , respectively.

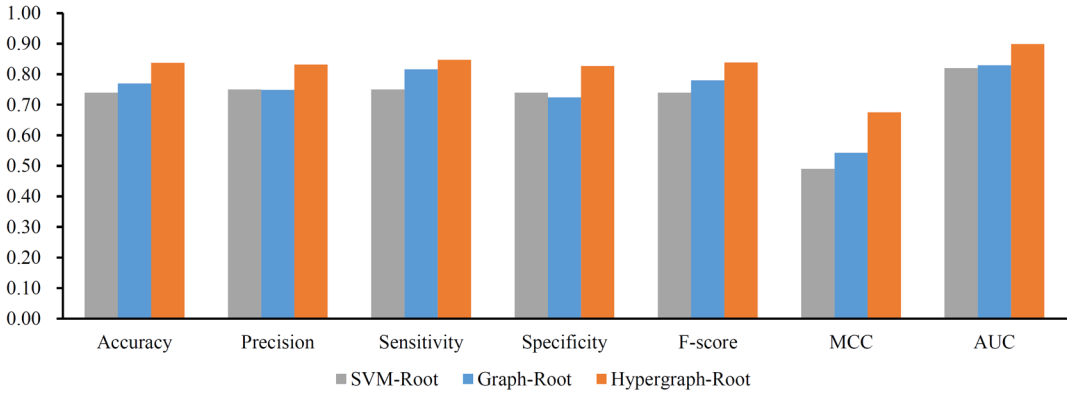


Fig. 3. Bar chart to compare Hypergraph-Root and two previous models on training datasets. Hypergraph-Root outperforms SVM-Root and Graph-Root.

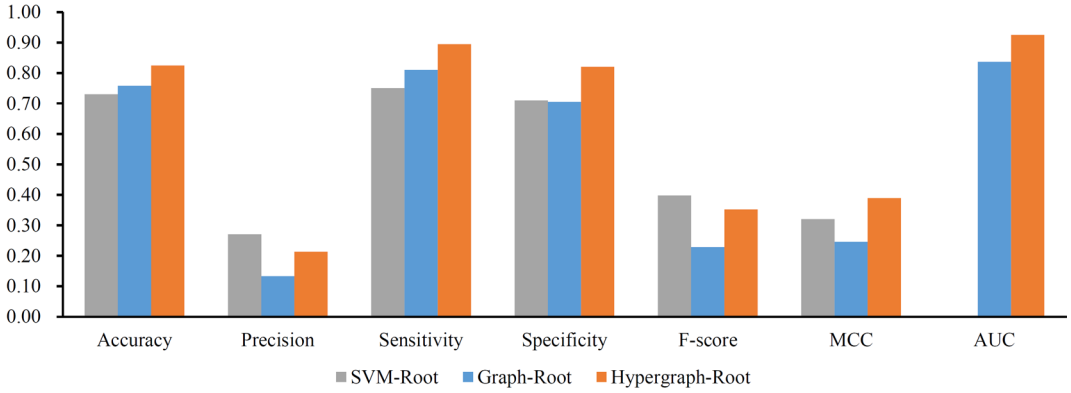


Fig. 4. Bar chart to compare Hypergraph-Root and two previous models on imbalanced test dataset. Hypergraph-Root has stronger generalization ability than SVM-Root and Graph-Root.

Hypergraph-Root on the imbalanced test dataset. The metrics of SVM-Root and Graph-Root not mentioned in their original studies were also inferred by reconstructing confusion matrices. However, this method cannot infer the AUC of SVM-Root, which is not listed in Fig. 4. Hypergraph-Root also yielded the highest performance on most metrics, proving it had stronger generalization ability than SVM-Root and Graph-Root.

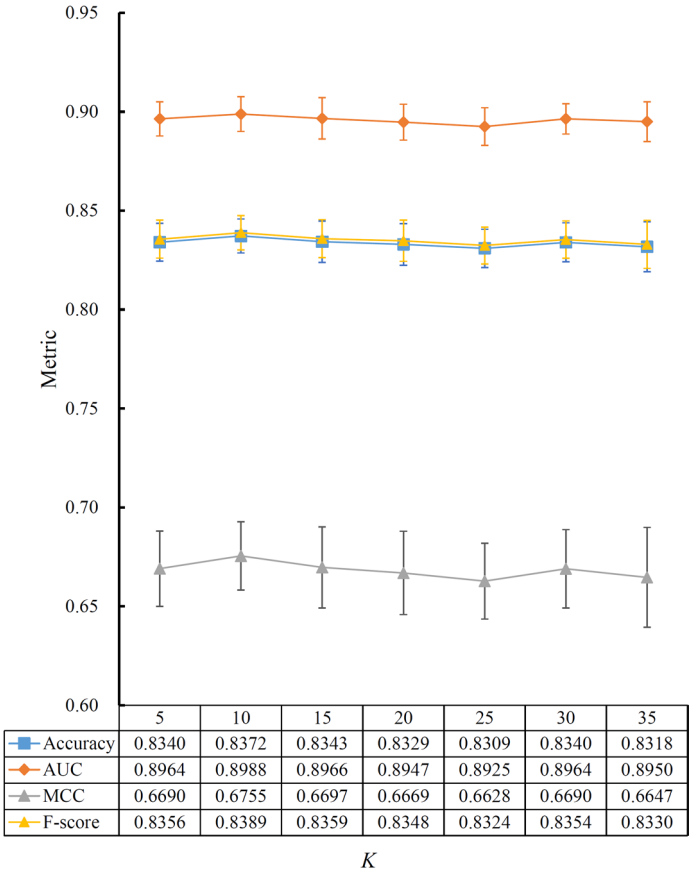
SVM-Root extracted protein features from sequences and used the classical classification algorithm, SVM, as the prediction engine. It cannot yield high-order features and the prediction ability of SVM was not very high, which was the main reason why its performance was low. As for, Graph-Root, although it utilized some deep learning algorithms, the original features cannot contain enough essential information of proteins. The Hypergraph-Root proposed in this study employed the features generated by a pLM, which included very abundant information of proteins. Furthermore, the HGCN in Hypergraph-Root can capture complicated relationships among amino acids in one protein sequence, which was helpful to refine protein features. Above two aspects induced the higher performance of Hypergraph-Root.

Influence of hypergraph on Hypergraph-Root

In this study, we employed HGCN to generate high-order features of proteins. The hypergraph clearly plays a key role in HGCN. The KNN was adopted to construct the hypergraph, where the hyperparameter K was essential. Here, we investigated its influence on the performance of Hypergraph-Root. It was set to seven values between 5 and 35 for constructing different hypergraphs and thus seven different models were built. These models were evaluated by five-fold cross-validation on training datasets. Four overall metrics (accuracy, F-score, MCC, and AUC) yielded by Hypergraph-Root with different values of K are illustrated in Fig. 5. It can be observed that when $K = 10$, the Hypergraph-Root yielded the highest overall performance. This result was reasonable because the small K cannot reflect the high-order relations between amino acids in sequences, whereas the large K may bring useless noises.

Case studies

In this study, a root-associated prediction model, Hypergraph-Root, was proposed. To prove its practicality, a case study was conducted. According to "Performance of Hypergraph-Root on the training and test datasets" section, each protein in the imbalanced test dataset was predicted 50 times by Hypergraph-Root with different



	5	10	15	20	25	30	35
Accuracy	0.8340	0.8372	0.8343	0.8329	0.8309	0.8340	0.8318
AUC	0.8964	0.8988	0.8966	0.8947	0.8925	0.8964	0.8950
MCC	0.6690	0.6755	0.6697	0.6669	0.6628	0.6690	0.6647
F-score	0.8356	0.8389	0.8359	0.8348	0.8324	0.8354	0.8330

Fig. 5. Effect of the hyperparameter K for constructing the hypergraph on Hypergraph-Root. The X-axis represents the parameter K when constructing hypergraph, which determines the number of nodes in hyperedges. The Y-axis denotes the metrics, including accuracy, AUC, MCC, and F-score. When $K = 10$, the model yields the highest performance.

GO ID	Description	Proteins	References
GO:0016020	Membrane	Q10LN5, Q6ZJ91, B0YPQ4	[56,57]
GO:0016567	Protein ubiquitination	O82353, Q10PI9	[58,59]

Table 7. Latent root-associated proteins and their gene ontology terms.

training datasets. Thus, each negative sample in this test dataset was assigned 50 labels (positive or negative). We picked up negative samples in this test dataset, which were all predicted to be positive by Hypergraph-Root, obtaining 56 proteins. These proteins may be latent root-associated proteins with high likelihoods. To show they were related to root, they were fed into InterProScan (Release 105.0)[55] to extract their gene ontology (GO) terms. Among the GO terms annotated to above 56 proteins, membrane (GO:0016020) was annotated to three proteins (Q10LN5, Q6ZJ91, B0YPQ4) and protein ubiquitination (GO:0016567) was annotated to two proteins (O82353, Q10PI9). This information is listed in Table 7.

Tsay et al. reveal the functions of nitrate transporters in the root, whereas most nitrate transporters are membrane proteins[56]. In addition, aquaporin PIP2;1 has been confirmed to affect water transport and root growth in rice[57]. The aquaporin is also a type of membrane protein. Above references proved the strong associations between membrane (GO:0016020) and root. Thus, the three proteins (Q10LN5, Q6ZJ91, B0YPQ4) annotated by this GO term may also have special associations with root, i.e., they may be latent root-associated proteins.

As for another GO term, protein ubiquitination (GO:0016567), Marrocco et al. reported that APC/C (anaphase promoting complex or cyclosome), a master ubiquitin protein ligase (E3), plays a role in plant vasculature development and organization[58]. OsHRZ1 and OsHRZ2 possess ubiquitination activity, which are susceptible to degradation in roots irrespective of iron conditions[59]. Accordingly, this GO term is also related to root in plant, inducing the special relationships between the proteins (O82353, Q10PI9) annotated by it and root.

With above argument, five proteins (Q10LN5, Q6ZJ91, B0YPQ4, O82353, Q10PI9) identified by Hypergraph-Root can be confirmed to be related to root. It implied that Hypergraph-Root had an ability for discovering novel root-associated proteins.

Conclusion

This study proposed a computational model for predicting root-associated proteins. The model employed some informative protein features and adopted several advanced computational methods, yielding a strong ability to identify root-associated proteins. The protein features yielded by ProtT5 were deemed to give high contributions to determine root-associated proteins. At present, our model provided the higher performance than all existing models. With the help of our model, the latent root-associated proteins can be identified. Then, the biochemistry experiments can be designed to validate the identified proteins, thereby reducing costs and time. It is hopeful that the proposed model can be a useful tool for identifying plant root-associated proteins. The data and codes in this study are available at https://github.com/Xxy0413-1119/Hypergraph-Root.

Data availability

The data underlying this study are openly available in RGPDB database at http://sysbio.unl.edu/RGPDB/. The codes and refined data are available at https://github.com/Xxy0413-1119/Hypergraph-Root.

Received: 17 September 2025; Accepted: 2 January 2026
Published online: 08 January 2026

References

1. Hodge, A., Berta, G., Doussan, C., Merchan, F. & Crespi, M. Plant root growth, architecture and function. *Plant Soil* **321**, 153–187 (2009).
2. Huang, B., Rachmilevitch, S. & Xu, J. Root carbon and protein metabolism associated with heat tolerance. *J. Exp. Bot.* **63**, 3455–3465 (2012).
3. Fageria, N. K. *The Role of Plant Roots in Crop Production* (CRC Press, 2012).
4. Dawson, N., Sillitoe, I., Marsden, R. L. & Orengo, C. A. The classification of protein domains. In *Bioinformatics: Volume I: Data, Sequence Analysis, and Evolution*, 137–164 (2017).
5. Moisseyev, G. et al. RGPDB: database of root-associated genes and promoters in maize, soybean, and sorghum. *Database J. Biol. Databases Curation* **2020**, baaa038. https://doi.org/10.1093/database/baaa038 (2020).
6. Almagro Armenteros, J. J. et al. SignalP 5.0 improves signal peptide predictions using deep neural networks. *Nat. Biotechnol.* **37**, 420–423. https://doi.org/10.1038/s41587-019-0036-z (2019).
7. Yang, L., Gao, J., Gao, M., Jiang, L. & Luo, L. Characterization of plasma membrane proteins in stylosanthes leaves and roots using simplified enrichment method with a nonionic detergent. *Front. Plant Sci.* **13**, 1071225. https://doi.org/10.3389/fpls.2022.1071225 (2022).
8. Iwasaki, Y. et al. Proteomics analysis of plasma membrane fractions of the root, leaf, and flower of rice. *Int. J. Mol. Sci.* **21**, 6988. https://doi.org/10.3390/ijms21196988 (2020).
9. Voothuluru, P., Anderson, J. C., Sharp, R. E. & Peck, S. C. Plasma membrane proteomics in the maize primary root growth zone: Novel insights into root growth adaptation to water stress. *Plant Cell Environ.* **39**, 2043–2054 (2016).
10. Kumar Meher, P. et al. SVM-root: Identification of root-associated proteins in plants by employing the support vector machine with sequence-derived features. *Curr. Bioinform.* **19**, 69–80. https://doi.org/10.2174/1574893618666230417104543 (2024).
11. Zhou, B., Liu, S. Y., Chen, L. & Dai, Q. Graph-root: Prediction of root-associated proteins in maize, sorghum, and soybean based on graph convolutional network and network embedding method. *Curr. Bioinform.* https://doi.org/10.2174/0115748936343410241008103219 (2024).
12. Feng, Y., You, H., Zhang, Z., Ji, R. & Gao, Y. in *Proceedings of the AAAI Conference on Artificial Intelligence* 3558–3565.
13. Szklarczyk, D. et al. The STRING database in 2023: Protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Res.* **51**, D638–D646. https://doi.org/10.1093/nar/gkac1000 (2022).
14. Yates, A. D. et al. Ensembl Genomes 2022: An expanding genome resource for non-vertebrates. *Nucleic Acids Res.* **50**, D996-d1003. https://doi.org/10.1093/nar/gkab1007 (2022).
15. UniProt Consortium. UniProt: The universal protein knowledgebase in 2023. *Nucleic Acids Res.* **51**, D523–D531. https://doi.org/10.1093/nar/gkac1052 (2023).
16. Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: Accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–3152. https://doi.org/10.1093/bioinformatics/bts565 (2012).
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. in *26th International Conference on Neural Information Processing Systems* 3111–3119 (2013).
18. Elnaggar, A. et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7112–7127 (2022).
19. Steinegger, M., Mirdita, M. & Söding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods* **16**, 603–606 (2019).
20. Steinegger, M. & Söding, J. Clustering huge protein sequence sets in linear time. *Nat. Commun.* **9**, 2542 (2018).
21. Suzek, B. E. et al. UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932 (2015).
22. Henikoff, S. & Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U. S. A.* **89**, 10915–10919. https://doi.org/10.1073/pnas.89.22.10915 (1992).
23. Yao, L. et al. DeepAFP: An effective computational framework for identifying antifungal peptides based on deep learning. *Protein Sci.* **32**, e4758. https://doi.org/10.1002/pro.4758 (2023).
24. Fang, Y. et al. AFP-MFL: Accurate identification of antifungal peptides using multi-view feature learning. *Brief. Bioinform.* **24**, bbac606. https://doi.org/10.1093/bib/bbac606 (2023).
25. Ning, Q. & Li, J. DLF-Sul: A multi-module deep learning framework for prediction of S-sulfinylation sites in proteins. *Brief. Bioinform.* **23**, bbac323. https://doi.org/10.1093/bib/bbac323 (2022).
26. Pearson, W. R. An introduction to sequence similarity ("homology") searching. *Curr. Protoc. Bioinform.* **42**, 3–1 (2013).
27. Cheol Jeong, J., Lin, X. & Chen, X.-W. On position-specific scoring matrix for protein function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **8**, 308–315 (2010).
28. Altschul, S. F. et al. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402. https://doi.org/10.1093/nar/25.17.3389 (1997).

29. Boeckmann, B. et al. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.* **31**, 365–370. https://doi.org/10.1093/nar/gkg095 (2003).
30. Singh, J., Litfin, T., Singh, J., Paliwal, K. & Zhou, Y. SPOT-Contact-LM: Improving single-sequence-based prediction of protein contact map using a transformer language model. *Bioinformatics* **38**, 1888–1894. https://doi.org/10.1093/bioinformatics/btac053 (2022).
31. Ouyang, D. et al. HGCLAMIR: Hypergraph contrastive learning with attention mechanism and integrated multi-view representation for predicting miRNA-disease associations. *PLoS Comput. Biol.* **20**, e1011927. https://doi.org/10.1371/journal.pcbi.1011927 (2024).
32. Peng, W., He, Z., Dai, W. & Lan, W. MHCLMDA: Multihypergraph contrastive learning for miRNA–disease association prediction. *Brief. Bioinform.* **25**, bbad524. https://doi.org/10.1093/bib/bbad524 (2024).
33. Lin, Z. et al. A structured self-attentive sentence embedding. arXiv preprint https://arxiv.org/abs/1703.03130. https://doi.org/10.48550/arXiv.1703.03130 (2017).
34. Kingma, D. P. & Ba, J. in *3rd International Conference on Learning Representations* (Louisiana, 2019).
35. Kohavi, R. in *International Joint Conference on Artificial Intelligence* 1137–1145 (Lawrence Erlbaum Associates Ltd).
36. Bao, Y. et al. Recognizing SARS-CoV-2 infection of nasopharyngeal tissue at the single-cell level by machine learning method. *Mol. Immunol.* **177**, 44–61 (2025).
37. Liao, H. et al. Machine learning analysis of CD4+ T cell gene expression in diverse diseases: Insights from cancer, metabolic, respiratory, and digestive disorders. *Cancer Genet.* **290–291**, 56–60. https://doi.org/10.1016/j.cancergen.2024.12.004 (2025).
38. Chen, L., Gu, J. & Zhou, B. PMiSLocMF: Predicting miRNA subcellular localizations by incorporating multi-source features of miRNAs. *Brief. Bioinfor.* **25**, bbae386 (2024).
39. Chen, L., Chen, Y. & Zhou, B. HCLAMCMI: Prediction of circRNA-miRNA interactions based on hypergraph contrastive learning and an attention mechanism. *J. Chem. Inf. Model.* **65**, 12099–12115 (2025).
40. Powers, D. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *J. Mach. Learn. Technol.* **2**, 37–63 (2011).
41. Matthews, B. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure* **405**, 442–451 (1975).
42. Chen, L. & Li, J. PDTDAHN: Predicting drug-target-disease associations using a heterogeneous network. *Curr. Bioinform.* in press (2025).
43. Chen, L., Zhang, S. & Zhou, B. Herb-disease association prediction model based on network consistency projection. *Sci. Rep.* **15**, 3328 (2025).
44. Chen, L., Lu, Y., Xu, J. & Zhou, B. Prediction of drug's anatomical therapeutic chemical (ATC) code by constructing biological profiles of ATC codes. *BMC Bioinform.* **26**, 86 (2025).
45. Chen, L., Zhu, W. & Chen, D. An end-to-end 3D graph neural network for predicting drug-target-disease associations. *Curr. Bioinform.* (2025).
46. Chowdhury, S. Y., Shatabda, S. & Dehzangi, A. iDNAProt-ES: identification of DNA-binding proteins using evolutionary and structural features. *Sci. Rep.* **7**, 14938 (2017).
47. Swain, P. H. & Hauska, H. The decision tree classifier: Design and potential. *IEEE Trans. Geosci. Electron.* **15**, 142–147 (1977).
48. Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995).
49. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).
50. Tang, S. & Chen, L. iATC-NFMLP: Identifying classes of anatomical therapeutic chemicals based on drug networks, fingerprints and multilayer perceptron. *Curr. Bioinform.* **17**, 814–824 (2022).
51. Chen, L. & Zhao, X. PCDA-HNMP: Predicting circRNA-disease association using heterogeneous network and meta-path. *Math. Biosci. Eng.* **20**, 20553–20575 (2023).
52. Wang, Y., Xu, Y., Yang, Z., Liu, X. & Dai, Q. Using recursive feature selection with random forest to improve protein structural class prediction for low-similarity sequences. *Comput. Math. Methods Med.* **2021**, 5529389. https://doi.org/10.1155/2021/5529389 (2021).
53. Onesime, M., Yang, Z. & Dai, Q. Genomic island prediction via chi-square test and random forest algorithm. *Comput. Math. Methods Med.* **2021**, 9969751. https://doi.org/10.1155/2021/9969751 (2021).
54. Pedregosa, F. et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
55. Jones, P. et al. InterProScan 5: Genome-scale protein function classification. *Bioinformatics* **30**, 1236–1240 (2014).
56. Tsay, Y. F., Chiu, C. C., Tsai, C. B., Ho, C. H. & Hsu, P. K. Nitrate transporters and peptide transporters. *FEBS Lett.* **581**, 2290–2300. https://doi.org/10.1016/j.febslet.2007.04.047 (2007).
57. Ding, L. et al. Aquaporin PIP2;1 affects water transport and root growth in rice (*Oryza sativa* L.). *Plant Physiol. Biochem.* **139**, 152–160. https://doi.org/10.1016/j.plaphy.2019.03.017 (2019).
58. Marrocco, K., Thomann, A., Parmentier, Y., Genschik, P. & Criqui, M. C. The APC/C E3 ligase remains active in most post-mitotic Arabidopsis cells and is required for proper vasculature development and organization. *Development* **136**, 1475–1485. https://doi.org/10.1242/dev.035535 (2009).
59. Kobayashi, T. et al. Iron-binding haemerythrin RING ubiquitin ligases regulate plant iron responses and accumulation. *Nat. Commun.* **4**, 2792. https://doi.org/10.1038/ncomms3792 (2013).

Author contributions

L.C. designed the research; L.C., X.X. and B.Z. conducted the experiments; X.X. and B.Z. analyzed the results; L.C. and X.X. wrote the manuscript. All authors have read and approved the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-026-35110-7.

Correspondence and requests for materials should be addressed to L.C.